Image and AES Inspired Hex Symbols Steganography (IAIS) for Anti-Forensic Artifacts

Somyia M. Abu Asbeh
Software Engineering Dept.
Princess 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*—Technology (including mobiles and computers) has become a basic, indispensable need in our daily life. With an initial purpose of achieving basic functions such as communication, technology has evolved into a virtual gate to the whole world connecting individuals through social media and various websites and applications. Most importantly, technology became the reservoir of our personal information and important, sensitive data. This has led to increased risks of security breaches and data thefts demanding countermeasure approaches. One of these approaches is Steganography. Steganography is a data hiding approach that allows for invisible, relatively safe communication. Several forms of steganography have been developed, among which are Image steganography and our previously developed AES Inspired Steganography. In this paper we propose a new variation in which we combine both of these approaches, Image and AES Inspired Steganography (IAIS). This approach proposes hiding the hex symbol format of the encrypted secret data into a carrier image file. The image file is converted to a hexadecimal representation in which the hex symbol could be embedded without applying any noticeable changes to the original image. Deciphering the hidden information requires secret keys agreed upon by the communicating parties confidentially. These carrier files can be exchanged among mobile devices and/or computers. Comparisons between the original cover images and the cover images with the hidden text have shown that no changes occurred in the colour histogram of the images. However, the noise test has shown that exposure to noise can affect the hexadecimal content of the image, hence the embedded hex symbol representation of the secret text.**

Keywords—Mobile forensics; anti-forensics; data hiding; steganography; AES; AIS

I. INTRODUCTION

Since the rise of the Internet, information security became an important aspect in the field of information technology and communication. Cryptography was created as a technique for ensuring the secrecy of communicated information. Various approaches of data encryption and decryption were developed to prevent access by third parties. Unfortunately, discovering the existence of an encrypted text would increase the risk of access by third party. Therefore, steganography was developed to conceal the existence of the secret text.

The basic concept of Steganography is to achieve invisible communication through hiding secret information in other information files of different formats. Current technological approaches of steganography use digital data files as carrier files and networks as high speed delivery channels [1]. In this paper, we have used both encryption and steganography to ensure a high level of information security against intrusions and hacking attempts. This approach will utilize our previously developed AES inspired hex symbol steganography [2] [3] in combination with the image steganography.

II. RELATED WORK

This section will be discussing some previously published approaches of cryptographic algorithms and data hiding.

A. Cryptographic algorithms

Generally, cryptographic algorithms can be classified into three basic classes; hash algorithm, symmetric key algorithms, and asymmetric key algorithms. They are defined by the number of cryptographic keys used in conjunction with the algorithm [4].

1) Cryptographic Hash Algorithms: No keys are required in this algorithm. They generate a relatively small digest (hash value, hash code, or hash sum) from a (possibly) large input in a way that is fundamentally difficult to reverse (i.e., hard to find an input that will produce a given output). Hash functions are used as building blocks for key management, namely; providing data authentication and integrity services (e.g. generating a hashed message authentication code, HMAC), message compression for digital signature generation and verification, generating deterministic random numbers [5]. Examples of this type are SH, SH-1, MD5, etc.

2) Symmetric Key Algorithms: A secret key is pre-shared between the sender and the receiver over a secure channel before the communication. The word "symmetric" refers to the fact that both sender and receiver use the same key to encrypt and decrypt the information, respectively [6]. Examples of the widely used modern symmetric cryptosystem are DES, 3DES, AES, Blowfish, RC5, RC2, CAST and IDEA.

3) Asymmetric Key Algorithms: Asymmetric algorithms rely on a pair of keys; one key is used for encryption and a different but related key is used for decryption, with the characteristic of being computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key [7].

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 or computer; 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 has 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 [8].

1) Least Significant Bit Encoding (LSB)

The principle of this technique is to find and replace the least significant bits in the image frames of the cover video. If each pixel in the gray carrier image consists of 8 bits for example, then the LSB of this pixel is replaced by a bit from the secret message. However, for an RGB (Red, Green and Blue) carrier image, each of the red, green, and blue components which are each of 8-bits lengths can be utilized for the same purpose by replay the LSB in them by bits of the secret message [9].

Thus the secret message to be hidden in the carrier image undergoes two processes, the first is conversion to ASCII code and the second is conversion to binary representation with 8 bits for each of the words.

2) A Hash Based LSB Technique (HLSB)

Dasguptal et al. [10] reported a hash based LSB Techniques in spatial domain. It is a utilization of an algorithm portrayed with AVI (Audio Video Interleave) file as a cover medium. A video stream (AVI) composed of collection of frames and the secret data is concealed in these frames as payload. The information of the cover video (AVI) such as number of frames (n), frame speed (fp/sec), frame height (H) and frame width (W) are extracted from the header.

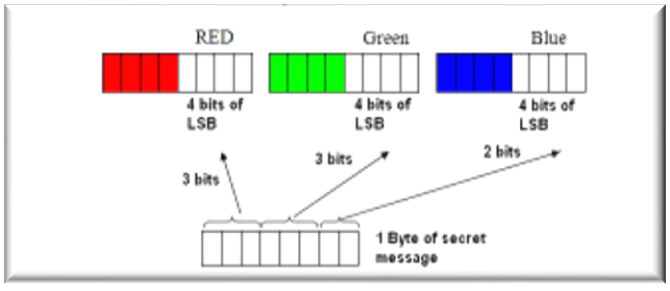


Fig. 1. Secret data embedded in RGB pixels of carrier frame [10].

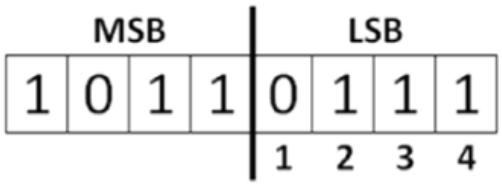


Fig. 2. The hiding positions of the four bits (4) available bits of LSB's side.

The cover video is then divided and separated into frames. The size of the message is irrelevant in video steganography due to the fact that the message could be embedded in multiple frames. The advised technique conceals 8 bits of secret data at a time in LSB of RGB pixel value of the carrier frames in 3, 3, 2 order respectively. Such that out of eight 8-bits of message 3-bits are inserted in R pixel, 3-bits are inserted in G pixel and remaining 2- bits are inserted in B pixel, as shown in Fig. 1. This distribution pattern is applied because the chromatic influence of the blue pixel color component to the human eye is more than that of the red and green pixel components.

The hash based LSB technique would also increase the payload. Moreover, the fact that the variation between the colours is small makes its detection by the human eye a very strenuous task. Using a hash function of this form, the hiding positions of the eight bits out of the four available bits of LSB as shown in Fig. 2 could be obtained

$$\boldsymbol{k} = \boldsymbol{p} \qquad \ldots\ldots \qquad (1)$$

Where, $\boldsymbol{k}$ is LSB bit position within the pixel, $\boldsymbol{p}$ stands for the position of each hidden image pixel and n represents the number of bits of LSB which is 4 for the present case.

A stego video is then formed after concealing data in multiple frames of the carrier video by grouping these frames together. That video will be, used as normal sequence of streaming frames.

The reverse steps could be followed by the intended user to uncover and extract the secret data. The decoding process of a setgo video consists of breaking the video into frames again after going through the header information. The data of the secret message is regenerated, by utilizing the same hash function known to the intended user.

3) Neighbourhood Pixel Information

Hossain et al. [11] proposed three effective steganography tools that use the neighborhood information to calculate the quantity of data which could be embedded in a cover image input pixel without causing a noticeable change. The smooth and complicated areas of an image are determined by the neighborhood relationship. According to this concept, small quantities of secret data are hidden in the smooth areas, and large 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. THE PROPOSED IMAGE AND AES INSPIRED STEGANOGRAPHY (IAIS)

A. Image and AES Inspired Steganography (IAIS) Design

In this paper, we will be introducing a new data hiding and encryption method in which we combine our previously developed AES Inspired Steganography (AIS) with image steganography. This combination would allow for an increased level of security and concealment of the hidden secret data. This Steganography approach will include the use of a table to specify the hiding location of the secret data. In general, this method consists of multiple steps of encryption applied to the secretly hidden message. First, the message is embedded into a

hex symbols carrier file, which is divided into embedding matrices and cipher key matrices according to varied patterns chosen by the communicating parties. This approach was discussed in detail in our previous paper on the AES Inspired Steganography [2]. The hex symbol carrier file as a whole is then embedded into the chosen cover image as shown in Fig. 3. Furthermore, throughout the steganography process, encryptions and rearrangements are continuously applied allowing for increased security measures.

B. Image and AES Inspired Steganography (IAIS) Algorithm

Initially, the communicating parties (i.e. sender and receiver) agree upon certain tables that will be used as keys for embedding and extracting of the secret message content. Secret content could be embedded at different locations in the image including the header, the tail or both or could be attached to the end of the image and embedding of AIS processed hex symbol within the image . The method of embedding the AIS encrypted text at each of these locations will be described next.

C. Attachment of AIS processed hex symbol to the end of the image

Once AIS is applied to a secret text, a cover image is chosen (e.g. Fig. 4(a)). The chosen picture is then converted to a hexadecimal file using WinHex. The created AIS hex symbol in then embedded in the carrier image by attaching it to the end of the hexadecimal format of the carrier image. A deciphering key is embedded as well in the hexadecimal file in the location agreed upon by the communicating parties. Afterwards, the resulting hexadecimal file is converted back to the image format which can be sent to the second communicating party Fig. 4(b). The second communicating party receives the cover image containing the hex symbol representation of the secret text and the decryption key. They key consists of two parts: 1) a hex symbol code book (in alphabetic codes); and 2) an index number determining the location and size of the hidden hex symbol in the image (Fig. 5).

D. Embedment of AIS processed hex symbol in the image's header, tail, or both

Similarly, a cover image is chosen after the creation of the hex symbol form of the hidden secret text (Fig. 6(a), 7(a)) and is converted to a hexadecimal file using WinHex. The hex symbol is then embedded in the header or tail of the cover image's hexadecimal format or both, simultaneously.

A deciphering key is embedded as well in the hexadecimal file in the location agreed upon by the communicating parties. The cover image carrying the hidden text and the key will then be converted back to the image format from the hexadecimal format and sent to the receiving party (Fig. 6(b), 7(b)).

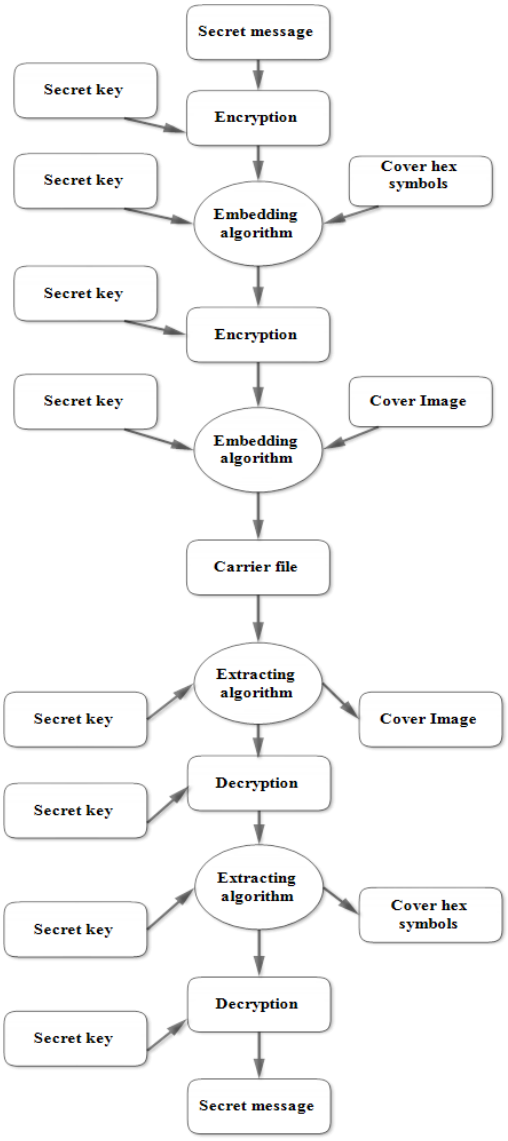


Fig. 3. The general Image and AES Inspired Steganography (IAIS) process.

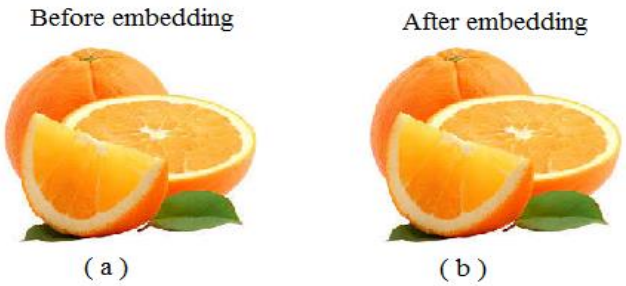


Fig. 4. Cover image before and after attaching the AIS encrypted text to the end of the image.

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
16 ||	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Index	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
32 ||	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
...	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..

Fig. 5. Example of the shared key index book.

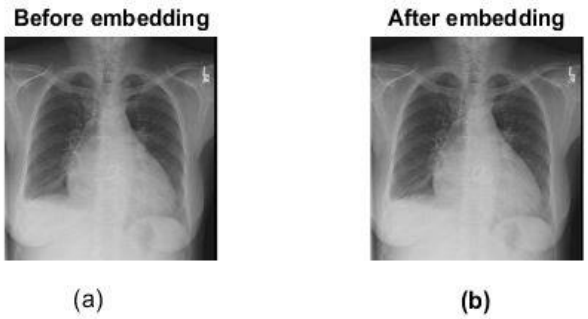


Fig. 6. Embedment of the hex symbol in the header.

The key embedded in the carrier file consists of two parts: 1) a hex symbol code book (in alphabetic codes); and 2) an index number specifying the location and size of the hidden hex symbol. Symbols specifying the allocation to the header, tail or both are specified as follows:

- Header: (#); in hexadecimal representation = 23
- Tail: ($); in hexadecimal representation = 24
- Header and tail: (^); in hexadecimal representation = 5e

In the key, these symbols are written in different locations for additional concealment and security.

E. Embedment of AIS processed hex symbol within the image

After the secret text is encrypted using the AIS approach, multiple cover images are chosen. The chosen images are then combined into a collage according to user's preferences (Fig. 8(a)). A snapshot (e.g. using the snipping tool) is then taken of the images' collage allowing for the creation of spaces available for embedment within the hexadecimal representation. The AIS processed hex symbol containing the secret text is then embedded into one of these available spaces. The deciphering key is embedded as well in the location agreed upon by the communicating part. The embedded key consists similarly of the hex symbol code book (in alphabetic codes) and an index number indicating the location (e.g. * (in hexadecimal representation= 2a) could be used to indicate the first character of the hex symbol) and size of the secret hex symbol. The hexadecimal file is then converted back into an image format and sent to the receiving party (Fig. 8(b)).

IV. IMPLEMENTATION

Referring back to the example used in our previous paper on AES Inspired Steganography [2], the secret message "Steganography is the art and science of hiding information in plain sight. EAS is a symmetric encryption" was embedded in a hex symbol carrier file (Fig. 9).

The next step is to hide this AIS processed hex symbol in a cover image using the third discussed approach in which the hex symbol is embedded within an image (specifically, a collage of multiple images). Therefore, multiple cover images are chosen for the image steganography step as shown in Fig. 10. The chosen images are then combined in any desired order and snapshot as one picture (Fig. 11).

The created cover image (Fig. 11) is converted to a hexadecimal file using Winhex. The hex symbols created using the AIS approach is then embedded into the hexadecimal format of the cover image (Fig. 12). A key is embedded as well in the cover image at a specific location as agreed initially by the communicating parties. The resulting file is then converted back to the image format, which is then sent to the receiving party as shown in Fig. 13.

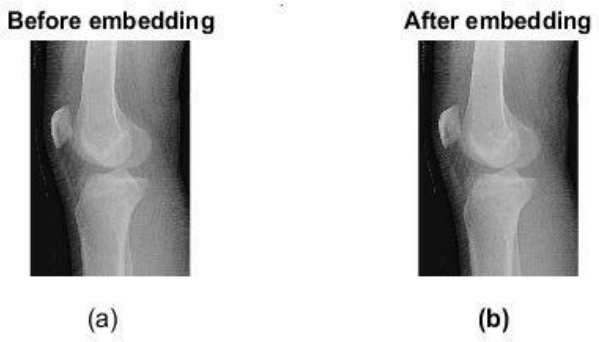


Fig. 7. Embedment of the hex symbol in the tail.

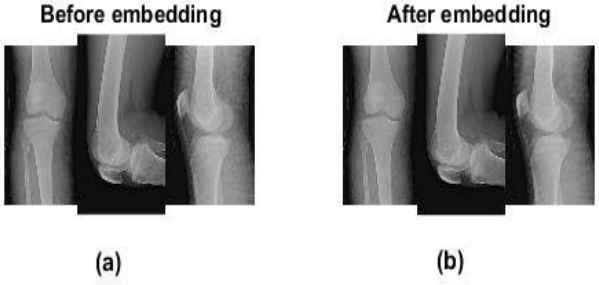


Fig. 8. Cover images before and after the embedment of the AIS encrypted text.

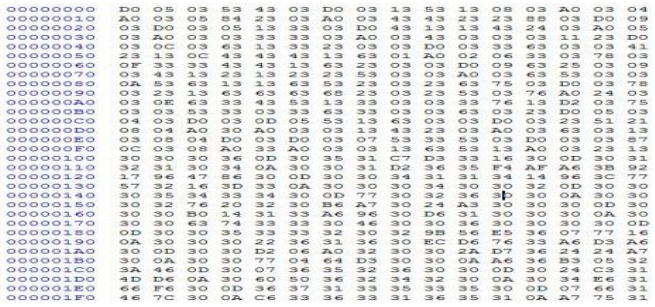


Fig. 9. Hex symbols after embedding.

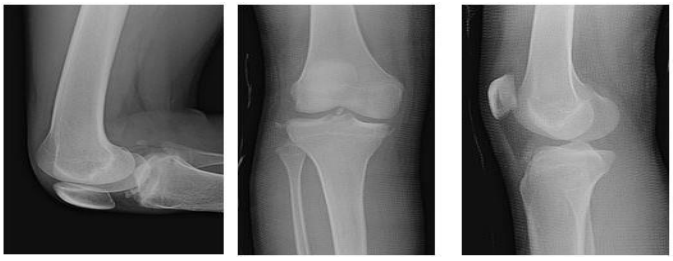


Fig. 10. An example of the multiple cover images that could be chosen for creating the cover images collage.

Fig. 11. Snapshot of the arrangement (collage) of the chosen cover images.

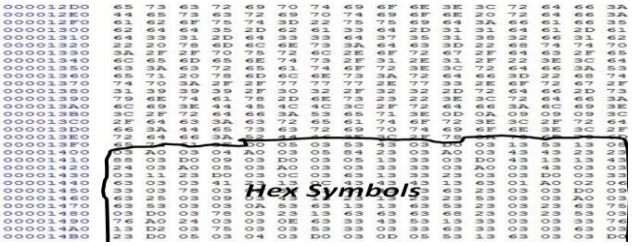


Fig. 12. Hex symbols in image.

Fig. 13. Carrier file containing hex symbols after the image steganography process.

V. Analysis and Discussion

A. Robustness

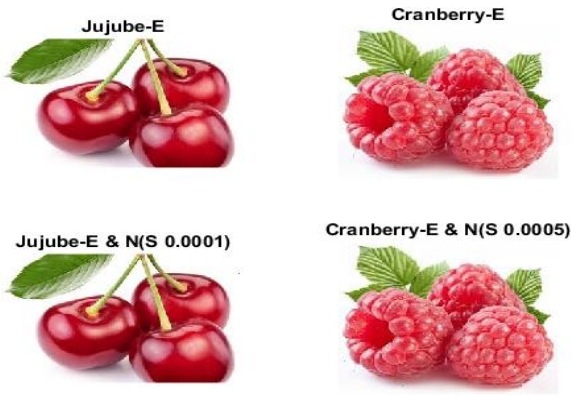
The stego-files have been found to be resistant against changes in size and content when compressed with WinRAR or ZIP file format and then processed for message extraction. This resistance indicates a kind of robustness against processing procedure that could be applied to the carrier file such as compression. Also, the robustness means the resistance to external effects such as noise or other image processing. Experiments of adding noise to the stego image were conducted as below. Two type of noise, namely Gaussien noise and Salt and Pepper noise were introduced to stego images and the mean square error (MSE) and peak signal to noise ratio (PSNR) were calculated for different percentages of noise. These results are listed in Fig. 14 and Fig. 15. The noise effect on the images of the data listed in Fig. 14 and 15 is shown in the Fig. 16 and 17 for both Salt pepper noise and Gaussian. The MSE and PSNR were also plotted in Fig. 18 and 19.

Numb	Pic name	Noise percentage	SNR	PSNR	MSE
1	jujube-E	0.0001	42.4780	45.2887	1.9240
2	Cranberry-E	0.0005	34.6368	37.3139	12.0696
3	Grape-E	0.001	28.5604	33.1917	31.1826
4	Pineapple-E	0.002	29.2412	30.3633	59.8073
5	Orange-E	0.005	24.8986	27.0396	128.5646
6	Banana-E	0.01	22.7457	23.6880	278.1515
7	Apple-E	0.02	17.9092	20.8372	536.2466
8	Kiwi-E	0.03	16.7490	18.9780	822.7686
9	Pomegranate-E	0.06	13.9033	16.3126	1.5199e+03
10	Strawberries-E	0.1	11.8404	13.9782	2.6017e+03

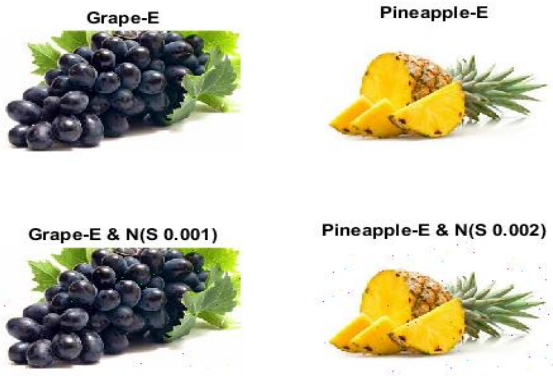
Fig. 14. Calculated the (MSE) and (PSNR) for Salt & pepper noise.

Num	Pic name	Noise percentage	SNR	PSNR	MSE
1	jujube-E	0.0001	18.3941	21.2049	492.7141
2	Cranberry-E	0.0005	18.2903	20.9674	520.4081
3	Grape-E	0.001	16.3100	20.9412	523.5536
4	Pineapple-E	0.002	20.9047	22.0267	407.7655
5	Orange-E	0.005	19.3355	21.4765	462.8409
6	Banana-E	0.01	21.1186	22.0609	404.5679
7	Apple-E	0.02	18.4884	21.4164	469.2909
8	Kiwi-E	0.03	19.5140	21.7430	435.2949
9	Pomegranate-E	0.06	18.4390	20.8482	534.8790
10	Strawberries-E	0.1	17.6281	19.7659	686.2553

Fig. 15. Calculated the (MSE) and (PSNR) for Gaussian noise.



(a)



(b)

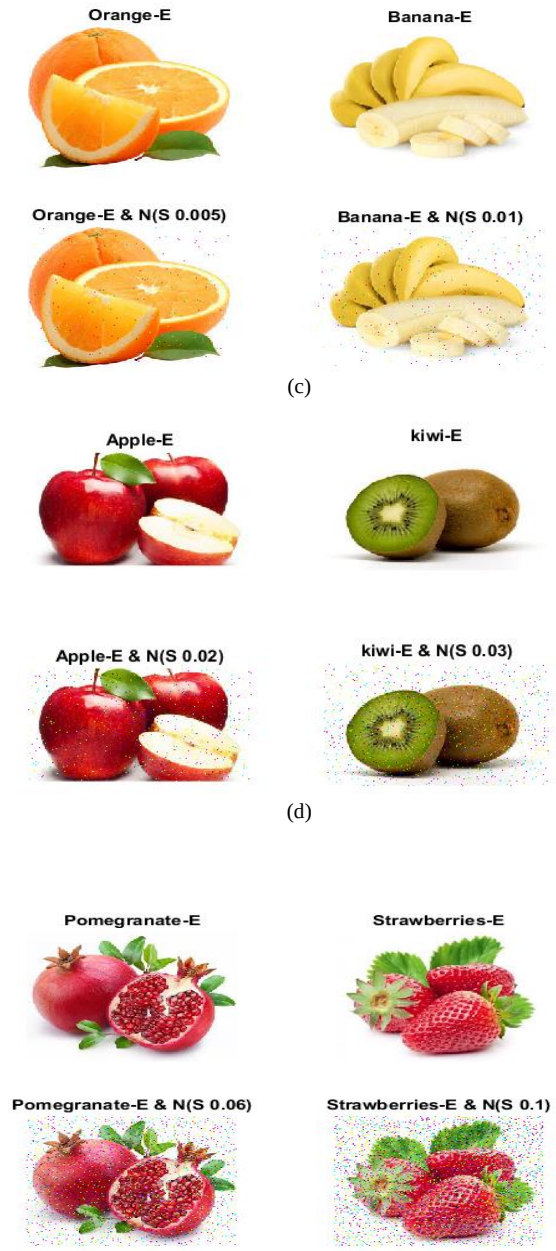
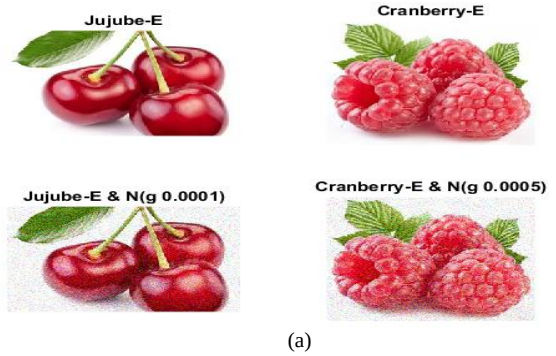


Fig. 16. The visual effect of salt and pepper noise on the images.



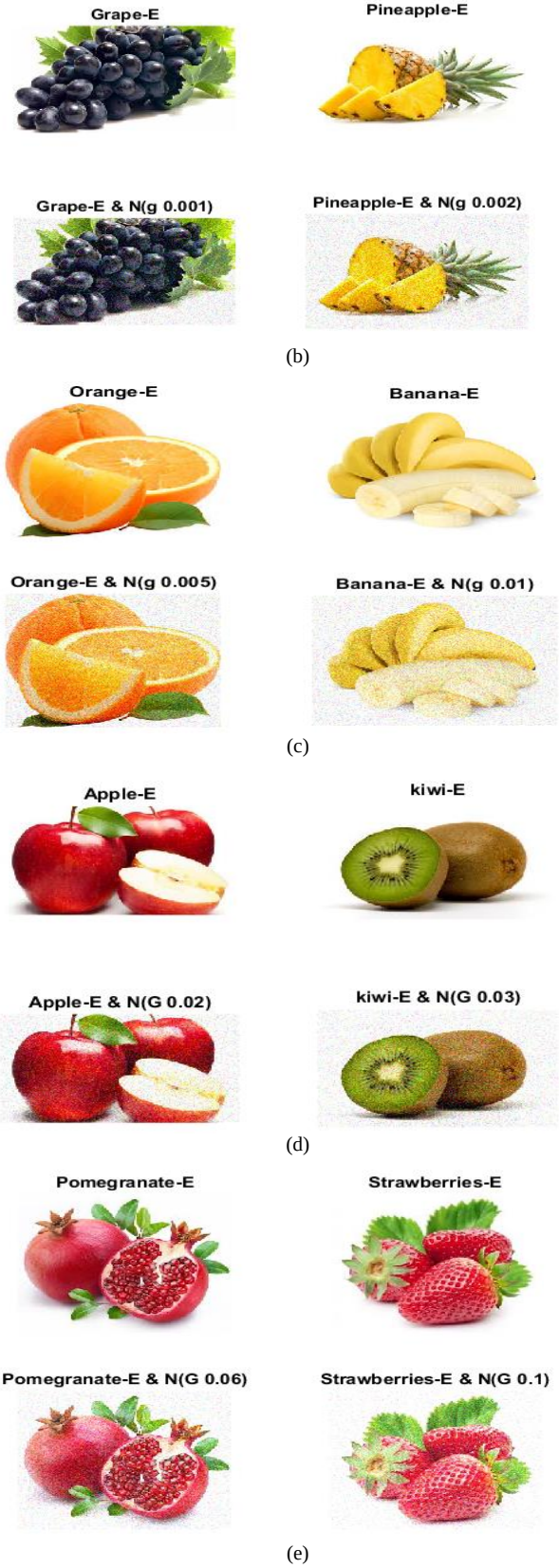
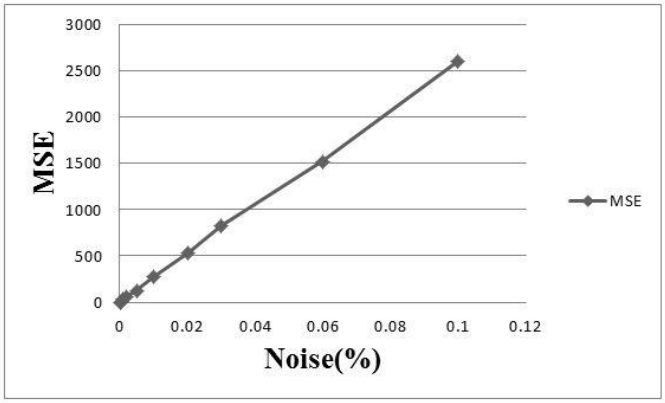
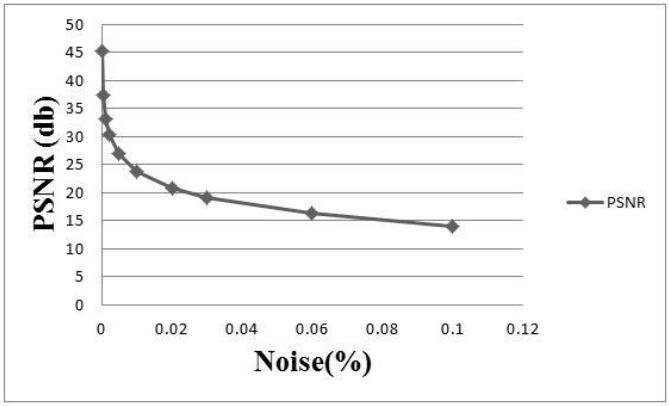


Fig. 17. The visual effect of Gaussian noise on the images.

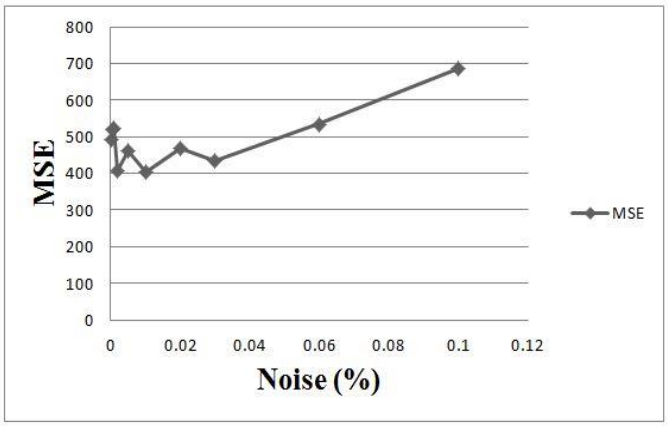


(a)

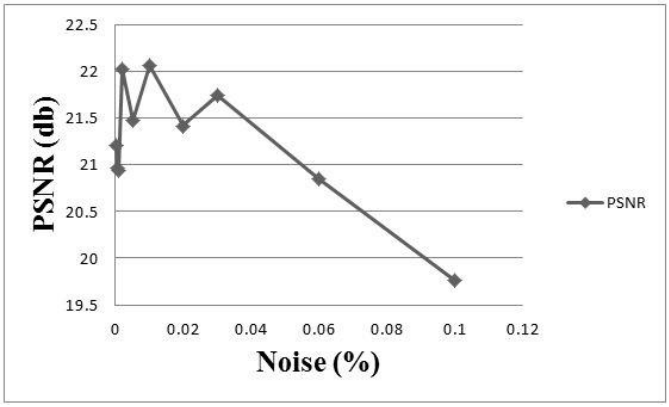


(b)

Fig. 18. The MSE and PSNR for different percentages of salt & pepper noise.



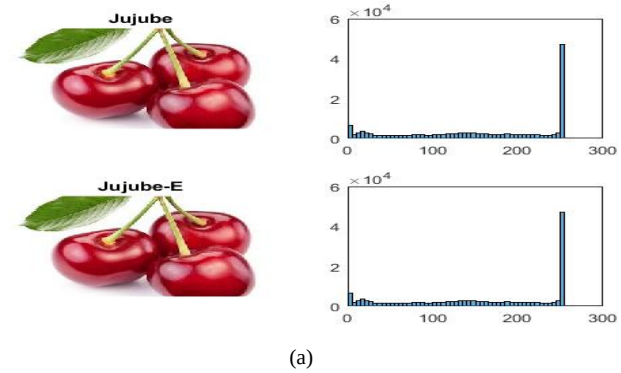
(a)



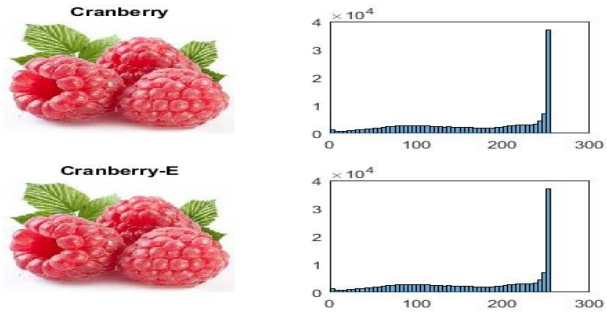
(b)

Fig. 19. The MSE and PSNR for different percentages of Gaussian noise.

To visualize the effect of adding the encrypted secret message of the HSA to the carrier image, the histogram for the images with and without the embedded message were compared and shown in Fig. 20. It was found that no difference was noticed in the histogram. This indicates a good advantage for the hex symbols technique.



(a)



(b)

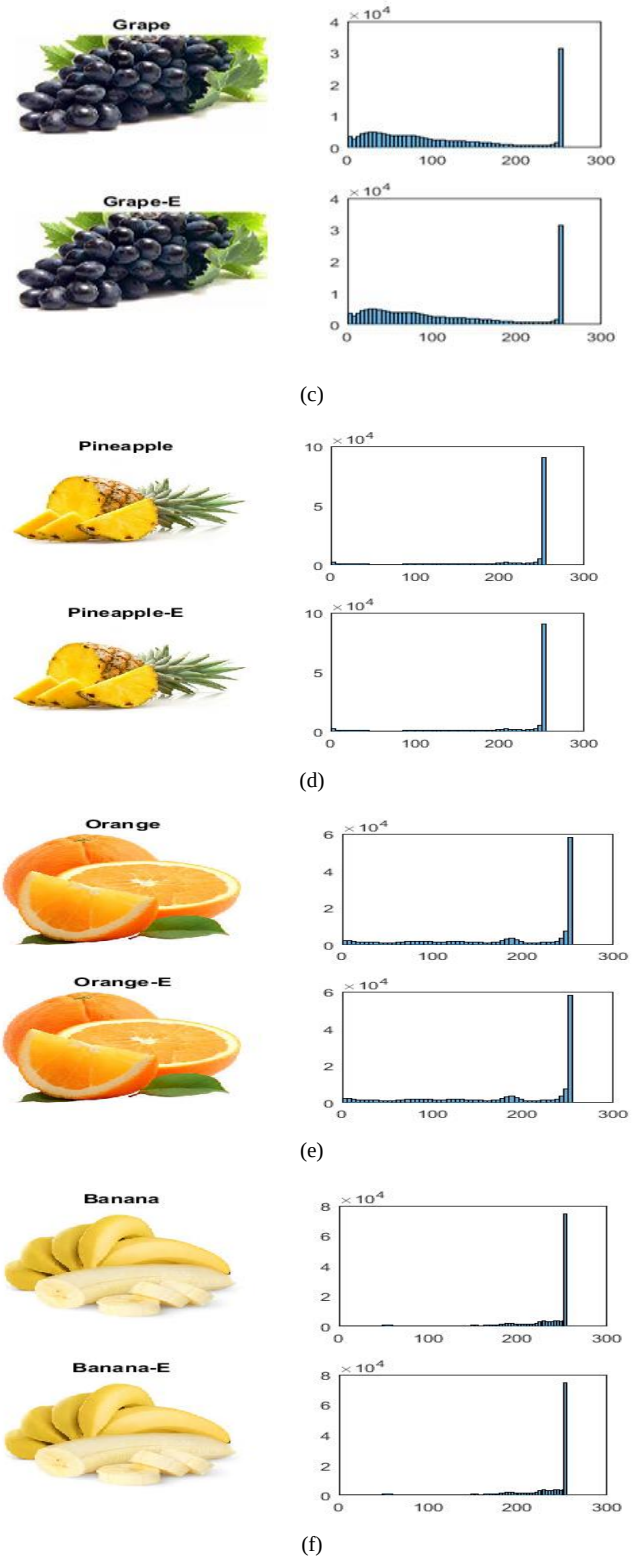


Fig. 20. Histogram comparisons before and after embedding.

VI. Conclusion and Future Work

In this paper we proposed a new variation of steganography approaches in which we combined image steganography with our previously proposed AES Inspired Steganography (IAIS). This approach represents an upgrade of the obstacles' level against potential external breaches. Hence, this will allow for increasing the information security while transferring secret information through the internet or private networks.

Future developments could be conducted to increase the complexity and capacity of the steganography approach allowing handling larger amounts of secret information. Furthermore, the approach could be incorporated in different areas of application including medical information and records security.

References

[1] T. Morkel, J. Eloff and M. Olivier, "An Overview of Image Steganography," *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sandton, South Africa, June/July 2005 (Published electronically) .

[2] S. Abu Asbeh, S. Hammoudeh and A. Hammoudeh, "AES Inspired Hex Symbols Steganography for Anti-Forensic Artifacts on Android Devices", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 5, 2016.

[3] S. Abu Asbeh, H. Al-Sewadi, S. Hammoudeh and A. Hammoudeh, "Hex Symbols Algorithm for Anti-Forensic Artifacts On Android Devices," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 4, 2016.

[4] E. Barker, "Recommendation for Key Management - Part 1: General", 4th Rev., *NIST Special Publication 800-57*, 2016.

[5] S. Galbraith, *Mathematics of public key cryptography*, 1st ed. Cambridge: Cambridge University Press, 2012.

[6] W. Stalling, *Computer Organization and Architecture; Designing for Performance*. 8th Ed., Prentice Hall, 2010.

[7] W. Stallings. *Cryptography and Network Security: Principles and Practices.* 5th Ed, Prentice Hall, 2010.

[8] A. Jain, and G. Chhabra, "Anti-Forensics Techniques: An Analytical Review", *IEEE Seventh International Conference on Contemporary Computing (IC3)*, pp. 412-418, 7-9 Aug. 2014.

[9] A. Swathi and S. Jilani, "Video Steganography by LSB Substitution Using Different Polynomial Equations", *International Journal Of Computational Engineering Research*, vol. 2, no. 5, pp. 1620-1623, 2012.

[10] K. Dasgupta, J. 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, 2012.

[11] 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 (IAJIT)*, vol. 7, no. 1, pp.34-38, 2010.