Jurnal Teknologi

Full paper

Enhancing Secured Data Hiding Using Dynamic Digital Signature for Authentication Purpose

Erfaneh Noroozi[a*], Salwani Mohd Daud[a], Ali Sabouhi[b]

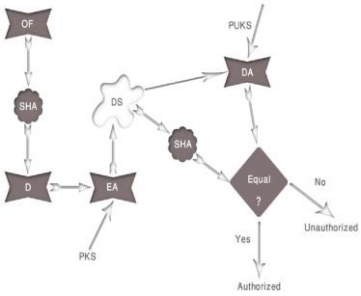
[a]*Advanced Informatics School (AIS), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia*
[b]*Software Engineering of Computer Science Kuala Lumpur, Malaysia*

*Corresponding author: nerfaneh2@live.utm.my

Article history

Received :1 December 2013
Received in revised form :
10 January 2014
Accepted :31 January 2014

Graphical abstract



Abstract

As a significant verification method, digital signature algorithm introduces a technique to endorse what the contents of the message. This message has not been altered throughout the communication process. Due to this, it increases the receiver confidence that the message was unchanged. However, two issues that required to be addressed are large size of the ciphered data in digital signature and making it closer to the original file. The objective of this paper is to present the adoption of digital signature as a stegano-image into the main image and the LSB steganographic method is capable to increase the security. The benefits of this encryption algorithm are computational efficiency, digital signature with the size as small as 8 bytes and minimize bandwidth in comparison with other digital signature methods. In messages with the sizes smaller than 1600 bytes, the hashed file reduces the original file up to 8.51%.

Keywords: Digital signature; hashed message; embedded image; image steganography

Abstrak

Sebagai kaedah pengesahan yang ketara, algoritma tandatangan digital memperkenalkan teknik untuk menyokong kandungan mesej. Mesej ini tidak diubah sepanjang proses komunikasi. Dengan itu, ia meningkatkan keyakinan penerima bahawa mesej tidak berubah. Walau bagaimanapun, terdapat dua isu yang perlu ditangani iaitu saiz data rahsia yang lebih besar dalam tandatangan digital dan untuk menjadikannya lebih hampir dengan fail asal. Objektif kertas kerja ini ialah untuk mengemukakan penggunaan tandatangan digital sebagai imej-stegano ke dalam imej utama dan kaedah LSB steganografi mampu untuk meningkatkan keselamatan. Manfaat algoritma penyulitan ini adalah kecekapan komputan, tandatangan digital dengan saiz sekecil 8 bait dan meminimumkan lebar jalur berbanding dengan kaedah tandatangan digital yang lain. Dalam mesej dengan saiz yang lebih kecil daripada 1600 bait, fail *hash* mengurangkan fail asal sehingga 8.51%.

Kata kunci: Tandatangan digital; mesej hash; imej terbenam; imej steganografi

1.0 INTRODUCTION

The security of information passed over an open channel has become a fundamental issue and therefore, the confidentiality and data integrity are required to protect against unauthorized access and usage. One hides the existence of the message and the other distorts the message itself. In Cryptography, the form of the data becomes complicated and encoded data are transmitted. In steganography, the data is embedded in an image file that is transmitted. By encoding information and using fewer bytes, we compress data and do source coding and bit-rate reduction. Compression of data means deleting unimportant information and decreasing the size of the data file. It contributes to the reduction of the resources and capacity of the storage and transmission.

2.0 DIGITAL SIGNATURE

The concept of identity-based cryptosystem (IBC) was used by Shamir and its main concept is the calculation of the public key of a user (Shamir, 1985). When the user registers in the network, he identifies himself by his email and IP address, telephone number and place and a private key is devoted to him by the private key generator (PKG). This is an identity-based signature (IBS) scheme (Fiat and Shamir, 1986). The verifier uses PKG's public key and the signer's identity.

This is called identity-based cryptography (IBC) that contributes to the problem of verification and brings about the identity-based multi-signature scheme (IBMS). Since these two cryptosystems use two keys, they come under the public key infrastructure (PKI) (Hess, 2002). By saying traditional public key cryptography and certificate based public key cryptosystem

(CB-PKC) and identity based cryptosystem (IB-PKC), we can distinguish between them (Sakai *et al.*, 2000).

A trapdoor as a public-key encryption system consists of a public key and the trapdoor that serves as a private key. Encrypting and decrypting is forward and reverse direction calculation.

Trapdoor permutation is comparable to a digital signature and signature verification matches with backward calculation with the private key and the calculation of forward direction. So, public key cryptosystem is considered as the basis of digital signature and signing and verification correspond decryption and encryption (Figure 1).

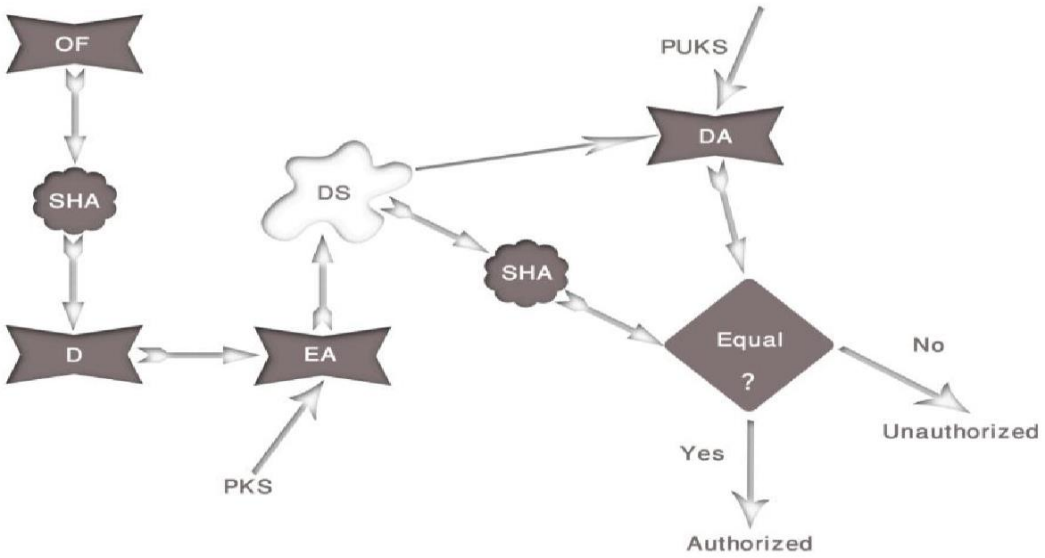


Figure 1 Generation and verification of digital signature

3.0 SYMMETRIC AND PUBLIC KEY CRYPTOGRAPHY

Most people assume that public key technology is used by public key infrastructures. Both public key and symmetric key algorithms are used by the functions in windows implementation. Symmetric key encryption consists of using a single key to encode and decode data in public key encryption as shown in Figure 2. A public key is sent with the message and the private key which the receiver has remains secure without being transmitted in the network and it is very slow, so it is very difficult to send a large bulk of data using it.

Attacks are avoided through the following procedure; message m is operated by the encoded hash, to which the RSA algorithm is added (Kang, 2003). After RSA, other digital signatures like Lamport signatures, Merkle signatures (Merkle, 1979) and Rabin signatures followed. In relation to the need for security in digital signature, Shafi Goldwasser, Silvio Micali, and Ronald Rivest explained attack models (Bellare and Don, 2009). When doing a key-only attack, the attacker can just reach to a public verification key. He knows about the messages, but cannot choose between them in an attack. The messages are chosen randomly for reaching the signature (Keromytis, 2010).

If this signature system is being used directly, there is some concern about the fake attack. The attacker randomly chooses a signature σ and decodes the corresponding message verification process. Nevertheless, this signature is not directly manipulated, and the message about to be signed is initially hashed to practically create a short digit to be signed later. This type of forgery attack solely fulfills the hash function corresponding to σ. The message that brings about the values is not produced like this. The signature of hash-and-decrypt is viewed even if we do it against a chosen-message attack in the random oracle model (Atkin, 2010).

Forgery attacks can forge a message signing key. The message signature is determined by attacker in chosen forgery attacks, but have not been formerly recognized by him in an existential forgery. GMR signatures are a way to point an attack to the message even in existential forgery. What is common in initial signature schemes is that they all have trapdoor permutation, like the RSA function or the composite n of square modulo (for Rabin signature). A parameter that can be calculated practically in the forward direction but not in the backward characterizes the trapdoor permutation kind. For backward direction calculation, another trapdoor is considered for the parameter, although there are other ways to calculate digital signature. If the signature system is used directly, the treat of forgery attack is conceivable. This is done by the attacker by choosing a signature σ and decoding the message by verification process. For creating a short digit to be signed, the message is hashed. The signature of hash-and-decrypt is viewed even when it is performed against a chosen-message attack in random oracle model.

The techniques in hash algorithm can be called md5, sha-1 and sha-2 (Jarvinen *et al.*, 2005). Ray West introduced MD5 algorithm for calculation of hashed message. Messages with any length are received by this algorithm and will be divided to a 512 piece is added and the messages with more length would be added at the end of the last piece and since the last piece must be 448 bits added to the end of this algorithm, one bit of (1) and zero are added to reach this size. In MD5, for making the block of data, four kind of 32 bit with a different repetition of a 64 step disorder the data and an operation like and, or, not and xor is used for an effective performance of software with hardware. Sha1 is presented for public use by the standard and modern technology of the USA and the messages with any length in the format of 512 bits will be organized and processed by it (Coleman, 2011).

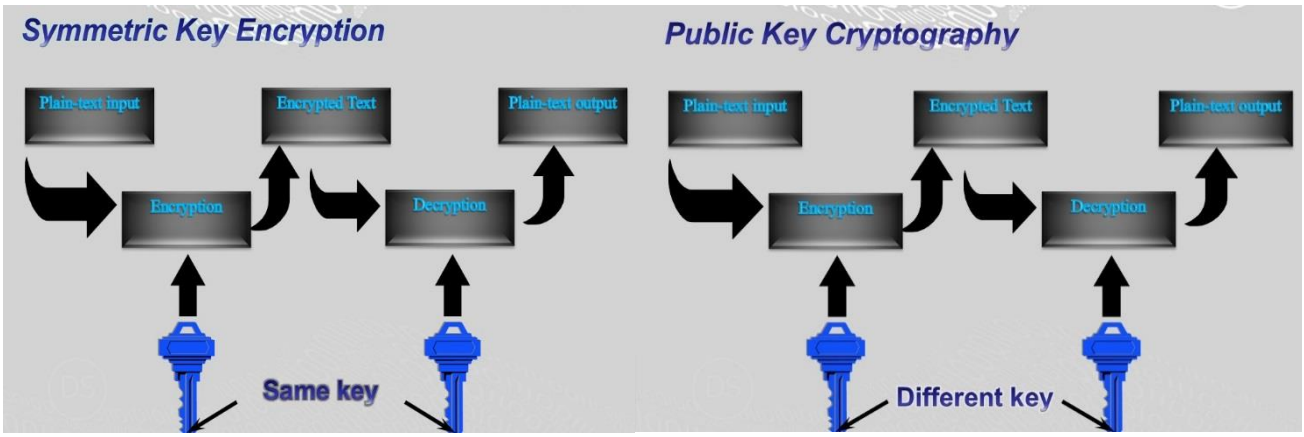


Figure 2 Comparing symmetric key and public key cryptography

■4.0 PROPOSED ALGORITHM

Digital signature usually encodes the signature in a file separated from the original image, thus require extra bandwidth to transmit it (Serret and Gilles, 2012). Scheme extracts the signature from the original image and embeds them into the image, avoiding additional signature file. The digital signature authentication scheme is a kind of sender-receiver protocol (Jansirani, 2011); the sender generates the signature and inserts it into the original image as watermark. In the receiver's side, the ownership and integrity is verified by comparing the signature and watermark both extracted. A digital signature is based upon the idea of public key encryption. The overall scheme in proposing a new method to generate an encrypted hash message, consists of three steps with the following details.

In this algorithm by doing operation on first 100 bytes, the first byte is read by fetch function. On the condition that, the rate of the loaded bytes is zero, we will change it to 1. So that these loaded bytes will be set in 32 different bits and the second byte from 100 byte block. On the condition that this is the last byte, the result will be put in 32 bits and changed in hexadecimal and displayed and if it is not the last one, the result will be multiplied in 32 different bits and with the result, less than 32 bits, from the left side, zero will be added to fill dates in that 4 bytes. If the result is more than 4 bytes, it puts lower 32 bits and we do it till the last byte.

When reading, the encoder starts the process of encoding the hashing file with opening it with (RB) that are the abbreviation of read only and binary. The next step is called "wt" which stand for write and text. Considering a 16 bytes private key and xor as the first byte of hashing file with first private key, the result will be put in the first key byte. This iteration will be continued to the end of the hashed file between private key and hashed message, because the hashed file should be matched to the original message. What results from this operation is a 16 byte key that is attached at the end of message and kept in a character array. At last, we change the encoded output to hexadecimal.

Placing embedded data in the original file is a kind of saving in sending data as shown in Figure 3. For this reason, operational process cannot be conducted on 3 bits; definitely the operation must be performed on one byte. In this case, we need an algorithm that can modify the original file, with 5 bits, in the left side is equal to '0' (00000 + data), and 3 bits in the least significant bit is the embedded data. There are a few functions needed to implement the algorithm. F binary, converts one byte of input data into one string. Another function named mid, converts one string and provides the address of this string, and number of bytes is given in the output. And Fdec converts one string into a decimal number between 0-255.

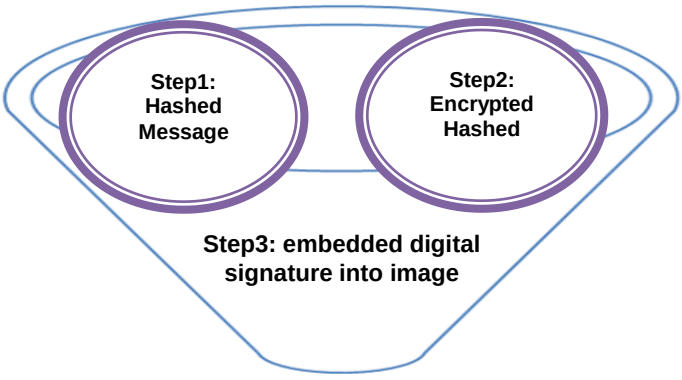


Figure 3 Process of embedding digital signature into the image

4.1 Findings and Results

This paper discusses about the initial findings of the algorithm and compared the output of the proposed algorithm with output of previous algorithm. The findings obtained when few algorithms were compared with the proposed algorithm in terms of size of image and compression of image involved is shown in Table 1.

An image I, contains a collection of pixels, color (R, G and B) occupies one byte (8 bits), and a collection of these three colors

forms a pixel. When you select one of these, the colors are displayed in the following way.

$$\begin{aligned} &R \quad 1111\ 1111 \equiv (255)_{10} \\ &G \quad 0000\ 0000 \equiv (0)_{10} \\ &B \quad 0000\ 0000 \equiv (0)_{10} \end{aligned}$$

If we display (1111 1110) instead of red color (1111 1111), there won't be any change in color, because it is not detectable due to limited powers of the human visual system. We should obtain more empty space as much as possible in order to locate embedded data. For instance, if embedded data is located in the final bit output, it will be represented as follows:

$$\begin{aligned} &1111\ 1111 \equiv (255)_{10} \\ &1111\ 1110 \equiv (254)_{10} \\ &1111\ 1100 \equiv (252)_{10} \\ &1111\ 1000 \equiv (248)_{10} \end{aligned}$$

As a result, instead of each byte or each bit of the original data, you can consider placing 5 bits in the original file and the remaining 3 bits as embedded data. Consequently instead of each byte of R, G and B, these operations are performed.

$$\begin{aligned} &(R\ \text{shR}\ 3)\text{shL}\ 3 \rightarrow R' \\ &(G\ \text{shR}\ 3)\text{shL}\ 3 \rightarrow G' \\ &(B\ \text{shR}\ 3)\text{shL}\ 3 \rightarrow B' \end{aligned}$$

The obtained output, for each byte of original images, 3 bits on the right side are equal to zero (xxxxx 000). In other words, from the mathematical point of view, for every byte, it can be done by dividing into 8 and multiply by 8 as follows:

$$\begin{aligned} &\lceil R/8 \rceil * 8 \rightarrow R' \\ &\lceil G/8 \rceil * 8 \rightarrow G' \\ &\lceil B/8 \rceil * 8 \rightarrow B' \end{aligned}$$

For a real example, if we save an image by paint screen software for desktop and saved it to .bmp format, and this image are considered as I and dimensions m × n with 1024 × 768, each image pixel is equivalent to 24 bits.

$$\begin{aligned} &1024 \times 768 \times 24 = 18874368 \text{ bits} \\ &1024 \times 768 \times 15 = 11796480 \text{ bits} \\ &\Rightarrow \quad 864 \text{ KB} \end{aligned}$$

It should be transformed into an image with the same dimension of m × n and each pixel of it will be equivalent to15 bits of the original data. Instead of data transferring with capacity m × n × 24, data are transmitted with capacity of m × n × 15, and the remaining bits will be regarded as embedded data. As a result, the proportion equal 3 to 8 bits will be equivalent to 37.5%. As a result it can be hide around 37.5% of the image size (Table 1).

Table 1 Summarizes the comparison of research works previously done of size and compression of image

Researchers	Algorithm/method	Size of image	Compression of image
(Zaidan *et al*., 2009)	Tried to test the largest amount of data that might be hidden in the image using pure Steganography.	The result obtained was 50% from the size of the images.	Implemented a solution to solve the problem of simple texture by filtering the images into complex and simple Texture.
(Naji *et al.*, 2009)	Implement a frame work to secure the hidden data within the video file, both LSB algorithm and AES Algorithm.	Not Available	Implemented over the MPEG video to ensure the robustness and was 50% compression of the data.
(Naji *et al.*, 2009)	Implemented multi-cover steganography using remote sensing image and general recursion neural cryptosystem.	Used a non-standard method to secure the data before hiding it, moreover, they create a multi-cover technique to ensure the robustness of their approach.	Not Available
(Kae-por, 2008)	Combined three steganography algorithms on GIF image.	Implementing encoded algorithm which hides around 33% by using PKI.	Not Available
(Hmood, Kasirun *et al*., 2010)	Using pure steganography and human vision system property (HVS).	Images that do not include any simple texture can hide up to 50% of the image size.	Not Available
Proposed Algorithm	Implement an efficient and robust stenographic technique which can avoid various image attacks.	Implementing stego algorithm which hides around 37.5% of the image size and also The average of hashed size is 8.51% of the size of the original file.	Implemented high rate and high secure data compression by the use of encryption method.

Then Table 2 compares the file size during transmission for these algorithms (Noroozi *et al*., 2012). The proposed algorithm had reduced significantly the sizes of the file to be only 8 bytes for various original file sizes compared to other algorithms. It illustrates the average of hashed size is 8.51% of the size of the original file (Noroozi *et al.*, 2012). One of the main advantage of the proposed algorithms is that, the algorithm generates a dynamic hashed file. It means that the size of hashed file depends directly on size of the original file with an average reduction of hashed file size is 8.51% compared to the original file (Table 1).

Table 2 Comparison of research works of size of hashed file

Size of original files (Byte)	Md5 Algorithm (Byte)	SHA-1 Algorithm (Byte)	SHA-2 Algorithm (Byte)	Proposed Algorithm (Byte)
14	32	40	64	8
18	32	40	64	8
72	32	40	64	8
1	32	40	64	8

4.2 Experimental Results

Experimental results that occured in the stego images due to embedding a large amount of secret message using our proposed method are undisclosed to human eye. The majority widely utilized measurements are: signal to noise ratio (SNR), peak signal to noise ratio (PSNR), mean squared error (MSE) and root mean square error (RMSE). The SNR represented by Equation (1):

$$SNR = 10 * \log_{10} \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(\mathrm{Aij})^2}{\sum_{i=1}^{n}\sum_{j=1}^{m}(\mathrm{Aij}-\mathrm{Bij})^2} \quad (1)$$

Aij represents one pixel of original image and Bij represents one pixel in the stego-image. The measuring unit is decibel (db). The MSE measurement for the steganographic objects is represented by Equation (2):

$$MSE = \frac{1}{m*n}\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathrm{Aij}-\mathrm{Bij})^2(\mathrm{Aij})^2 \quad (2)$$

M and N represents the height and width of the image. The peak signal to noise ratio (PSNR) is the metric that resulted from the computed the peak signal to noise ratio, between two images (Jarvinen, *et al*., 2005); (Jansirani, 2011). The PSNR is represented in Eq. 3:

$$PSNR = 10 * log_{10} \frac{(Max)^2}{\frac{1}{m*n}\sum_{i=1}^{n}\sum_{j=1}^{m}(Aij-Bij)^2} \quad (3)$$

MSE illustrates square error and Max shows the high cost of calories that is 255. Denote squared error (MSE) and peak signal to noise ratio (PSNR) has been utilized in the preceeding for video and image doling out society since reliability metrics (PSNR) is immediately a logarithmic demonstration of MSE while the RMSE demonstrating the square root of (MSE).

The implementing of these metrics is fast and easy and the compute for these metrics are simple to comprehend because these metrics are gaining popularity illustrate in Equation (4):

$$PSNR = \frac{1}{m*n}\sum_{i=1}^{n}\sum_{j=1}^{m}(Aij-Bij)^2\ 10 * log_{10}\frac{(Max)^2}{} \quad (4)$$

The researchers used 51,219 bytes from the image to hide the data and the PSNR value for their approach is 4l.ldb (Wang *et al*., 2007). On the other hand, (Chang and Chung, 2002) used 53,248bits from the image to hide the data. The PSNR value of their method is 34.84db, Moreover (Chang and Tseng, 2004) increases capacity to 389,004 bits while the PSNR value is only 4l.22db. The RMSE has been used previously as a metric to measure the quantity of deformation being additional to the image overall following embedding the protected of text. In

(Wu and Tsai, 2003) the RMSE has been used to measure the efficiency of their proposed method and found that by hiding 50,960 bits, the RMSE value is 2.07 db. Satish *et al.*, (2004) have used the size of image 256*256 (64 KB) to hide data. The authors used the SNR to evaluate the presented come near in their paper where the SNR to calculate the proposed advance in their document where the SNR value of 18.1476 db has been obtained for the steganographic image after embedding the data. Additionally, (Chang and Tseng, 2004) used the image with a size of 512*512 and 256 Gray levels to hide data by modifying the normal LSB insertion method with an enhanced one and compare the measurements of the distortion level using the MSE metric with the distortion level by using the normal LSB insertion method for the same image. The authors are claiming that the enhanced LSB has the MSE value of 2168.6 db while the normal LSB insertion method has the MSE value of 52l9.4 db.

For the presented metrics, the distortion level considered as acceptable for the image overall but with no accurate measure for the image pixels that have been affected more than the other pixels. Table 3 shows the comparison of SNR, PSNR and RMSE by using different methods from the researchers works.

Table 3 Comparison of different image steganography methods

Researchers	Size of Image	PSNR (db)	SNR (db)	RMSE (db)
(Wang *et a*l., 2007)	51,219 bytes	4l.l	…….	…….
(Chang and Chung, 2002)	53,248 bits	34.84	…….	…….
(Chang and Tseng, 2004)	389,04 bits	4l.22	…….	…….
(Wu and Tsai, 2003)	50,960 bits	…….	…….	2.07
(Satish *et al.*, 2004)	64 Kb	…….	18.1476	…….
(Hossain *et al*., 2009)	51,219 Bytes	41.1468	…….	…….
Proposed Algorithm	51,219 Bytes	53. 7 61 8	…….	…….

Using Equation (2) and Equation (3), we calculated PSNR value of stego image, these PSNR values are displayed in Table 3. The experimental results by applying our proposed method on different standard images are compared with other methods employed by few researchers.

Another point to highlight is that, without secret key no one able to know the exact position where the hidden information is placed. Then to extract the hidden information, the secret key is needed. So the proposed method provides an effective way to implant hidden information into the cover image without producing clear distortion.

5.0 CONCLUSION

The purpose of this paper was to present a new digital signature algorithm which produces the hashed file with dynamic size that means the result of hash function depends on size of the message. The proposed method is an effective way to integrate hidden information reporting without significant distortion. And the use of the secret key gives a way to secure the information from illegal user. It works with all kinds of files and can be used for sending low size messages, like multi agent system with agents that send few byte messages. It can be applied for transferring small size file messages (like multi agent systems) that makes it fast, simple and secure. Our proposed method provides better PSNR value where larger PSNR indicates better quality of the image or in other terms lower distortion.

Acknowledgement

In preparing this paper, I wish to express my sincere appreciation to my main supervisor for encouragement and guidance. I am also thankful to Advanced Informatics School (AIS), Universiti Teknologi Malaysia (UTM) for funding my paper.

References

[1] Atkin, S. and Yan, S. 2005. Constructing Dynamic Multilingual Pages in a Web Portal. *Proceeding of U.S. Patent Application.* 11/089: 393–399.

[2] Bellare, M., Micciancio, D. and Warinschi, B. 2003. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. Proceedings *of Advances in Cryptology—Eurocrypt Springer Berlin Heidelberg*. 614–629.

[3] Coleman, S., de la Iglesia, E., Khasgiwala, A., King, S., and Lowe, R. 2010. Verifying Captured Objects Before Presentation Proceedings of U.S. Patent. Washington, DC. No. 7,774,604.

[4] Fiat, A. and Shamir A. 1986. How To Prove Yourself: Practical Solutions to Identification and Signature Problem. *Crypto'86, LNCS* 0263. Springer-Verlag. 186–194,

[5] Hess, F. 2002. Efficient Identity Based Signature Schemes Based on Pairings. *Selected Areas in Cryptography (SAC'02), LNCS* 2595. Springer-Verlag. 310–324,

[6] Hmood, A., Zaidan, K., Zaidan B. B., Zaidan A. A. and Jalab, H. A. 2010. An Overview on Hiding Information Technique in Images. *Journal of Applied Sciences*. 10: 2094–2100

[7] Jarvinen, K., Tommiska, M., and Skytta, J. 2005. Hardware Implementation Analysis of The MD5 Hash Algorithm. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 05)*. 298a.

[8] Jansirani, A. 2011. Hi-Tech Authentication for Palette Images using Digital Signature and Data Hiding. *The International Arab Journal of Information Technology*. 8(2): 117–123.

[9] Kae-por, F. C. 2008. U.S. Patent. No. 7,328,344. Washington, DC: U.S. Patent and Trademark Office.

[10] Kang, J. 2003. Mobile Communication System Having Multi-Band Antenna. U.S. Patent. 6,662(.21).

[11] Keromytis, A. D. 2010. Voice-Over-IP Security: Research and Practice. *IEEE Security & Privacy*. 8(2): 76–78.

[12] Merkle, R. 1979. Secrecy, Authentication and Public Key Systems/ A Certified Digital Signature. Ph.D. dissertation, Dept. of Electrical Engineering, Stanford University.

[13] Naji, A. W., Hameed, S. A., Zaidan, B. B., Al-Khateeb, W. F., Khalifa, O. O., Zaidan, A. A., & Gunawan, T. S. 2009. Novel Framework for Hidden Data in the Image Page within Executable File Using Computation between Advanced Encryption Standard and Distortion Techniques. *(IJCSIS) International Journal of Computer Science and Information Security*. 3(1).

[14] Naji, A. W., Gunawan, T. S., Zaidan, A. A., Zaidan, B. B., Al-Khateeb, W. F., & Hameed, S. A. (2009). New Approach of Hidden Data In The Portable Executable File Without Change the Size of Carrier File Using Statistical Technique. *International Journal of Computer Science and Network Security (IJCSNS)*. 9(7): 218–224.

[15] Noroozi, E., Salwani, M., Sabouhi, A., Hafiza, A. 2012. A New Dynamic Hash Algorithm in Digital Signature. *Procs. of Advanced Machine Learning Technologies and Applications*, Springer-Verlag Berlin Heidelberg. 583–589.

[16] Noroozi, E., Salwani, M., Sabouhi, A and M., SalehNamadi. 2012. New Implementation of Hashing and Encoding in Digital Signature, *International Conference on Security Science and Technology–ICSST*. Hong Kong. 29: 50–54.

[17] Sakai R., Ohgishi K., and Kasahara M. 2000. Cryptosystems Based On Pairing. *2000 Symposium on Cryptography and Information Security* (SCIS'00). Okinawa, Japan. 26–28.

[18] Serret A., Gilles B. 2012. Methods And Systems For Encoding And Protecting Data Using Digital Signature And Watermarking Techniques. U.S. Patent. No. 8,099,601. 17 Jan. 2012.

[19] Shamir A. 1984. Identity-Based Cryptosystems And Signature Scheme. *CRYPTO'84, LNCS* 196. Springer-Verlag. 47–53,

[20] Zaidan, A. A., Zaidan, B. B., Abdulrazzaq, M. M., Raji, R. Z., & Mohammed, S. M. 2009. Implementation Stage For High Securing Cover-File Of Hidden Data Using Computation Between Cryptography And Steganography. *International Association of Computer Science and Information Technology (IACSIT)*. 19: 482–489.

[21] Wang, C. Ming., Wu, N. Tsai, Hwang, C.S. 2008. A high quality steganographic method with pixel-value differencing and modulus function. *The Journal of Systems and Software*. 81(1): 150–158.

[22] Chang, C.C., Chen, T.S., Chung, L.Z. 2002. A Steganographic Method Based Upon JPEG And Quantization Table Modification, *Information Sciences.* 141: 123–138.

[23] Chang, C.C., Tseng, H.W. 2004. A Steganographic Method for Digital Images Using Side Match. *Pattern Recognition Letters*, 25(2004): 1431–1437.

[24] Wu , D.C., Tsai, W.H. 2003. A Steganographic Method for Images by Pixel-Value Differencing. *Pattern Recognition Letters*, 24(2003): 1613–1626.

[25] Satish, K. T. Jayakar, Madhavi, Charles, Tobin K. and Murali, K. 2004. Chaos Based Spread Spectrum Image Steganography. *IEEE Transactions on Consumer 588 Electronics*. 50(2): 587–590.

[26] Hossain, M., Haque, S.A. and Sharmin, F. 2009. Variable Rate Steganography in Gray Scale Digital Images Using Neighborhood Pixel Information, *Proceedings of 2009, 12th International Conference on Computer and Information Technology (ICCIT* 2009), Dhaka, Bangladesh. 21–23.