Hex Symbols Algorithm for Anti-Forensic Artifacts on Android Devices

Somyia M. Abu Asbeh
Software Engineering Dept.
Princess Sumaya University for Technology
Amman, Jordan

Hamza A. Al-Sewadi
Computer Sience Dept.
Prince Sumaya University for Technology
Amman, Jordan

Sarah M. Hammoudeh
Faculty of Medical and Human Sciences
University of Manchester
Manchester, UK

Arab M. Hammoudeh
College of Medicine
University of Sharjah
Sharjah, UAE

***Abstract*—Mobile phones technology has become one of the most common and important technologies that started as a communication tool and then evolved into key reservoirs of personal information and smart applications. With this increased level of complications, increased dangers and increased levels of countermeasures and opposing countermeasures have emerged, such as Mobile Forensics and anti-forensics. One of these anti-forensics tools is steganography, which introduced higher levels of complexity and security against hackers' attacks but simultaneously create obstacles to forensic investigations.**

A new anti-forensics approach for embedding data in the steganography field is proposed in this paper. It is based on hiding secret data in hex symbols carrier files rather than the usually used file multimedia carrier including videos, image and sound files. Furthermore, this approach utilizes hexadecimal codes to embed the secret data on the contrary to the conventional steganography approaches which apply changes to binary codes. Accordingly, the resulting data in the carrier files will be unfathomable without the use of special keys yielding a high level of attacking and deciphering difficulty. Besides, embedding the secret data in the form of hex symbols, the agreed upon procedure between communicating parties follows a random embedding manner formulated using WinHex software. Files can be exchanged amongst android devices and/or computers. Experiments were conducted for applying the proposed algorithm on rooted android devices, which also are connected to computers. The proposed methods showed advantages over the currently existing steganography approaches, in terms of character frequency, capacity, security, and robustness.

Keywords—Mobile Forensics; Anti-Forensics; Artifact Wiping; Data Hiding; Wicker; Steganography

I. INTRODUCTION

In our days, not only water, air and food are considered to be our basic living needs but a consensus have been established to add technology including computers and mobile phones to this list. As mobile phones rapidly evolved from communication means to reservoirs of personal information and smart applications [1], they allowed their users to be exposed to increasing dangers and complexities. Consequently, many fields and technologies have been developed as counter-measures to such dangers. One of these fields is the Mobile Forensics, which aims at collecting and analyzing digital evidence to resolve mobile issues. However, on the other side, opposing measures such as Anti-Forensics technologies have been developed to hinder the use of mobile forensics [2]. One of these anti-forensics tools is steganography. Steganography systems are utilized to embed secret data in image, audio and video files that can only be discovered by the parties informed of the secret key of the steganography chosen algorithm. Thus, steganography introduces a higher level of complexity that would protect against attacks but at the same time create an obstacle for forensic investigations [3]. Current steganography approaches have been observed to present some disadvantages that require the development of a new approach or a solution to be overcome. This paper will be introducing some of the currently existing anti-forensics approaches and techniques. Thenceforth, the paper will be proposing a new steganography approach that utilizes Hex Symbols to hide data. The proposed approach in this paper has advantages over the currently existing steganography approaches in its capacity, security and robustness. Capacity indicates the quantity of the information that could be embedded in the stego-medium. Security is essential to keep confidential communication, a secret that can't be detected by intruders. Robustness refers to the ability of the stego-medium to handle alterations while maintaining the integrity of the embedded information [4].

The paper will be presenting background information and related work on anti-forensic techniques, artifact wiping, data hiding, and steganography tools and approaches in sections 2 and 3. Then the paper will be elaborating further on anti-forensics steganography in section 4. The description of the newly proposed steganography approach using hex symbols is presented in section 5 accompanied by the explanation of the implementation process in section 6. Finally the new approach is analyzed and discussed in section 7.

II. RELATED WORK

Several steganography tools have been developed, aiming at achieving the best method to embed secret messages, including the least significant bit (LSB) encoding technique, the hash based LSB Techniques, and the Neighborhood Pixel Information.

The use of the LSB substitution technique in video steganography was introduced by Swathi and Jilani [5]. The principle of the technique revolved around finding and replacing the least significant bits in the image frames of the cover videos. all the color image components (i.e. red, green and blue) may be utilized for the same purpose by replacing the LSB in them by bits of the secret message. Thus the message to be hidden undergoes two conversions; the conversion to ASCII code and the conversion to binary representation.

In their work, Dasgupta el al. [6] presented a hash based LSB Techniques in spatial domain, utilizing an algorithm portrayed with AVI (Audio Video Interleave) file as a cover medium. A video stream (AVI) is composed of collection of frames in which the secret data can be concealed as payload. 8 bits of secret data would be concealed at a time in LSB of RGB pixel value of the carrier frames in 3, 3, 2 order respectively. This technique increased the payload and the difficulty of detection by human eyes due to the small variations in colours.

Another steganography tool was described by Hossain et al. [7] in which neighbourhood information are used to calculate the quantity of data that can be embedded in a cover image without causing a noticeable change. The complication and density of the different areas in the cover image are determined. Thence, small quantities of secret data are hidden in the smooth areas, while larger quantities are hidden in the complicated ones. This whole concept is built on psycho visual repetition in grey scale digital images; the edged parts can withstand more change in comparison to the smooth ones.

III. Anti-Forensics

Anti-forensics (AF) techniques are used to avoid and eliminate the possibility of evidence detection by the mobile forensics tools [1]. AF techniques and tools are continuously and rapidly evolving. By understanding the basics and principles of these techniques, more complex approaches or opposing forensic tools can be developed. Two major types of Anti-Forensic techniques, artifact wiping and data hiding, will be briefly presented next.

A. Artifact Wiping

Artifact wiping, also known as sanitation, overwrites data files from digital devices permanently erasing them. Some artifact wiping tools, including Binary Code (BC) wipe, Eraser, and Pretty Good Privacy (PGP) wipe, target empty and unallocated spaces [8].

One of the applications that function against mobile forensics is Wicker, a free application that allows users to send self-destructing files and messages [9]. According to its developers, Wicker has the ability to function without leaving any evidences or traces behind for forensic investigators. The unique characteristic of this application is that it allows the user to set a self-destruct timer to anything they send. To investigate the efficacy of Wicker, an experiment was conducted using an Android smart phone. A newly created account was used to initiate an instant massaging conversation with a Wicker a certain friend. A timer for self-destruction was set for a sample of messages (self-destruction duration: 5 days). Upon testing and searching for traces left from Wicker’s conversations after the set time, none were found.

B. Data Hiding

Data hiding tools have been developed to secretly embed and hide undiscoverable data through multiple approaches. These approaches include transferring data to other portable storage devices and then wiping the data from the phone; making data “invisible” and concealing their existence; embedding data in multimedia (image, audio and video) files; and altering file extensions. Although some of these approaches, such as altering file extensions, are relatively old, evidence have shown that they can still bypass some forensic analysis methods. For instance, an experiment has shown that changing the extension of an .mp4 file to .pdf allowed for hiding it from the evidence tree generator, FTK imager, without applying any changes to its location [10].

IV. The Anti-Forensic Steganography

According to [3], "Steganography is the art and science of hiding information in plain sight". Thus, through steganography, a stego-system unknown to third, uninvolved parties can be created to allow for data exchange under extremely secure conditions. Digitally, data hiding techniques are important tools for the utilization of steganography. Through these tools, image, audio and video steganography can be applied.

Stego algorithms delete the repeated bits in the cover multimedia files and replace these bits with the secret data. Video and audio files with higher qualities contain larger quantities of the repeated bits required for steganography. The perk of relying on video files in hiding information is their relative immunity to hacker attack due to the relative intricacy of their structure in comparison to that of image files. The idea of video steganography is to hide the data in compressed or uncompressed domains that combine sets of frames and audio files [11]. Therefore, the complexity and efficacy of the process is increased in comparison with the simpler forms of steganography, the image and audio steganography. Such complexity allows for higher security against infiltration and hacker attacks. Video based steganography techniques are generally categorized into Spatial domain and frequency domain.

A spatial domain technique embeds the information to be concealed in the intensity pixels of the carrier multimedia file. The advantage of this category of techniques is their use of the Least Significant Bit (LSB) algorithms to embed the load of data. However, the drawback is that the majority of the LSB techniques are susceptible to attacks. In frequency domain techniques, on the other hand, images are transformed to frequency components by using some techniques, such as Fast Fourier Transform (FFT), Discrete Cosine Transformation (DCT) or Discrete Wavelet Transform (DWT). Thenceforth, the messages are planted and hidden in some or all of the transformed coefficients [6].

In brief, the process of steganography is commenced through an agreement of two parties on a stego-system and a secret key for the embedding algorithm. The accordingly chosen embedding algorithm would be responsible for allocating the carrier files according to their bits content. The redundant bits are modified and replaced with the bits of the secret message to be exchanged by parties involved. This process prevents any third party lacking the knowledge of the secret key and the chosen embedding algorithm from discovering the embedded data or breaching the carrier file contents [3]. The general steganography process is summarized in Fig.1.

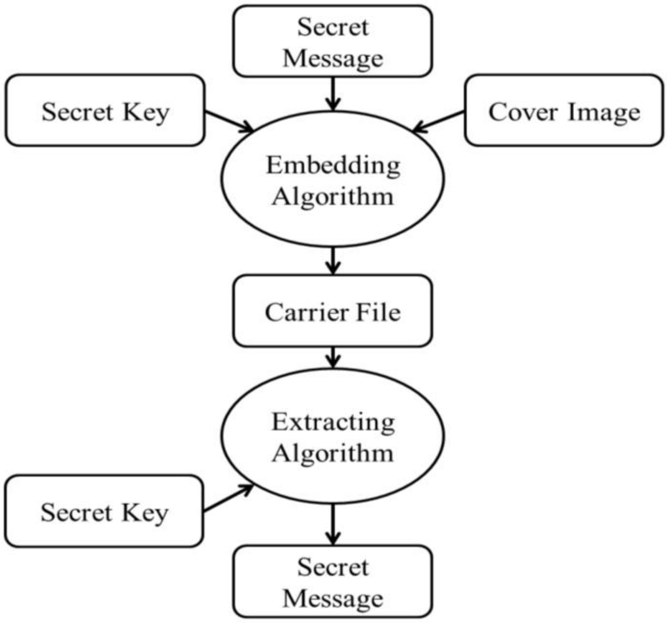


Fig. 1. The general steganography process

V. The Proposed Hex Symbol (HSA) Scheme

A. HSA Design

In this paper, we present a new anti-forensics approach for ciphering data in the steganography field. This approach hides the target data in hex symbols rather than the usually used multimedia files. Accordingly, the resulting data files will be unfathomable without the use of special keys; hence extremely hard to breach and decipher. The approach embeds the target data in the form of hex symbols in a random manner. Winhex facility is used to view the resulting file. The files can then be exchanged through android devices and computers. In this work, experiments were conducted on rooted android devices in combination with computer networks.

The new approach is expected to have several advantages over the traditional steganography approaches. One of these advantages is the impossible detection of these hidden data by the human eyes, as the content will be presented in the form of hex symbols. This problem was constantly observed in the other approaches in the form of disruptions in the sound and video files and changes in image frames and colors. One of the underlying causes of this problem is the limited capacity to conceal data in the traditional multimedia files. Exceeding this capacity leads to the leak of traces of the hidden data that are observable by external parties [4]. The nature of the carrier file used in the new approach allows for surmounting this problem.

Additionally, the traditional steganography methods apply supplementary tools to embed the secret data resulting in the appearance of additional errors and detectable traces during the data hiding process. These tools have been found to cause the concentrated addition of data to a single location in the file, the consistent addition of certain strings to multiple files, or the addition of signatures linking the file to the embedment tool. These flows brought about by the supplementary tools are eliminated in the HSA approach.

The randomness in hiding the data further upraises the level of security as it increases the difficulty of locating the hidden data while the hiding approach increases the difficulty of deciphering the hidden data. However, although this approach would provide a higher degree of security against attacks, it adds a new complication to forensic investigations.

B. Hex Symbols Algorithm (HSA)

To start with, certain patterns would be agreed upon amongst the communicating parties. These patterns are created by converting a chosen file into hexadecimal symbols using WinHex software, segmenting the file's content into 16x16 matrices, and numbering the matrices sequentially as shown in fig. 2. Some of these segments will be selected and used for embedding the secret messages. Embedding the messages would be guided by the previously agreed upon patterns representing the hiding keys shared by the communicating parties, as shown in fig. 3. However, any shape or number of shapes can be designed and used generally. These patterns are arranged randomly in a table, as shown in table I, and shared between the communicating parties to serve as the secret key codebook. Each table entry represents a string of pattern numbers denoted by a letter that is going to be the key.

After the hiding keys and the secret codebook are prepared, the secret message can be embedded in the carrier file according to the following steps:

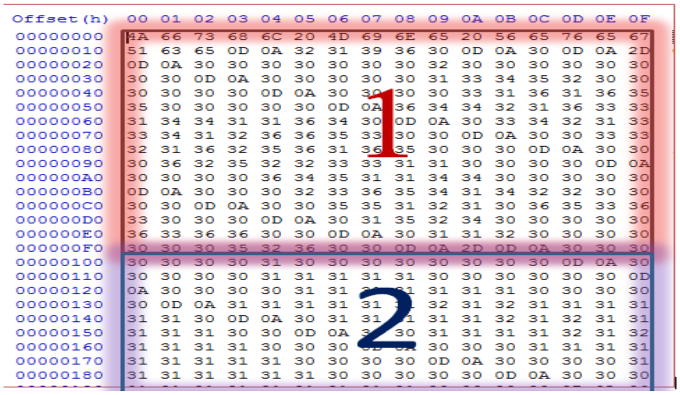

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   4A 66 73 68 6C 20 4D 69 6E 65 20 56 65 76 65 67
00000010   51 63 65 0D 0A 32 31 39 36 30 0D 0A 30 0D 0A 2D
00000020   0D 0A 30 30 30 30 30 30 30 32 30 30 30 30 30 30
00000030   30 30 0D 0A 30 30 30 30 30 31 33 34 35 32 30 30
00000040   30 30 30 30 0D 0A 30 30 30 30 33 31 36 31 36 35
00000050   35 30 30 30 30 30 0D 0A 36 34 34 32 31 36 33 33
00000060   31 34 34 31 31 36 34 30 0D 0A 30 33 34 32 31 33
00000070   33 34 31 32 36 36 35 33 30 30 0D 0A 30 30 33 33
00000080   32 31 36 32 35 36 31 36 35 30 30 30 0D 0A 30 30
00000090   30 36 32 35 32 32 33 33 31 31 30 30 30 30 0D 0A
000000A0   30 30 30 30 36 34 35 31 31 34 34 30 30 30 30 30
000000B0   0D 0A 30 30 30 32 33 36 35 34 31 34 32 32 30 30
000000C0   30 30 0D 0A 30 30 35 35 31 32 31 30 36 35 33 36
000000D0   33 30 30 30 0D 0A 30 31 35 32 34 30 30 30 30 30
000000E0   36 33 36 36 30 30 0D 0A 30 31 31 32 30 30 30 30
000000F0   30 30 30 35 32 36 30 30 0D 0A 2D 0D 0A 30 30 30
00000100   30 30 30 30 31 30 30 30 30 30 30 30 30 0D 0A 30
00000110   30 30 30 30 31 31 31 31 31 30 30 30 30 30 30 0D
00000120   0A 30 30 30 30 31 31 31 31 31 31 31 30 30 30 30
00000130   30 0D 0A 31 31 31 31 31 31 32 31 32 31 31 31 31
00000140   31 31 30 0D 0A 30 31 31 31 31 31 32 31 32 31 31
00000150   31 31 31 30 30 0D 0A 31 30 31 31 31 31 32 31 32
00000160   31 31 31 31 30 30 30 0D 0A 30 30 30 31 31 31 31
00000170   31 31 31 31 31 30 30 30 30 0D 0A 30 30 30 30 31
00000180   31 31 31 31 31 31 30 30 30 30 30 0D 0A 30 30 30
```

Fig. 2. The segmented and numbered hex code of the carrier file

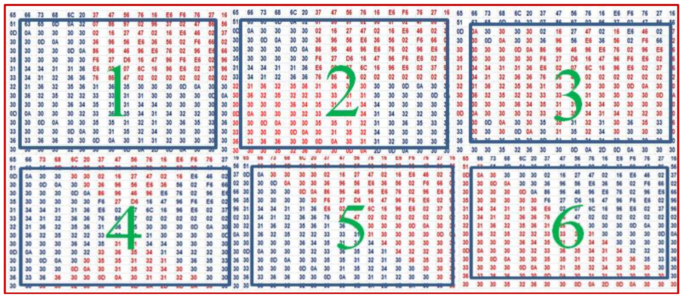


Fig. 3. Specific pattern design for matrices

1) The input secret message is converted into hex symbol.

2) The hex symbol decimals are then invered. For example, if the letter 'n' hex symbol is 63, then it is inverted into 36.

3) the resulting hex codes are then embedded into the carrier file which contains a randomly chosen sequence of matrix segments.

4) The contents of these matrix segments are relocated by exchanging rows with columns in order to increase the difficulty for hackers.

5) Once all the contents of the secret message are hidden, these segments (stego-file) are concatenated with the randomly chosen sequence of the matrix segments (or the key), and sent to the receiving party.

Fig. 4 shows an example of the matrix segment sequence when the key symbol 'S' was selected from table I.

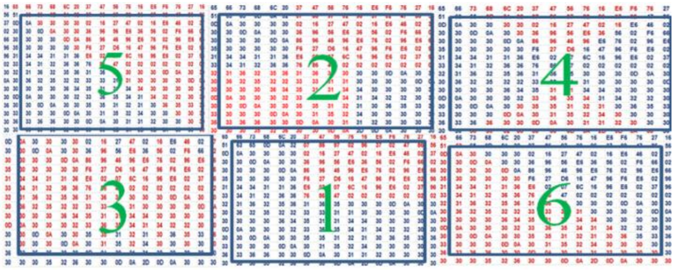


Fig. 4. Key symbol ″S″matrix segment sequence ″524316″

TABLE I. EXAMPLE OF THE SHARED KEY SYMBOL CODEBOOK

Key symbol	**Selected random sequence**
S	524316
O	643125
M	365123
Y	513642

A summary of the proposed HSA steganography algorithm is presented in fig. 5-a. The receiving party would be get the stego-file carrying the secret message as well as a key indicating the chosen pattern in a numeric representation. Accordingly, the receiving party would be able to comprehend the arrangement of the matrices by referring back to the secret codebook. Thereafter, the steganography steps can be executed in reverse with the guidance of the chosen secret keys and patterns to decode the hidden message (Fig. 5-b).

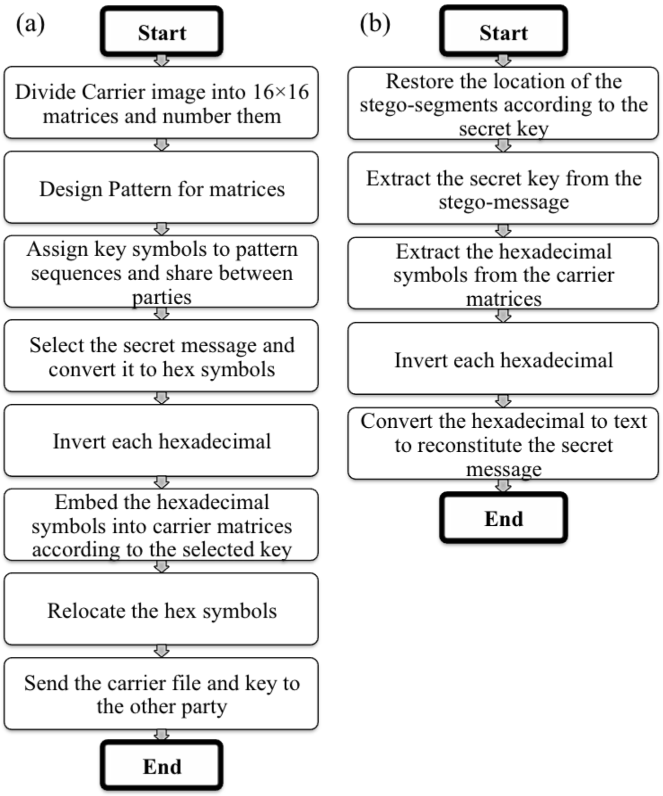
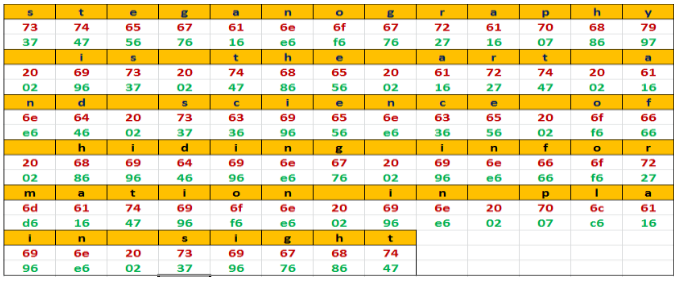

(a)
Start
Divide Carrier image into 16×16 matrices and number them
Design Pattern for matrices
Assign key symbols to pattern sequences and share between parties
Select the secret message and convert it to hex symbols
Invert each hexadecimal
Embed the hexadecimal symbols into carrier matrices according to the selected key
Relocate the hex symbols
Send the carrier file and key to the other party
End
(b)
Start
Restore the location of the stego-segments according to the secret key
Extract the secret key from the stego-message
Extract the hexadecimal symbols from the carrier matrices
Invert each hexadecimal
Convert the hexadecimal to text to reconstitute the secret message
End

Fig. 5. The Flow Charts summarize the procedures of (a) embedding secret messages by the sending party and (b) extracting hidden messages by the receiving party according to the proposed HSA scheme

VI. IMPLEMENTATION

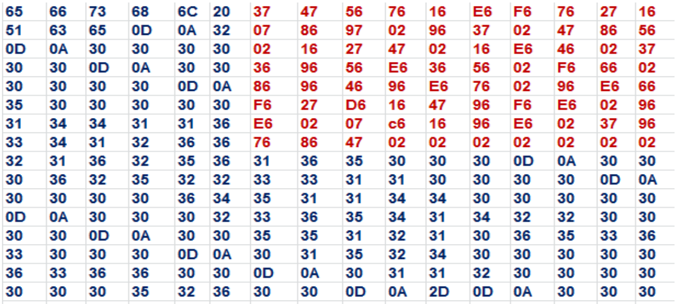
To implement the proposed HSA scheme, a message was embedded in a carrier file according to the following process. The secret message to be hidden was "Steganography is the art and science of hiding information in plain sight". First, the message was converted to Hexadecimal symbol representation. In this representation, each letter of the message is represented by two hexadecimal character components as shown in each first and second row of fig. 6. The two hexadecimal character components representing each letter are then inverted, resulting in the new hex symbols content shown in each third row of fig. 6.



s	t	e	g	a	n	o	g	r	a	p	h	y
73	74	65	67	61	6e	6f	67	72	61	70	68	79
37	47	56	76	16	e6	f6	76	27	16	07	86	97
	i	s		t	h	e		a	r	t		a
20	69	73	20	74	68	65	20	61	72	74	20	61
02	96	37	02	47	86	56	02	16	27	47	02	16
n	d		s	c	i	e	n	c	e		o	f
6e	64	20	73	63	69	65	6e	63	65	20	6f	66
e6	46	02	37	36	96	56	e6	36	56	02	f6	66
	h	i	d	i	n	g		i	n	f	o	r
20	68	69	64	69	6e	67	20	69	6e	66	6f	72
02	86	96	46	96	e6	76	02	96	e6	66	f6	27
m	a	t	i	o	n		i	n		p	l	a
6d	61	74	69	6f	6e	20	69	6e	20	70	6c	61
d6	16	47	96	f6	e6	02	96	e6	02	07	c6	16
i	n		s	i	g	h	t					
69	6e	20	73	69	67	68	74					
96	e6	02	37	96	76	86	47					

Fig. 6. Secret message hex symbols after inversion

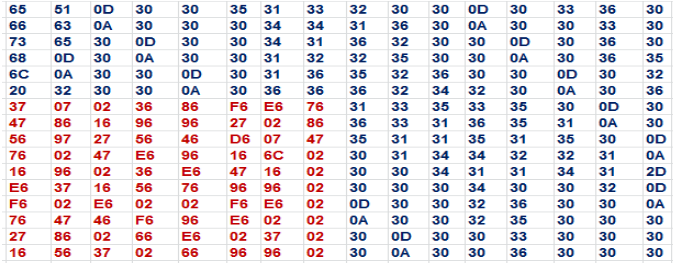
The resulting hex symbols, representing of the secret message, were henceforth embedded into the carrier hex file according to the chosen pattern, as shown in Fig. 7.



65	66	73	68	6C	20	37	47	56	76	16	E6	F6	76	27	16
51	63	65	0D	0A	32	07	86	97	02	96	37	02	47	86	56
0D	0A	30	30	30	30	02	16	27	47	02	16	E6	46	02	37
30	30	0D	0A	30	30	36	96	56	E6	36	56	02	F6	66	02
30	30	30	30	0D	0A	86	96	46	96	E6	76	02	96	E6	66
35	30	30	30	30	30	F6	27	D6	16	47	96	F6	E6	02	96
31	34	34	31	31	36	E6	02	07	c6	16	96	E6	02	37	96
33	34	31	32	36	36	76	86	47	02	02	02	02	02	02	02
32	31	36	32	35	36	31	36	35	30	30	30	0D	0A	30	30
30	36	32	35	32	32	33	33	31	31	30	30	30	30	0D	0A
30	30	30	30	36	34	35	31	31	34	34	30	30	30	30	30
0D	0A	30	30	30	32	33	36	35	34	31	34	32	32	30	30
30	30	0D	0A	30	30	35	35	31	32	31	30	36	35	33	36
33	30	30	30	0D	0A	30	31	35	32	34	30	30	30	30	30
36	33	36	36	30	30	0D	0A	30	31	31	32	30	30	30	30
30	30	30	35	32	36	30	30	0D	0A	2D	0D	0A	30	30	30

Fig. 7. Example of embedded message into the carrier file segment (the embedded secret message represented by bold red color)

The contents of the carrier stego-file segment were then rearranged by interchanging the positions of rows and columns as shown in Fig. 8.



65	51	0D	30	30	35	31	33	32	30	30	0D	30	33	36	30
66	63	0A	30	30	30	34	34	31	36	30	0A	30	30	33	30
73	65	30	0D	30	30	34	31	36	32	30	30	0D	30	36	30
68	0D	30	0A	30	30	31	32	32	35	30	30	0A	30	36	35
6C	0A	30	30	0D	30	31	36	35	32	36	30	30	0D	30	32
20	32	30	30	0A	30	36	36	36	32	34	32	30	0A	30	36
37	07	02	36	86	F6	E6	76	31	33	35	33	35	30	0D	30
47	86	16	96	96	27	02	86	36	33	31	36	35	31	0A	30
56	97	27	56	46	D6	07	47	35	31	31	35	31	35	30	0D
76	02	47	E6	96	16	6C	02	30	31	34	34	32	32	31	0A
16	96	02	36	E6	47	16	02	30	30	34	31	31	34	31	2D
E6	37	16	56	76	96	96	02	30	30	30	34	30	30	32	0D
F6	02	E6	02	02	F6	E6	02	0D	30	30	32	36	30	30	0A
76	47	46	F6	96	E6	02	02	0A	30	30	32	35	30	30	30
27	86	02	66	E6	02	37	02	30	0D	30	30	33	30	30	30
16	56	37	02	66	96	96	02	30	0A	30	30	36	30	30	30

Fig. 8. Embedded message into the carrier file segment (the embedded secret message represented by bold red color)

For example, the whole segment elements could be flipping around the diagonal according to eq. 1:

$$x'_{ij} = x_{ji} \tag{1}$$

Where x'_{ij} are the new matrix elements of the stego-file and x_{ji} are the old matrix elements.

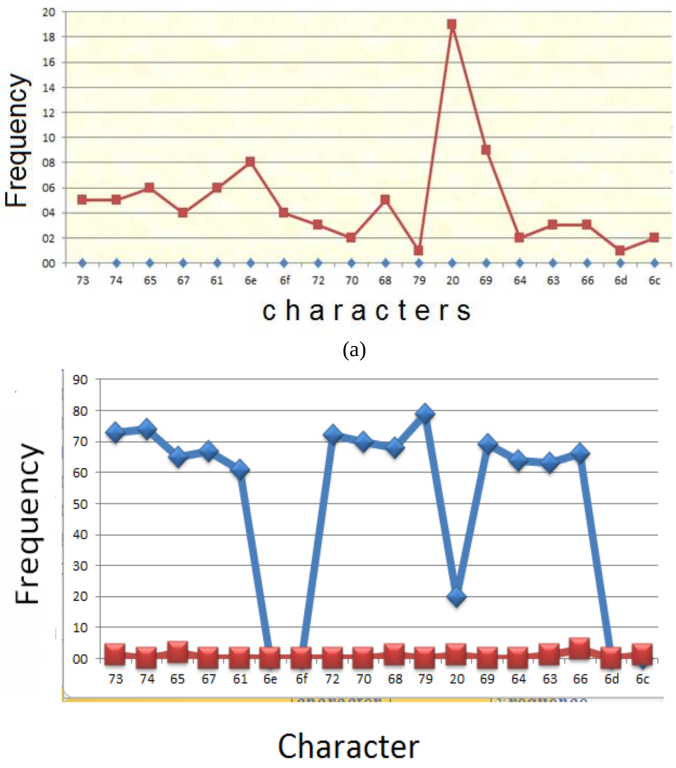
The resulting final form of the stego-file can be safely sent to the other parties. With the chosen secret patterns and keys, they can retrace the steps back and recover the embedded text.

VII. Analysis and Discussion

The frequency, capacity, using WinHex, the evidences and robustness of the proposed HSA scheme were analyzed as follows.

A. Frequency

In As the two hexadecimal character components of the hex symbol are inverted, the calculated frequency of occurrence for each character before inversion will differ from that after inversion (fig. 9-a and 9-b).



(a)

(b)

Fig. 9. Character frequency for embedded message (a) before inversion (b) after inversion

From the compiled frequencies analysis in table II and fig. 10, a clear change was observed between the frequency of characters before and after the inversion process. This is a positive indicator of the high level of security against third attacking parties. Furthermore, elongating the embedded sentence is expected to further increase the security factor of the process.

TABLE II. Character Frequency Before (F.B) and After (F.A) Inversion

Character	F.B	F.A
73	05	01
74	05	00
65	06	02
67	04	00
61	06	00
6e	08	00
6f	04	00
72	03	00
70	02	00
68	05	01
79	01	00
20	19	01
69	09	00
64	02	00
63	03	01
66	03	03
6d	01	00
6c	02	01

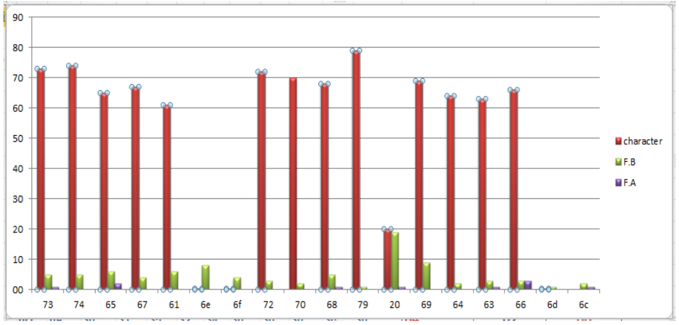


Fig. 10. Character frequency before (F.B) and after (F.A) inversion

B. The Capacity

TABLE III. THE CAPACITY COMPARISON

Steganography Approaches		Comparison category
Hash based LSB Techniques representation	Hex Symbols representation	Method for embedded
Binary	Hex	Type of ASCII Numerical representation

A= 01000001
R= 01010010
T= 01010100
	A= 41
	R= 52
	T= 54
		"ART" in ASCII representation

Select color value for example:
G=10010100
R=10110111
B=11001001
	Select hexadecimal for example:
	61 62 64
		Carrier Image

1001**010**0 1011**00**1**0** 110010**01** 1001**010**0 1011**00**1**1** 110010**10** 1001**010**0 1011**01**1**1** 110010**00**	**41 52 54**	Carrier Image containing the embedded word "ART"
• Binary numbers allow the use of only 2 symbols (0, 1) to symbolize any number and it becomes a tiresome job to express large numbers [12]. • When a large number is represented in binary system, it results into extremely difficult, lengthy and non- readable by human [12]. • More code but less capacity	• Hexadecimal numbers allow the use of 16 symbols 0 to 9 and additional symbols (A, B, C, D, E, F). Hexadecimal numbers were presented to fulfil the aim of symbolizing binary numbers in a more human readable form [12]. • When numbers are represented in hexadecimal system, they are considered to be easier and more human readable than binary number [12]. • Less code and more capacity	Output

Comparisons of some capacity and size characteristics between the proposed HSA scheme and the hash based LSB technique have been enlisted in table III. This comparison has shown the lower capacity requirement of the proposed steganography approach, which allows for the addition of larger extensions of messages without having a large effect on the carrier file.

C. Using WinHex

The use of WinHex to formulate the hex symbols during the hiding process is advantageous as the content will be difficult to trace and compare with previous versions. This advantages is further boosted by the frequent and continuous rearrangements of the hex symbols according to the chosen codebooks and patterns throughout the steganograohy procedure.

Moreover the alteration of the hex symbols by the inversion of each character element doesn't increase the original file size, leaving it stable and unchanging in terms of elements number.

Furthermore, the use of random numbers to select the segments provides an extra complication against deciphering the hidden text. A comparison between available steganalysis tools and hex symbols is presented in Table I V.

TABLE IV. A COMPARISON BETWEEN STEGANALYSIS AND HEX SYMBOLS

STEGANALYSIS TOOLS		HEX SYMBOLES
OurSecret OmniHide BDV DataHider Max file encryption Masker StegoStick	These tools work by embedding information within videos by attaching it bluntly to the end of the file EOF [3].	Hex symbols substitutes the hexadecimal precisely on the same position.
OurSecret	This signature can be found after the last byte of the authentic unmodified file.	A valid signature similar to OurSecret does not apprear.
OmniHide Pro	White space characters tailing the initial sequence of bytes.	Hex symbols do not show the name of the embedded file.

D. The Evidences

To be able identify hidden data, investigators search for steganography tools, such as S-Tools, DPEnvelope, jpgx, on the suspect's personal devices (i.e. phones and computers). The presence of such tools would imply the highly possibility of finding hidden carrier files modified using these tools. Therefore, the investigators further expand their search to identify any possible multimedia or text files that could have been used to hide data [13]. However, with regards to the proposed HSA scheme, no specific tools are used to insert the embedded text; Common programming means found in widely used software, such as VBA found in Microsoft Excel, can be used in this approach to eliminate the shortcomings of the use of external tools. Moreover, the hex symbol file extension can be changed to thwart hackers and investigators.

E. Compression

The stego-files have been found to be resistant against changes in size and content when compressed to WinRAR or ZIP file format and when processed for message extraction. This resistance indicates the robustness of the proposed approach against processing procedure that could be applied to the carrier file such as compression.

VIII. Conclusion & Future Work

Hex symbol algorithm (HSA) scheme is a newly proposed steganography approach developed for hiding secret messages in hex symbols rather than the usually used multimedia files. The files can be exchanged with random keys through android devices or computers. In terms of character frequency, capacity, using WinHex, the evidences and robustness, HSA stagnography has shown improved outcomes in comparison with other steganography approaches. Hidden data using this approach are impossible to detect by the human eyes. Furthermore, traces such as changes in file size and clarity as well as additions of extra information at the end of files have been found to be eliminated using HAS steganography. In the future, this approach can be developed further to increase its complexity and utilize it in various applications. In the future, this approach can be developed further to increase its complexity and utilize it in various applications.

Acknowledgment

We thank prof. Nihad Yusuf – Dean of King Abdullah I Faculty of Graduate Studies and Scientific Research, prof. Arafat Awajan – Dean of King Hussein Faculty for Computing Sciences, and Dr. Saqer Abdel Rahim – Head of the computer science department in King Huessein Faculty for computing Sciences, for their continuous support of this work.

References

[1] A. Distefano, G. Me and F. Pace, "Android anti-forensics through a local paradigm," Digital Investigation, vol. 7, pp. S83-S94, August 2010.

[2] K. Dahbur and B. Mohammad “The Anti-Forensics Challenge,” Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications - ISWSA '11, ACM Press, April 2011.

[3] T. Sloan and J. Hernandez-Castro, “Forensic analysis of video steganography tools,” PeerJ Computer Science, vol. 1, pp. e7, May 2015.

[4] S. Sirsikar and A. Deshpande, “Steganographic Tools for BMP Image Format,” International Journal of Computer Science & Emerging Technologies (IJCSET), vol. 2, pp. 200-204, February 2011.

[5] A.Swathi, Dr. S.A.K Jilani “Video Steganography by LSB Substitution Using Different Polynomial Equations” International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5.2012.

[6] K.Dasgupta1, J.K. Mandal and P.Dutta “HASH BASED LEAST SIGNIFICANT BIT TECHNIQUE FOR VIDEO STEGANOGRAPHY(HLSB) “ International Journal of Security, Privacy and Trust Management (IJSPTM), Vol. 1, No 2, April 2012

[7] M.Hossain, S. Al Haque, and F.Sharmin “ Variable Rate Steganography in Gray Scale Digital Images Using Neighborhood Pixel Information “ The International Arab Journal of Information Technology, Vol. 7, No. 1, January 2010.

[8] P. A. Kotsopoulos and Y. C. Stamatiou, “Uncovering Mobile Phone Users’ Malicious Activities Using Open Source Tools,” Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on, Istanbul, pp. 927-933, August 2012.

[9] T. Mehrotra and B. M. Mehtre, “Forensic Analysis of Wickr Application on Android Devices,” 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Enathi, pp. 1-6, December 2013.

[10] A. Jain and G. S. Chhabra, “Anti-Forensics Techniques: An Analytical Review,” 2014 Seventh International Conference on Contemporary Computing (IC3), Noida, pp. 412 – 418, August 2014.

[11] K. Dasguptaa, J. K. Mondalb, and P. Dutta, “Optimized Video Steganography using Genetic Algorithm (GA),” Procedia Technology, vol. 10, pp. 131-137, 2013. [International Conference on Computational Intelligence: Modeling, Techniques and Applications (CIMTA) 2013].

[12] "Difference between binary and hexadecimal", Schoolelectronic.com. [Online]. Available: http://www.schoolelectronic.com/2012/10/difference-between-binary-and-hexadecimal.html. [Accessed: Jan- 2016].

[13] B. Nelson, A. Phillips and C. Steuart, “Guide to computer forensics and investigations,” 4th ed. Boston, MA: Course Technology Cengage Learning, 2010.