جامعة نايف العربية للعلوم الأمنية
Naif Arab University for Security Sciences

Naif Arab University for Security Sciences
Journal of Information Security & Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR

JISCR

Secure Mobile Computing Authentication Utilizing Hash, Cryptography and Steganography Combination

Muneera Alotaibi [1], Daniah Al-hendi [1], Budoor Alroithy [1], Manal AlGhamdi [2], Adnan Gutub [1*]

[1] *Computer Engineering Department, College of Computers & Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia.*

[2] *Computer Sciences Department, College of Computers & Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia.*

Received 10 Nov. 2018; Accepted 07 Feb. 2019; Available Online 01 Apr. 2019

Abstract

In this paper, we propose a modification for security authentication systems in mobile devices. Our enhancement is designed to secure information transformation over the internet by combining hash, cryptography and steganography mechanisms. We used the combination for authentication, to secure mobile computing to transfer data in a trusted manner. The proposed work will use hash function to store the secret password to provide increased security. The hashed password is encrypted using AES encryption then hidden inside an image to be called back for authentication whenever needed. The security services provided by this combination mechanisms can assure authenticity, confidentiality, and integrity. Results and comparisons to different options of available mobile computing methods proved that our proposed technique is a promising research direction for real mobile security.

I. Introduction

Mobile computing security is one of the most important issues in the research field of information and communication technology. As has been recently noted of higher and higher number of users of the internet mobile computing, system security is required more and more to protect users and their data [1]. Nowadays, people are using the internet mobile computing most of the time; they need their connection to the internet to be secure against any malicious attacks by strangers and hackers. There are requirements that have to be met in order to have a secure internet connection. These requirements protect data against unauthorized access, data loss, unauthorized modifications, etc; needing authentication of the mobile computing systems passwords to guard the mobile usages and their data [2].

The famous methods to protect personal user data and their passwords within systems are steganography and cryptography [3]. Steganography achieves this motivation by hiding passwords inside an image. It means no one can pay attention to the hidden data within it. Cryptography (to encrypt the data) is used when data is transferred over unsafe media on the internet and the data needs to be protected from intruders [4]. It saves data by converting texts from readable text to unreadable text. The attacker does not know what the message is because it is hidden in the image, and it is encrypted to another, so that the intruder cannot read it. The security services provided by combining the two mechanisms can assure authenticity, confidentiality, and integrity as detailed in [5].

Keywords: *Mobile security; LSB image steganography; Hash function; AES(Advanced Encryption Standard).*

Production and hosting by NAUSS

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

doi: 10.26735/16587790.2019.001

The aim of this paper is to present an authentication scheme of enhancing the vulnerability in registration passwords to be used in mobile computing. The work focused on the Android platform to improve the research of T. Mantoro et al. [6] whom proposing an authentication technique for Android mobile devices using AES with steganography. The work is improved via two implementation assumptions. First, we redesign T. Mantoroidea combining Steganography with Hash, i.e. Least Significant Bit (LSB) Steganography and hash function, together as authentication method for Android systems. Then, the second assumption tested authentication implementation combining the first assumption of Hash Steganography with Cryptography making the proposed security system, i.e. the objective of the work as digital signature authentication for secure mobile computing utilizing LSB Steganography, hash function, and AES (Advanced Encryption Standard) in order to be compared to previous options showing interesting results. This work can have different important real applications as found similar in principle to the research of integrity verification using cryptographic hash function and compression as detailed in [7].

The paper is organized as follows. Section 2 will discuss a brief theoretical background of the security techniques utilized, i.e. steganography, cryptography and hash functions. Section 3 will include a description of related work for mobile computing security. We will cover a literature survey of similar attempts to use as references to compare our testing with. Section 4 will present our proposed security enhancement. Section 5 will emphasis on the testing results and detailed comparisons followed by Section 6 to conclude the work.

II. Background

In this paper, the combination of cryptography and image steganography insertion technique are used to effectively ensure authenticity, confidentiality, and integrity. This section will briefly describe the needed LSB image steganography, AES (Advanced Encryption Standard), and used Hash function.

A. LSB Image Steganography

Steganography comes from the Greek word “stegos” meaning cover and “grafia” meaning writing. Scientifically, it is used to hide or cover messages in a communication between sender and receiver so that no-one can find them. In other words, it’s the transmission of information that is masked inside another media information. Steganography comprised three types of concealing messages inside other information: Text Steganography, Image Steganography and Audio/Video Steganography [8]. All these types ensure the person who views the object that contains the hidden information will not recognize any hidden information as it will be difficult to be detect. The purpose is to provide confidentiality of communication when data is stored and protected from modification or manipulation; this means securing the integrity of the data. Steganography can be formed via many algorithms such as Least Significant Bit (LSB), JStegand F5 algorithm [6]. Our system adopts the Least Significant Bit (LSB) technique as justified in [9].

The Least Significant Bit (LSB) stego technique is used to embed information inside the image file. Both information or data is converted from ASCII value to binary value and the image to be used as cover object are converted from pixel format to binary value. The Least Significant Bit of the image is substituted with the bit of the data to be transferred so this demonstrates that the message is hidden [9]. This means each bit of the data is replaced in the cover image of the least significant bit. For example, if the secret data is ‘a’, its ASCII code is (01100001) and there are three pixels’ bits of an image as follows:

(10110101 01101100 10101101)
(10110110 11001101 00111110)
(10110101 01100011 10001110)

Then the resulting One-LSB algorithm will be like this:

(10110100 01101101 10101101)
(10110110 11001100 00111110)
(10110100 01100011 10001110)

B. AES (Advanced Encryption Standard)

Cryptography has been used for some decades now [10]. It is a technique used for hiding data by encryption and decryption. Cryptography is a Greek word, the first part of the word is “Krypto”, which means hidden, and the second part of the word, “graphene”, means writing. It works by converting plaintext to cipher-text by using certain algorithms.

Many forms of cryptography are famous and popular to use [11]. The wide range of cryptographic algorithms started to publicize for decades by DES (Data Encryption Standard) then standardized, after 2000, as AES (Advanced Encryption Standard) [12]. AES (Advanced Encryption Standard) is a symmetric-key algorithm [12].

Initially, when the algorithms first started to appear, it was known as the Rijndael algorithm. It was introduced by the National Institute of Standards (NIST). It is one of the most famous and popular algorithms used for encryption and decryption data [13]. AES uses the same key during the encryption and decryption processes that can vary between three key length sizes: 128-bits, 192-bits, and 256-bits. The number of rounds depends on the length of the key. For example, 128 bits key needs 10 rounds, 192 bits key requests 12 rounds and 256 bits key uses 14 rounds. Each round consists of four specific layers: ByteSub, ShiftRow, MixCol, and Key addition. The reader is referred to literature [14] for details of AES as well as further related cryptography elaboration.

C. Hash Function

The hash function provides a message authentication service to verify authentication and integrity, as found in many Hash algorithms such as the common MD5 (Message Digest 5), SHA1(Secure Hash Algorithm 1), and the version SHA-2 (Secure Hash Algorithm 2) family consists of SHA256, SHA384, and SHA521 [15]. The hash function accepts any length of message input and computes it to a fixed-length size output known as a message digest or hash value. The idea of this mechanism is to generate a fingerprint of the message to be utilized effectively and efficiently so that it prevents duplication of data. This means each message has a unique message digest. The hash function has many properties including a one-way function where the hash function makes the output impossible to reverse to the input, acceptance of any input data size to the applied hash function, production of a fixed data size output, and efficiency, making it easy to compute for any input. Any modification in the input means new hash value and multi-input to hash function means multi-hash value output [7]. There are many purposes of hashing such as comparing a large amount of data or producing a fixed length size output so that it is easier to compare the hash value than the entire data itself. It also makes it easy to avoid duplication of data stored in databases and makes it easy to find the record. This means the hash function is an efficient way to store passwords. A real interesting example of using hash for authentication is found in [7].

III. Related Works

Most mobile computing authentication attempts adopted Advanced Encryption Standard (AES) alone as is or combined to other security techniques such as hash or steganography. To elaborate more, different attempts have been reviewed. For example, S. Ladgham et al. [9] proposed an improved approach for the LSB technique. They worked to reduce the length of the hidden secret message by using the Deflate algorithm. This algorithm is used to make a lossless data compression algorithm. They then protected data by using AES. The message passes through three main phases to be injected into the cover image. The first phase encrypts data by AES. Then, the data is compressed with the Deflate algorithm. Finally, the data is converted to binary mode.

S. Bukhari et al. [14] proposed an approach that depended on image steganography and cryptography by using double random phase encoding. To make a stego image they proposed to hide the message image inside another image. Then, the stego image was divided in two part 8 * 8 blocks. The next step was to perform cryptography double random phase encoding (DRPE) on the divided stego image. The result of this technique enhances the security of the message when transmitted wirelessly.

R. Tresa et al. [15] proposed a steganography scheme based on a hash function coupled with AES encryption. They presented a combination of two techniques, steganography and cryptography. The textual data of userswas encrypted by AES. After that, the hash function was applied to encrypted textual data to be stored in the cover image.

S. M. Moe [16] proposed a message authentication system using image steganography. To ensure integrity, they used hashing cryptography. The sender applied SHA-512 to the plaintext to get digested. Then, the plaintext and digest were embedded into a cover image to get a stego image. At the other end, the receiver extracted the plaintext and digest from a stego image. After that, the receiver computed SHA-512 on the plaintext to get a digest to verify and compare with the digest from the sender.

T. Mantoro et al. [6] presented an authentication technique for Android mobile devices using AES with steganography. They consider users attempt to enter username, password, email and cover images that are used to hide the user specific password inside it. Then, the application encrypts the user password by using AES. The encrypted password is further hidden in the chosen cover image by Least Significant Bit (LSB) steganography technique. This LSB stego technique produces a stegano-image that is used as a digital signature to identify the user. To extract the hidden encrypted password from stegano-im-

age, the AES algorithm decodes the encrypted password to extract the original password. The extracted password is then compared with the password that is saved in the database. If they match, user authentication is successfully completed, otherwise it is rejected [6]. T. Mantoro research is studied more in this work to be improved as detailed next.

A. Stegano-Image for Digital Signature

This research is benefitting from different ideas to improve the work presented by T. Mantoro et al.[6], with its authentication system based on two main operations: Signup and Login operations. The Signup operation is the setup process used to authenticate a user to the system. It performs the tasks of encrypting and hiding the data inside an image. First, the user sets his/her username, password, and email. Also, he chooses the cover image as a digital signature. The system performs three steps. In the first step, the system saves the entered user's data to the database. The second step is encrypting the password and the username used as a key in AES encryption. Then the encrypted password is hidden in the image by the LSB technique [6]. The general idea of LSB is to hide the bits of data in the least significant bit in each pixel of the image (i.e. changing the least significant bit of each pixel according to the bits of secret data). Each pixel can be used to hide three bits of data because each pixel consists of three bytes. Suppose that the size of data to be hidden is one byte, then it needs three pixels to hide the bits. The Login operation is the inverse process of the Signup operation, whereas the system verifies the identity of the user. It extracts the hidden data from the image using the LSB algorithm then decrypts the extracted result by AES decryption with the username as a key to get the data of user identity as illustrated in the pseudo code below.

The pseudo code of the authentication system studied [6] is presented below:

The authentication algorithm

Choose one operation from the list in the main interface.

a- If the user chooses the Signup operation:

1. Go to the Signup interface.
2. Enter your username, password and email.
3. Choose the cover image as a digital signature.
4. Press the Signup button.
5. Store data to the database.
6. Encrypt the password with AES algorithm using username as a key.
7. Hide the result of AES algorithm in cover image using LSB algorithm.
8. Save the result image as a stego-image.
9. Print "Registration Success…".
10. Go to the main interface.

b- If the user chooses the Login operation:

1. Go to the Login interface.
2. Enter your username.
3. Choose the saved stego-image.
4. Click the Sign in button.
5. Pick the hidden data from the stego-image using the LSB algorithm.
6. Decrypt result from LSB algorithm by using AES algorithm and using the username as a key.
7. Set the password to the result from the AES algorithm.
8. Find a row in database such that username column is equal to entered username and password column is equal to the password.

If the server finds the row in database:

1. Print "CONGRATULATIONS Login Successful...".
2. Go to the access granted interface.
3. Show the username.

Otherwise:

Print "Login Failed...Try Again".

B. Graphical User Interface (GUI) Implemented

The authentication system studied to be improved consists of four major interfaces: the main interface, the signup interface, the login interface and the access granted interface, as shown in [6, Fig. 1] . The main interface is the first interface that appears to users and displays a list of options such as signup and login. When a user clicks on the signup option, it transfers to the signup interface. The signup interface allows new user to register within the system. Also, it creates digital signature of the new user. The login interface allows the user to be authenticated on the system. If the authentication process completes successfully, the access granted interface appears, otherwise the system stays in the login interface [6]. Screenshots of the system's interface are shown in Fig. 1.

Fig. 1. Screenshots of the system's interface [6].

IV. Proposed Security Enhancement

This work enhances the security of the digital signature technique proposed by T. Mantoroet al. [6]. The aim of this work is to encrypt the passwords and hide them in a cover image using LSB steganography. We implemented this work using the Android platform and using a MySQL database. T.Mantoro [6] technique has an important weak point which is storing the passwords as clear text in the database thus leaving them vulnerable to exposure to database attacks such as SQL injection attack. Therefore, we in this work avoid the previously mentioned weak point by saving the hash digest of the password in the database instead of saving the password itself, i.e. using the hash function. Another reason for choosing the hash function to improve the weak point is its time efficiency, i.e. hash digest consumes a little bit less time in its AES computations. Also, the hash function always produces the same output length independent to the length of its input (e.g. passwords) which is a data storage standardization benefit. The improvement is implemented based on two main assumptions as follows:

1. Implement the digital signature security method based on hash function and LSB stego technique.
2. Implement the digital signature security scheme based on hash function combined with AES as well as LSB stego technique.

For the first, hash stego implementation, assumption, we tested five different kinds of hash function which are Message Digest 5 (MD5), Secure Hash Algorithm 1 (SHA-1), SHA-256, SHA-384, and SHA-512. After extracting the hash digest of the password, the digest is embedded into a cover-image using LSB technique. The processes of the first assumption are illustrated in Fig. 2. Moreover, this assumption guarantees the confidentiality of the password by LSB technique and the integrity of the password by hash function as overcoming the drawback of research in [6]. Also, the merging between LSB technique and hash function emphasizes the authenticity of the user.

The sequence of operations in the system is based on the first assumption as in the following steps:

Choose one operation from the list in the main interface.

a. If the user chooses the Signup operation:

1. Go to the Signup interface.
2. Enter your username, password, and email.
3. Choose the cover image as a digital signature.
4. Press the Signup button.
5. Store data in the database.
6. Compute the hash value of the password.
7. Hide the result of the hashing algorithm in the cover image using the Two-LSB algorithm.
8. Save the resulting image as a stego-image.
9. Print "Registration Success…".
10. Go to the main interface.

b. If the user chooses the Login operation:

1. Go to the Login interface.
2. Enter your username.
3. Choose the saved stego-image.
4. Click the sign in button.
5. Pick the hidden data from the stego-image using the LSB algorithm.
6. Find a row in the database such that the username column is equal to the entered username and the hashing password column is equal to the hashing password.

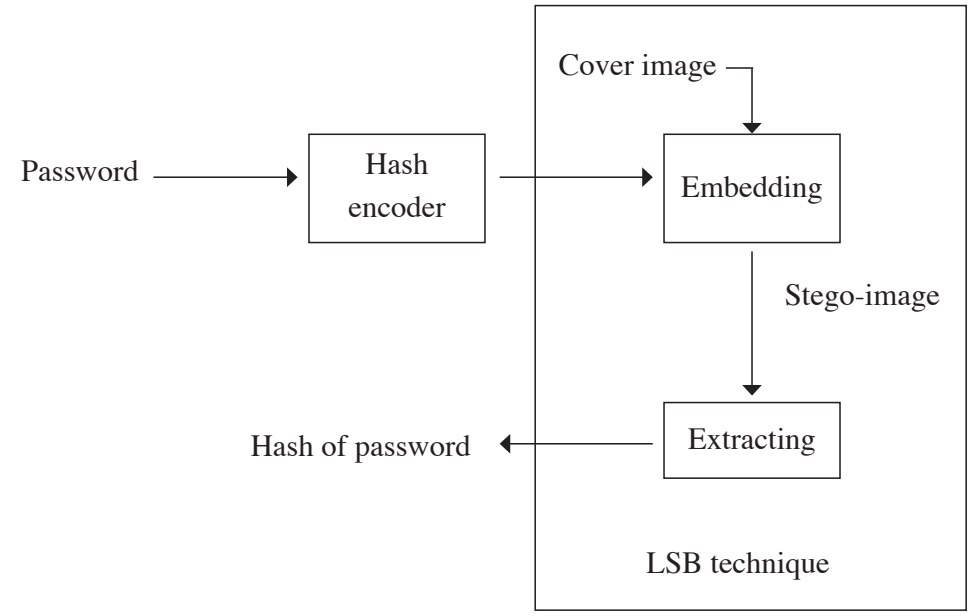

Fig. 2. First assumption (Hash + LSB Stego technique).

If the server finds the row in the database:

1. Print "CONGRATULATIONS Login Successful...".
2. Go to the access granted interface.
3. Show the username.

Otherwise:

Print "Login Failed...Try Again".

In the second assumption, the security of the digital signature technique is upgraded by adding another level of security. Therefore, it used three algorithms to improve the security which are the hash function, AES crypto algorithm, and LSB stego technique. The type of hash function that is used in the second assumption is SHA-256, as considered a fair secure algorithm to be used for research bases [15]. The sequence of the processes in the second three level assumption is shown in Fig. 3. In addition, this assumption is considered to provide a higher level of password confidentiality by combining the AES algorithm and the LSB technique as proven in [17]. Also, the use of the hash function guarantees the integrity of the password which is an integrity verification essential requirement as justified in [7], i.e. this assumption checks the security and authenticity of the user by using the three algorithms.

The sequence of the operations in the system is based on the second assumption as in the following steps:

Choose one operation from the list in the main interface.

a. If the user chooses the Sign up operation.

1. Go to the Signup interface.
2. Enter your username, password, and email.
3. Choose the cover image as a digital signature.
4. Press the signup button.
5. Store data in the database.
6. Compute the hash value of the password.
7. Encrypt the hashing password with the AES algorithm using the username as a key for the AES algorithm.
8. Hide the result of the AES algorithm in the cover image using the Two-LSB algorithm.
9. Save the resulting image as a stego-image.
10. Print "Registration Success…".
11. Go to the main interface.

b. On the other hand, if the user chooses the Login operation

1. Go to the Login interface.
2. Enter your username. Choose the saved stego-image.
3. Click the sign in button.
4. Pick the hidden data from the stego-image using the LSB algorithm.

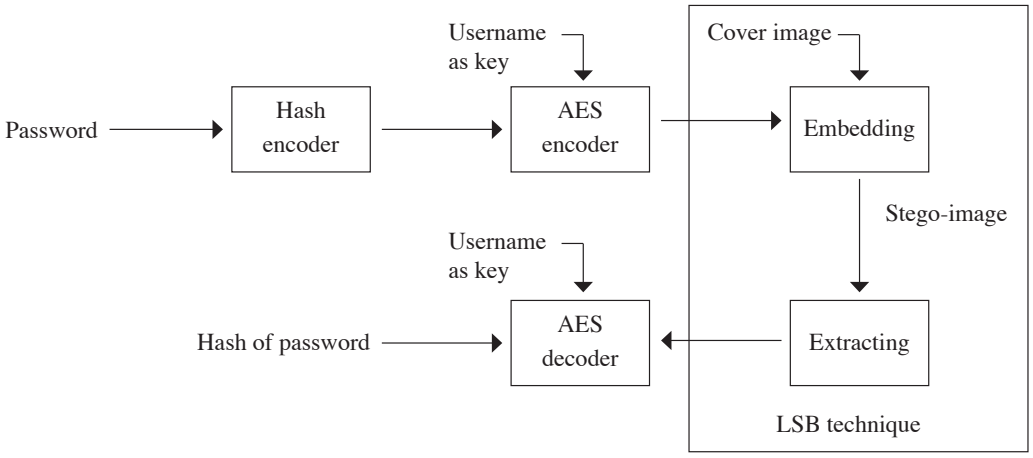

Fig. 3. Second assumption (Hash + AES + LSB technique)

5. Decrypt the result from the Two-LSB algorithm by the AES algorithm using the username as a key.
6. Set the hashing password to the result of the AES algorithm.
7. Find a row in the database such that the username column is equal to the entered username and the hashing password column is equal to the hashing password.

If the server finds the row in the database:

1. Print "CONGRATULATIONS Login Successful...".
2. Go to the access granted interface.
3. Show the username.

Otherwise:

Print "Login Failed...Try Again".

V. Results and Comparisons

The proposed digital signature system was tested and compared with the previous digital signature technique [6] under three testing factors which are the elapsed time that occurred while creating the digital signature, elapsed time spent verifying the digital signature, and Peak Signal to Noise Ratio (PSNR). These timing factors and the PSNR testing are found appropriate for this proposed digital signature authentication differently than other experimentation factors used within the elliptic polynomial cryptography study [18] as well as the performance evaluation of hardware designs found in the expandable multipliers [19] and security arithmetic inverters [20]. In fact, the timing performance is found common within this authentication study similar in principle to the cryptography security analysis in previous studies of GF(p) hardware modeling [21] as well as elliptic curve GF(2) crypto processors simulations [22]. The security analysis can have different factors to build its test on such as bit-ratio differences found within secret sharing studies [23]. This work testing scope is based on the timing and PSNR assumed to provide fair comparison showing attractive open research improvement over [6].

The elapsed time spent creating the digital signature is the encryption time plus the time for hiding data in the cover image, whereas the elapsed time spent verifying the digital signature is the time taken to extract the hidden data from the stego-image plus the decryption time of the extracted data. The elapsed time spent creating and verifying the digital signature are tested in MATLAB using an image of size 320 * 240 pixels. This image was tested five times according to the number of characters in the secret data that ranged between 20 to 60 characters.

The PSNR is "The ratio of the square of the peak value the signal could have to the noise variance" [6]. It is computed by the following equation:

$$\text{PSNR}=10 \log \frac{255^2}{\text{MSE}} \text{ dB}$$

The calculated PSNR as well as the elapsed time for creating and verifying the digital signature for the T. Mantoro et al. [6] technique and for our proposed first and second assumptions are all computed within the execution time. The resulting times required to create and verify the digital signature are represented in Fig. 4 and 5, respectively. The result of testing the T. Mantoro et al. [6] technique and testing the first and second assumptions under PSNR factors are shown in Fig. 6.

According to the comparison figures, the results provide different variations expressing trade-off analysis. For example, the elapsed time for creating a digital signature considers the SHA-256 + LSB technique consuming the highest amount of time compared to the other approaches; while the SHA1 + LSB technique takes the least time related to the others. For elapsed time of verifying the digital signature, the AES + SHA-256 + LSB technique consumes the highest time compared to the other approaches, while the SHA-1 + LSB technique takes the least time related to the others. Moreover, theSHA-256 + LSB technique guarantees the highest level of security according to the PSNR values whereas the SHA-512 + LSB has the least value of PSNR thus it provides the lowest level of security.

Therefore, it is recommended to use the digital signature approach with three levels of security for highly critical applications such as e-banking applications, because it uses the highest practical level of security compared to the other approaches, as shown briefly in Table I. For applications that requires high-speed processing with a reasonable level of security, such as game applications, it is recommended to use the lighter digital signature technique based on the hash-stego function, i.e. especially based on SHA-1.

VI. Conclusion

This work enhances the security of the authentication system in mobile devices. The weak point studied in previous research is storing the passwords as clear text in the database. Thus, leaving passwords vulnerable to exposure to database attacks such as SQL injection attack. Therefore, this work avoids the previously mentioned weak point by saving the Hash digest of the password in the database instead of saving the plain password itself, i.e. adopting using the hash function. Our improvements are based on the two main assumptions: 1) Creating the digital signature based on two levels security which are the hash function combined to the LSB stego technique. 2) Creating the digital signature based on three levels security combining cryptography to the first option, i.e. combining the hash function, AES crypto algorithm, and the LSB stego technique for maximum security. The results showed that the combination of all security methods is giving the best attributes, i.e. confidentiality, integrity, authentication. While the previous work of crypto stego method is losing in integrity. A moderate security method can be of hash stego functions providing all three at-

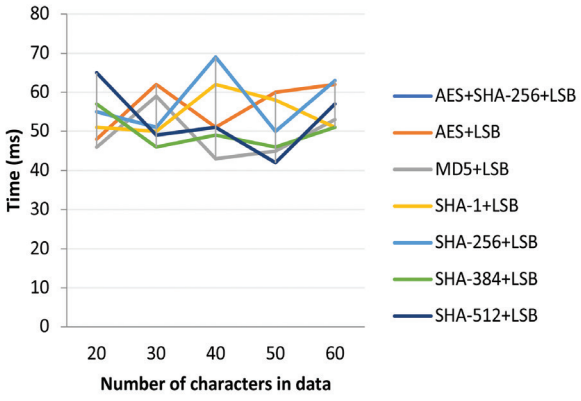

Fig. 4. Time required to create a digital signature for all techniques

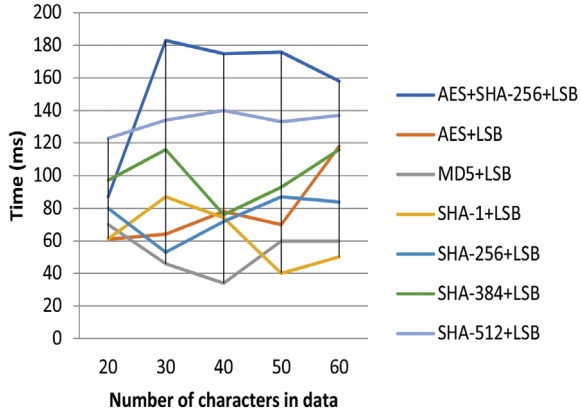

Fig. 5. Time required for verifying a digital signature for all techniques

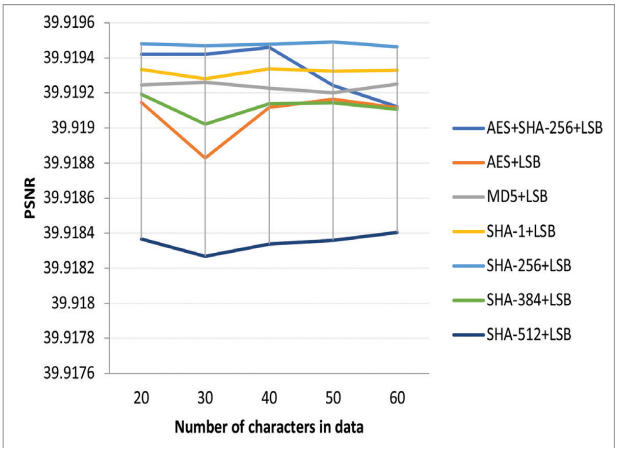

Fig. 6. PSNT testing for all techniques

TABLE I
THE SECURITY GOALS WITH THE DIFFERENT DIGITAL SIGNATURE APPROACHES

The digital signature technique	Differentiating Features		
	Confidentiality	Integrity	Authentication
AES + LSB	√		√
Hash + LSB	√	√	√
Hash + AES + LSB		√	√

tributes for speedy applications but considered slightly less secure when compared to the method involving AES cryptography.

ACKNOWLEDGMENT

We would like to thank Umm Al-Qura University (UQU) for supporting this research. We thank the College of Computers and Information Systems for helping us work this contribution through its master graduate program. Many thanks for the fruitful cooperation between Computer Sciences Department (via Dr. Manal AlGhamdi) and Computer Engineering Department (via Prof. Adnan Gutub) for supervising this work.

REFERENCES

[1] A. Gutub, "Exploratory Data Visualization for Smart Systems," presented at 3rd Annu. Digit. Grids Smart Cities Workshop, Riyadh, Saudi Arabia, May 2015.

[2] N. Alharthi and A. Gutub, "Data Visualization to Explore Improving Decision-Making within Hajj Services," in *Sci. Model. Res.,* vol. 2, no. 1, pp. 9-18, 2017, doi: 10.20448/808.2.1.9.18.

[3] N. A. Al-Juaid, A. A. Gutub and E. A. Khan, "Enhancing PC Data Security via Combining RSA Cryptography and Video Based Steganography," *J. Inf. Secur. Cybercrimes Res.*, vol. 1, no. 1, pp. 8-18, May 30, 2018, doi: 10.26735/16587790.2018.006.

[4] A. Gutub, "Subthreshold SRAM Designs for Cryptography Security Computations," in *Int. Conf. Softw. Eng. Comput. Syst.,* in Software Engineering and Computer Systems, in Communication in Computer and Information Science, vol. 179, pp 104-110, doi: 10.1007/978-3-642-22170-5_9.

[5] N. A. Al-Otaibi and A. Gutub, "2-Leyer Security System for Hiding Sensitive Text Data on Personal Computers," *Lect. Notes Inf. Theory*, vol. 2, no. 2, pp. 151-157, June 2014, doi: 10.12720/lnit.2.2.151-157.

[6] T. Mantoro, D. D. Permadi and A. Abubakar, "Stegano-image as a digital signature to improve security authentication system in mobile computing," in *2016 Int. Conf. Inf. Comput. (ICIC),* Mataram, 2016, pp. 158-163, doi: 10.1109/IAC.2016.7905708.

[7] M. Almazrooie, A. Samsudin, A. A. Gutub, M. S. Salleh, M. A. Omer and S. A. Hassan, "Integrity verification for digital Holy Quran verses using cryptographic hash function and compression," *J. king Saud Univ. Comput. Inf. Sci.,* vol. 32, no. 1, pp. 24-34, Jan. 2020, doi: 10.1016/j.jksuci.2018.02.006.

[8] A. Gutub and N. Al-Juaid, "Multi-bits stego-system for hiding text in multimedia images based on user security priority," *J. Comput. Hardw. Eng.,* vol. 1, Apr. 2018, doi: 10.63019/jche.v1i2.513.

[9] S. L. Chikouche and N. Chikouche, "An improved approach for lsb-based image steganography using AES algorithm," *2017 5th Int. Conf. Electr. Eng. - Boumerdes (ICEE-B)*, Boumerdes, 2017, pp. 1-6, doi: 10.1109/ICEE-B.2017.8192077.

[10] A. Gutub, "High Speed Low Power GF (2k) Elliptic Curve Cryptography Processor Architecture," *IEEE 10th Annu. Technical Exchange Meetin*g, KFUPM, Dhahran, Saudi Arabia, Mar. 23-24, 2003.

[11] A. A. Gutub and H. A. Tahhan, "Efficient Adders to Speedup Modular Multiplication for Cryptography," presented at 5th *IEEE Int. Workshop Signal Process. Appl.,* University of Sharjah, UAE, Mar. 18-20, 2018.

[12] A. A. Gutub and F. A. Khan, "Hybrid Crypto Hardware Utilizing Symmetric-Key and Public-Key Cryptosystems," *2012 Int. Conf. Adv. Comput. Sci. Appl. Technol. (ACSAT)*, Kuala Lumpur, 2012, pp. 116-121, doi: 10.1109/ACSAT.2012.44.

[13] N. Alassaf, B. Alkazemi and A. Gutub, "Application Light-Weight Cryptography to Secure Medical Data in IOT Systems," *J. Res. Eng. Appl. Sci.,* vol. 2, no. 2, pp. 50-58, Apr. 2017, doi: 10.46565/jreas.2017.v02i02.002.

[14] S. Bukhari, M. S. Arif, M. R. Anjum and S. Dilbar, "Enhancing security of images by Steganography and Cryptography techniques," *2016 Sixth Int. Conf. Innov. Comput. Technol. (INTECH),* Dublin, 2016, pp. 531-534, doi: 10.1109/INTECH.2016.7845050.

[15] R. Tresa, A. M. Babu and Sobha T, "A Novel Steganographic Scheme Based On Hash Function Coupled With AES Encryption," *Adv. Comput.: Int. J. (ACIJ),* vol. 5, no. 1, pp. 25-34, Jan. 2014, doi: 10.5121/acij.2014.5103.

[16] M. S. S. Moe and S. W. Phyo, "Massage Authentication System using Image Steganography," *Int. J. Sci. Eng. Technol. Res.,* vol. 3, no. 11, pp. 2453-2457, June 2014.

[17] N. A. Al-Otaibi and A. A. Gutub, "Flexible Stego-System for Hiding Text in Image of Personal Computers Based on User Security Priority," in *Proc.2014 Int. Conf. Adv. Eng. Technol. (AET-2014),* Dubai, UAE, Dec. 25-26, 2014, pp. 250-256.

[18] L. Ghouti, M. K. Ibrahim and A. A. Gutub, " Elliptic Polynomial Cryptography with Secret Key Embedding," U.S. Patent 8 351 601 B2, Jan. 8, 2013.

[19] A. A. A. Gutub and A. A. M. Amin, "An expandable Montgomery modular multiplication processor," *ICM'99. Proc. Eleventh Int. Conf. Microelectron. (IEEE Cat. No.99EX388),* Kuwait, 1999, pp. 173-176, doi: 10.1109/ICM.2000.884833.

[20] A. A. Gutub and A. F. Tenca, "Efficient scalable VLSI architecture for Montgomery inversion in GF (p)," *Integr.,* vol. 37, no. 2, pp. 103-120, May 2004, doi: 10.1016/j.vlsi.2003.12.001.

[21] A. A. Gutub, "Merging GF(p) Elliptic Curve Point Adding and Doubling on Pipelined VLSI Cryptographic ASIC Architecture," *Int. J. Comput. Sci. Netw. Secur. (IJCSNS),* vol. 6, no. 3A, pp. 44-52, Mar. 2006.

[22] A. A. Gutub, "Area Flexible GF(2k) Elliptic Curve Cryptography Coprocessor," *Int. Arab. J. Inf. Technol.,* vol. 4, no. 1, Jan. 2007.

[23] A. A. Gutub, N. Al-Juaid and E. Khan, "Counting-based secret sharing technique for multimedia application," *Multimed. Tools Appl.,* vol. 78, pp. 5591-5619, Mar. 2019, doi: 10.1007/s11042-017-5293-6.