

3-LAYER PC TEXT SECURITY VIA COMBINING COMPRESSION, AES CRYPTOGRAPHY 2LSB IMAGE STEGANOGRAPHY

Noorah Alanizy¹, Alanood Alanizy¹, Noura Baghoza¹, Manal AlGhamdi², *Adnan Gutub¹

¹Computer Engineering Department, Umm Al-Qura University, Makkah, Saudi Arabia;

²Computer Sciences Department, Umm Al-Qura University, Makkah, Saudi Arabia

*Corresponding Author Email: aagutub@uqu.edu.sa

Abstract

In today's scenario, one of the biggest challenges facing computer users is how to secure data on a personal computer or in any communication. Various forms of data security and hiding algorithms have been developed in the last decade. Cryptography and steganography are known familiar security methods to be used for hide sensitive data. In this paper, we have developed a data hiding method in images with three security layers. The first layer is compression to reduce the redundancy in data representation, i.e. to decrease the storage capacity required for that sensitive data. The second layer is cryptography using the Advance Encryption Standard (AES) to make the compressed data unusable. The third layer hides the data in the 2 least significant bits (LSB) of images presenting high 3-layers security scheme. The results proven interesting outcomes compared to other methods. This 3-layer technique is showing promising research direction for further text security developments to be coming in the future.

Key Words : Security for personal computers, AES cryptography, Image steganography, Hiding text on PC, LSB steganography, Data compression.

1. Introduction

There are several ways to hide sensitive data in a personal computer (PC). Cryptography and steganography are known common security methods to hide sensitive data. Cryptography reorder/replace the data such that it becomes unbeneficial [1]. Steganography hides the data within other media so that the hidden information is invisible to humans [2]. However, there are concerns about how effective these methods are in terms of confidentiality, especially if the data is classified sensitive to the person, such as e-mail messages and credit card information [3]. This made-up the motivation to benefit from both, i.e. combining cryptography and steganography assuming more privacy. In fact, this security combination is assumed to more trusted, confidential and only to be known by PC user [4].

The research proposes utilizing cryptography process of converting plain text into unintelligible text, rendering it unreadable without the secret knowledge [5], merged to steganography as art or practice of concealing messages, images, or files within other multimedia message, image, or file [6].

The objective of this work presents multi-layers of algorithms to hide sensitive data in an image, similar in principle to the strategy presented in [7] but hiding within images instead of videos. This multi-layers proposed research innovation is designed with its first layer to reduce the size of the embedded data, i.e. by compressing the data before encrypting it. Applying the compression algorithm before encrypting the data helps reduce the size

of data to enable faster and secure transmission, which furthermore increase the overall speed of the encryption process. The second layer is cryptography, as the responsible confidentiality layer applied in the system. In this step, we are encrypting the secret plain text and converting it to cipher text using the AES algorithm [8]. Cryptography requires a secret key to be agreed upon between the sender and receiver for the encryption and decryption processes [9].

In this work, the sensitive data passes through the crypto layer after compression to be followed by the steganography layer, as third layer, resulting in the output file as trusted Stego-Image [2]. Fig. 1 shows the proposed 3-layers security system. This proposed 3-layer system is implemented on a software platform using Java programming language. We assumed cryptography layer to apply acceptable security using AES algorithm as well as steganography layer adopting image-based steganography, i.e. hiding the encrypted data in the least significant bits (LSB) of images, similar in theory to previous work presented in [8].

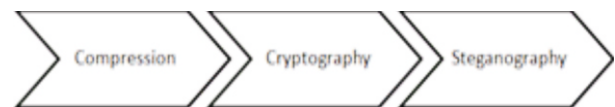


Fig. 1 : Steps of our proposed system

2. Background

There are many methods available that can be used to secure data combining cryptography and steganography using various multimedia contents motivating this research. For example, M. Hussain [6] proposed to hide data bits in an image by changing the LSB of each RGB image pixel's colour byte. The method [6] described storing three bits in each pixel by changing the LSB bit of the red, green, and blue colour components, since every colour is represented by a byte. The research showed a real advantage of using LSB to hide secret data where the change in the pixels had a very low effect, i.e. unnoticed in observation, as also been proven in the work presented in [10]. Each shading RGB cover pixel is of 3 bytes making 24-bits in need for every pixel representation. The 3 bytes for each pixel is to demonstrate a shading quality hiding three bits in each chosen pixel. Assume, for example, we want to hide decimal number 300 in binary form into the three below adjoining pixels (9 bytes).

```
10010101 00001101 11001001
```

```
10010110 00001111 11001011
```

```
10011111 00010000 11001011
```

This secret number 300 represented as 100101100 in binary can be embedded into the LSBs as below:

```
10010101 00001100 11001000
```

```
10010111 00001110 11001011
```

```
10011111 00010000 11001010
```

In this way, here the perception expresses that exclusive 5-bits are changed, comparing to the original pixels bits, out of 9 bits, as typical example of hiding number 300 which is completely affected by the data bits. This LSB steganography is found revisited in different flavours to find benefits for different focuses such as the work of "Pixel Indicator high capacity Technique for RGB image Based Steganography" [11], or "Vibrant Color Image Steganography using Channel Differences and Secret Data Distribution" [12]. The LSB image steganography even found modifications using truth table methods [13] which are considered less secure than normal AES crypto involved systems [14].

Komal Patelet et. al. [15] discussed security method in which the sensitive message is first encrypted by the Blow-fish crypto algorithm and then hidden similarly in a cover file using steganography. The steganography used was also based on the LSB algorithm for both embedding and extraction processes. They used C#.Net language to implement their proposed work in brief description manner. This multi-level security system is found common in previous "Triple-A" work discussed in [16].

Satwinder Singhet et. al. [1] and Nouf Al-Otaibi et. al. [8], both proposed interesting approaches that provide good security to hide PC data. They encrypt the data via AES

cryptography followed by hiding it within digital images using LSB image steganography. Nouf system [8] is clear to analyze since it is designed on visual basic platform. We tested this Nouf system thoroughly and carried out this study focusing on improving it reducing the data hidden capacity proposing this 3-layer PC text security via combining compression, AES cryptography and 2LSB image steganography.

3. The Proposed System

The 3-layer security system for hiding sensitive text data has three layers, i.e. a compression data capacity layer followed by two security layers as illustrated in framework overview of Fig. 2. The crypto layer is using the AES algorithm. We implemented AES utilizing javax.crypto package using a secret key (password) with of 6-character length. In this layer, each character of the secret message is converted into an array of binary bytes providing the result as encrypted ciphertext. Then the ciphertext of this crypto layer is combined to a header message containing information about

colour channels used, number of LSBs for stego layer, and password for encryption. This result of ciphertext and header passes to the stego layer for LSB imbedding process.

In the steganography layer the system involves a PC available RGB image (PNG or BMP) decided by user to convert its pixels into an array of binary bytes. This stego layer preparation can start its process while the crypto layer is running, i.e. preparing the image as binary bits array, but it cannot start hiding data except after the cipher-text is generated from the crypto layer. Each pixel within the RGB image has three channels, namely red, green and blue (RGB), representing a byte of 8 bits each. Therefore, using the least significant bits (LSB) image-based steganography in our original system hides 3 bits in each pixel as will be detailed and improved in the interfaces subsections below.

3.1 System Interfaces

The system interface presented is described by the process showing the platform figures. The program first runs selection main interface shown in Fig. 3. It asks about the mechanism to be chosen by the user, i.e. "Encryption and Hide" or "Decryption and Show".

If the choice is "Encryption and Hide", then the process will conduct the following procedure:

1. Choose an image to make it a stego cover.
2. Choose the location to save the stego image.
3. Enter secret data as plain text (Fig. 4).
4. Check the "encrypt" checkbox to encrypt the message.
5. Click next.

6. The interface has a “view” button to show the selected image (Fig. 5).
7. Choose the colour channels “R, G or B” to hide the data on them with “B” as default.
8. Choose the testing number of LSB “1 to 8” to hide in each channel with “1” as default.
9. Detailed information will appear in the same window.
10. Click “Merge text with image” as the process is shown in Fig. 6.

- If false password is entered, an error (invalid message) is shown (Fig. 9).
5. It can save the secret message as a text file (Fig. 10).
 6. The interface has a “view” button to show the selected image (Fig. 11)

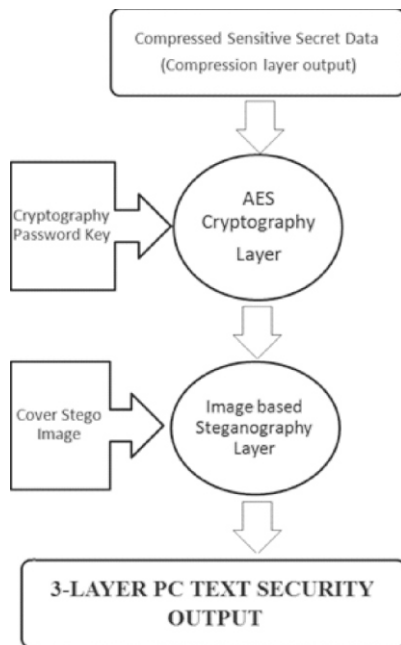


Fig. 2 : Security system crypto - stego layers



Fig. 4 : Stego (encode) interface



Fig. 3 : Main interface

If the choice of main system interface as Fig. 3 is selected "Decryption and show", then, the process will be as follows:

1. Choose a stego cover image (Fig. 7).
2. Click “Decryption and show” button.
3. Popup window appear; ask to enter password to decrypt the secret data (Fig. 8).
4. If password entered correctly, secret message will appear in the text area.



Fig. 5 : Image shown when icon “view” is clicked

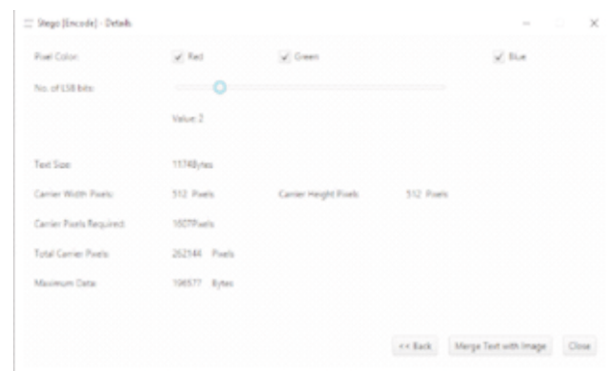


Fig. 6 : System interface showing detailed information of the encoding process

4. Testing and Verification

The research studied images comparing them before and after steganography. It involved several images for the testing and verification process providing very similar results. The comparison of all original images vs. stego-images is remarking that hiding data in a stego image can work fine only using 1LSB and 2LSB as the stego image appears the same as the original image, i.e. providing acceptable security. This issue of using more LSBs is elaborated in the following subsection relating it to capacity improvement.

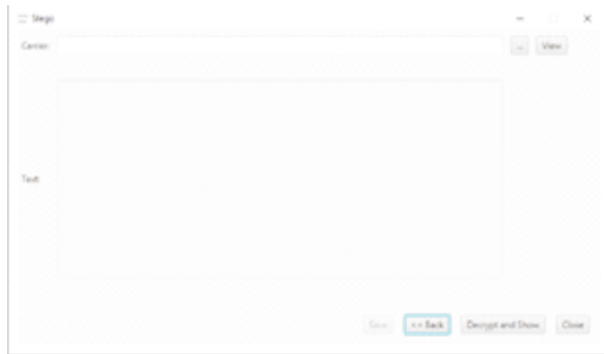


Fig. 7 : Decryption showing interface: “Decoding”

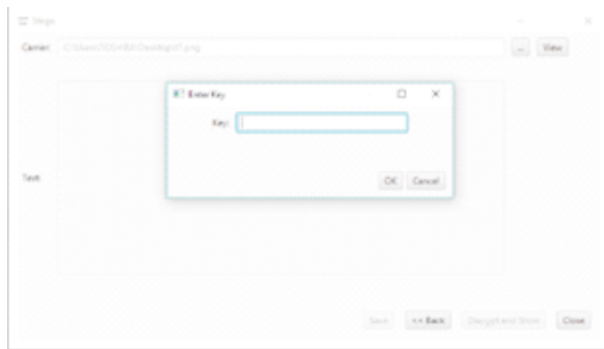


Fig. 8 : Password 'key' popup window

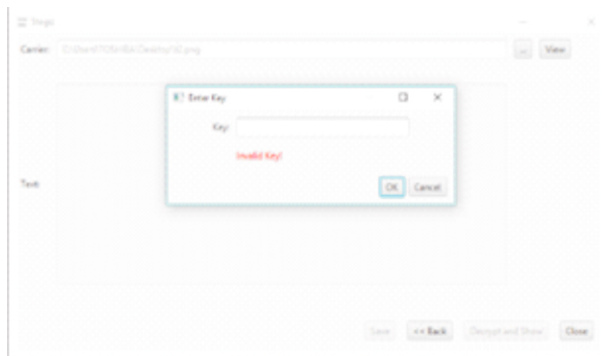


Fig. 9 : Wrong 'invalid' password entered

4.1 Capacity vs. Security

To improve capacity of the proposed security system multi-bits steganography is used. However, increasing the LSBs is degrading the system security. Fig. 12 presents

seven images with data hiding in 1,2,3... 7 LSBs. We observe that when hiding data in 1 and 2 LSBs, the changes are not noticeable providing acceptable security. But when hiding data in 3,4,5,6, and 7 LSBs, the image distortion appears rendering the main objective of the system.

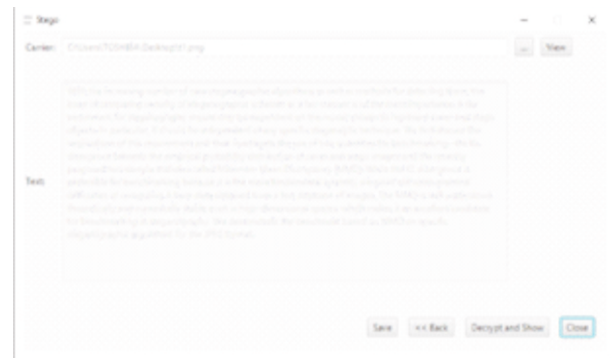


Fig. 10 : Decryption interface

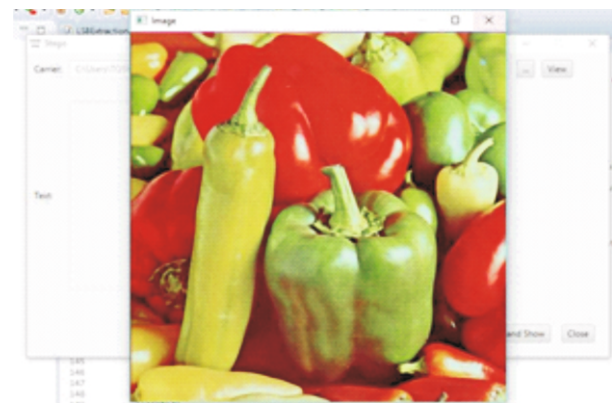


Fig. 11 : Image shown when “view” button is clicked

The study also examined hiding small amount of data in a critical comparison resulting negligible image distortion. This lead to the recommendation of trying storing data in large-sized images that contains more pixels to avoid noticeable distortion. Fig. 13 shows a sample of three testing images, which are stego covers that hide data in 1LSB and 2LSB proving the concept of acceptable security adopting 2LSB steganography. This suggestion of using 1LSB as well as 2LSB is analyzed independently in the following subsection.

The research work further tested several different PC images with the same extensions of PNG assuming

different numbers of pixels to be used as cover images, i.e. for the stego layer shown in Fig. 14. We found that when hiding the sensitive text in 1LSB and 2LSB, the stego image does not have any noticeable distortion on it as pretended also within others work [17]. Moreover, the two Least Signification Bit (2LSB) steganography increased the capacity of hiding information double the usage of 1LSB with acceptable security, as mentioned in paper [18].

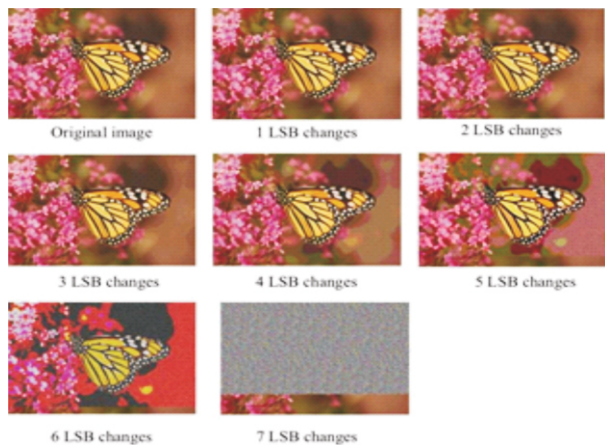


Fig. 12 : Original image followed by images with data hiding in 1, 2, 3, 4, 5, 6, 7 LSBs



Fig. 13 : Original image followed by stego-images hiding data in 1LSB and 2LSB

4.1 Considerations and Remarks

Considering the future work suggested by paper [8], which suffers language flexibility, it suggests making the system to support different languages, i.e. specified for Arabic language. It was found that the original work was implemented using the Visual Basic programming [8], which do not support Arabic string Unicode texts directly. A number of adjustments steps are required to solve this problem. In our improved implementation, we modeled this 3-layer security system using Java platform, which supports Arabic string Unicode, CharsetDecoder and CharsetEncoder, directly. Notes that Java programming uses Unicode as native encoding, so any text will be converted to Unicode for proper handling. Java already supports almost all known encodings [19], making it the practical choice to be selected for language consideration.

The paper [8] also suggested testing its 2-layer system by using other cryptographic symmetric algorithm for possible improvements. Through research [20], AES is currently found to be the most efficient crypto algorithm in security features, as shown in Table 1.



Fig. 14 : Different PC images used for testing as cover images for the stego layer.

Table 1

Comparison of Symmetric Cryptography Algorithms [20]

Algorithms	Blow Fish	AES	3DES	DES
Key size (bits)	32-448	128, 192, 256	112 or 118	64
Block size (bits)	64	128	64	64
Round	16	10, 12, 14	84	16
Structure	Feistel	Substitution Permutation	Feistel	Feistel
Flexible	Yes	Yes	Yes	No
Features	Secure enough	Excellent Security	Adequate security Replacement for DES,	Not structure, Enough
Speed	fast	fast	Very slow	slow

The paper [8] suggested studying the capacity and security for practical applications. Therefore, we involved addition of a pre-security layer to the system, namely a compression layer, before the cryptographic – steganography layers. Data compression in cryptography is believed important in reducing the number of bits for text that would be hidden in images. Compression data can save storage capacity contribution in speed up of the encryption process.

Text compression can be described as removing all unneeded characters, inserting a single repeat character to indicate a string of repeated characters and substituting a smaller bit string for a frequently occurring bit string. Compression is often compared to data deduplication, but the two techniques operate differently. Deduplication is a type of compression that looks for redundant chunks of data across a storage or file system and then replaces each duplicate chunk with a pointer to the original. We use the

GZIP compression technique which is based on the Deflate algorithm. Deflate is a lossless data compression algorithm and associated file format that uses a combination of the LZ77 algorithm and Huffman coding supported fully by our Java Platform [19].

4.1 Comparisons and Analysis

In our proposed approach, we use the compression followed by security encryption and stego techniques. The compression and security are performed in sequence processes. In fact, we notice that these procedures consume less execution time compared to independent encryption/decryption and stego techniques as shown in the tables below. In general, joint compression and security algorithms are more efficient than independent algorithms. Since the encryption is done after compression, these algorithms provide higher processing speeds.

Security: In the combined compression, encryption and stego techniques, the compression process includes one or more encryption steps. In addition, a separate encryption process is tested after compression. As a result, the combined compression and encryption technique provides same levels of security when compared to the independent compression, encryption, stego techniques.

Performance: Using compression combined to security encryption and stego algorithms is more effective than separating all algorithms. This is because when the encryption is performed after compression, some unnecessary interfacing routines are found slowing the overall process. This made-up the confidence of providing high security speed via combining the 3-layers of compression, cryptography and steganography, as detailed in Table 2 for the different timings of hiding process and Table 3 for the different timings of retrieving process. In other words, our study has shown that the joint procedures of compression followed by encryption and steganography algorithms can increase the system overall speed more than independently running the algorithms. This fact of this performance issue is getting more clear as the size of data increase in real life applications.

5. Conclusions

This study presented 3-layer PC text security via combining compression, AES cryptography and 2LSB image steganography in a Java platform system. The testing involved compression to be performed on the sensitive data to be secured to increase the capacity. After compression, the data is processed to cryptography followed by steganography. The system is implemented intentionally on a Java platform to benefit from multi-language flexibility using Eclipse IDE. The study considered AES cryptography since its found the practical symmetric crypto procedure. For image steganography, we have used the simplest yet most effective method

known as 2LSB image steganography algorithm which is found appropriate compared to different higher LSB stego attempts.

Table 2
Execution Time in Hiding Process

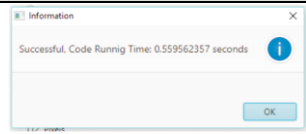
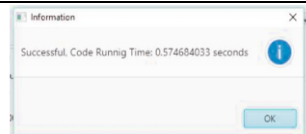
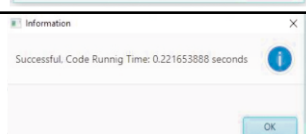

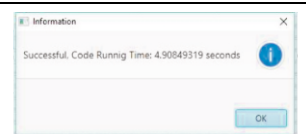
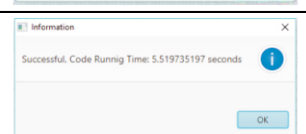
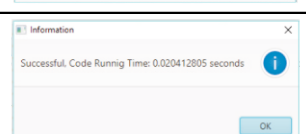
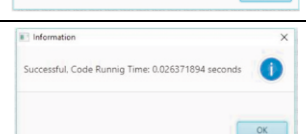
Screenshot for Analysis and Verification	Security	Compression	Execution time (seconds)
	Yes	Yes	0.5596
	Yes	No	0.5749
	No	Yes	0.2217
	No	No	0.2307

Table 3
Execution Time in Retrieving Process

Screenshot for Analysis and Verification	Security	Compression	Execution time (seconds)
	Yes	Yes	4,9085
	Yes	No	5.5197
	No	Yes	0.0204
	No	No	0.0264

The elaboration also remarked on building the system with a combined Java platform interface which gave efficient performance compared to separating the layers showing attractive results and opening the research for further improvements in terms of compression, crypto, and stego algorithms to be used.

Acknowledgment

Authors would like to thank Umm Al-Qura University (UQU) for hosting this research. Special thanks to the cooperation between the two departments via Prof. Adnan Gutub from Computer Engineering and Dr Manal AlGhamdi from Computer Sciences for motivating this research.

References

- [1] Satwinder Singh and Varinder Kaur Attri, "Dual Layer Security of data using LSB Image Steganography Method and AES Encryption Algorithm", Vol. 8, No. 5, pp. 259-266, 2015.
- [2] Nouf Alotaibi, Adnan Gutub, and Esam Khan, "Stego-System for Hiding Text in Images of Personal Computers", The 12th Learning and Technology Conference: Wearable Tech / Wearable Learning, Effat University, Jeddah, Kingdom of Saudi Arabia, 12-13 April 2015.
- [3] Adnan Gutub, Manal Fattani, "A Novel Arabic Text Steganography Method Using Letter Points and Extensions", International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol. 1, No. 3, pp. 502-505, published by World Academy of Science, Engineering and Technology, 2007.
- [4] Nouf Al-Otaibi and Adnan Gutub, "Flexible Stego-System for Hiding Text in Images of Personal Computers Based on User Security Priority", Proceedings of 2014 International Conference on Advanced Engineering Technologies (AET-2014), pp. 250-256, Dubai UAE, 25-26 December 2014.
- [5] Norah AlAssaf, Basem AlKazemi, Adnan Gutub, "Applicable Light-Weight Cryptography to Secure Medical Data in IoT Systems", Journal of Research in Engineering and Applied Sciences (JREAS), Vol. 2, No. 2, pp. 50-58, April 2017.
- [6] M. Hussain and M. Hussain, "A survey of image steganography techniques", International Journal of Advanced Science and Technology, Vol. 54, pp. 113-124, May 2013.
- [7] Nouf Al-Juaid, Adnan Gutub, Esam Khan, "Enhancing PC Data Security via Combining RSA Cryptography and Video Based Steganography", Journal of Information Security and Cybercrimes Research (JISCR), Vol. 1, No. 1, Published by Naif Arab University for Security Sciences (NAUSS), June 2018.
- [8] Nouf Al-Otaibi and Adnan Gutub, "2-Layer Security System for Hiding Sensitive Text Data on Personal Computers", Lecture Notes on Information Theory, Vol. 2, No. 2, 2014.
- [9] Adnan Gutub and Esam Khan, "Using Subthreshold SRAM to Design Low-Power Crypto Hardware", International Journal of New Computer Architectures and their Applications (IJNCAA), Vol.1, No.2, pp. 474-483, 2011.
- [10] Mohammad Tanvir Parvez and Adnan Gutub, "RGB Intensity Based Variable-Bits Image Steganography", APSCC 2008 – Proceedings of 3rd IEEE Asia-Pacific Services Computing Conference, Yilan, Taiwan, 9-12 December 2008.
- [11] Adnan Gutub, Mahmoud Ankeer, Muhammad Abu-Ghalioun, Abdulrahman Shaheen, and Aleem Alvi, "Pixel Indicator high capacity Technique for RGB image Based Steganography", WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications, University of Sharjah, U.A.E. 18 – 20 MARCH 2008.
- [12] Mohammad Tanvir Parvez and Adnan Gutub, "Vibrant Color Image Steganography using Channel Differences and Secret Data Distribution", Kuwait Journal of Science and Engineering (KJSE), Vol. 38, No. 1B, pp. 127-142, June 2011.
- [13] Walaa Abu-Marie, Adnan Gutub, and Hussein Abu-Mansour, "Image Based Steganography Using Truth Table Based and Determinate Array on RGB Indicator", International Journal of Signal and Image Processing (IJSIP), Vol. 1, No. 3, pp. 196-204, May 2010.
- [14] Farhan Khan and Adnan Gutub, "Message Concealment Techniques using Image based Steganography", The 4th IEEE GCC Conference and Exhibition, Gulf International Convention Centre, Manamah, Bahrain, 11-14 November 2007.
- [15] Komal Patel, Sumit Utareja, Hitesh Gupta, "Information Hiding using Least Significant Bit Steganography and Blowfish Algorithm", Vol. 63, No. 13, February 2013.
- [16] Adnan Gutub, Ayed Al-Qahtani, and Abdulaziz Tabakh, "Triple-A: Secure RGB Image Steganography Based on Randomization", AICCSA-2009 - The 7th ACS/IEEE International Conference on Computer Systems and Applications, pp. 400-403, Rabat, Morocco, 10-13 May 2009.
- [17] Adnan Gutub, Nouf Al-Juaid, "Multi-Bits Stego-System For Hiding Text in Multimedia Images Based on User Security Priority", Journal of Computer Hardware Engineering, Vol. 1, No. 2, DOI: 10.63019/jche.v1i2.513, EnPress Publisher, 2018.
- [18] Adnan Gutub, "Pixel Indicator Technique for RGB Image Steganography", Journal of Emerging Technologies in Web Intelligence (JETWI), Vol. 2, No. 1, Pages: 56-64, February 2010.
- [19] Network Dictionary. Saratoga CA: Javvin Technologies, 2007.
- [20] Sohrabi, M. K., & Ghods, V., "A comparison of symmetric key algorithms DES, AES, Blowfish, RC4, RC6: A survey". Journal of Computers, Vol. 11, No. 2, pp. 140-148, 2016.