Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 3973-3983
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6211

Using steganography techniques for implicit authentication to enhance sensitive data hiding

Rafal Najeh Kadhum[a,*], Nada Hussein M. Ali[a]

[a]Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

Smartphones recently become indispensable to users due to the services provided, like Internet access and important applications (e.g., financial and health applications). which contain personal and confidential information about the user. Consequently, this information is vulnerable to hackers and data privacy violations. This article describes the effective safeguard of sensitive data from different attacks. It's the second level of protection after using implicit authentication. If implicit authentication allows an unauthorized user to access sensitive data, we encrypt it using AES-GCM and hide it under a cover image with LSB Steganography. We noticed that recommended solution is highly effective at data protection while consuming less Memory and being relatively fast. To evaluate the system's efficacy, Peak Signal-to-Noise Ratio(PSNR), Means Square Error(MSE), and Pearson Coefficient Correlation (PCC)are employed as evaluation metrics. Best results are found in sunset image 27KB, with values of MSE (0.005245) and PSNR (70.9547). The (PCC) values are always zero, indicating that there is no relationship between the original text and the encrypted text, implying that the AES-GCM encryption is effective.

Keywords: Encryption, Steganography, information security, data protection, AES, GCM, LSB, sensitive data, features, Android Studio, AES-GCM, Mobile Computing, hiding, Implicit Authentication
2010 MSC: 68P25, 68M25, 94A62

1. Introduction

Smartphones now have capabilities comparable to desktop computers or laptops, thanks to fast growth. As a result, cell phones are the preferred device for entertainment, internet browsing, and

*Corresponding author
Email address: rafal.najeh@gmail.com (Rafal Najeh Kadhum)

Received: November 2021 *Accepted:* February 2022

storing sensitive data [2]. Therefore, Unauthorized access to this sensitive data must be prevented, there are various security techniques such as cryptography and steganography that are used for security purposes.

Cryptography converts the secret message into some other forms, such that it is not understandable to anyone [6]. Steganography is a method of concealing private messages in digital media (images, audio, video, and text, for example) so that no one suspects their existence.

Steganography varies from cryptography in that cryptography is concerned with keeping the contents of messages secret, whereas steganography is concerned with concealing the presence of the message. Both techniques are effective at keeping data from unauthorized access, but neither is perfect and can be exploited. The functionality of steganography is mostly defeated once-hidden information is revealed or even suspected. Hybridizing steganography with cryptography should increase its strength [7].

Cover message, hidden message, secret key, and embedding method are the four essential terminology used within steganography systems. The data or information that must be buried in the proper digital media is referred to as a secret message. The cover message, on the other hand, is thought to be the carrier of the secret message, which might be a picture, video, text, or any other digital material. The most essential element is the embedding algorithm, which may be characterized as a technique or set of concepts for embedding secret information in a cover message to prevent unauthorized access [1].

This paper proposes an Android-based application that gives its users the ability to hide their encrypted sensitive data like personal information (e.g., name, email, age, gender, phone number, weight, height, passwords, and credit card number) or privacy information (e.g., user location) inside a cover image via LSB steganography algorithm for hiding and ASE encryption algorithm for sensitive data encryption.

2. Related Work

The proposed method in [12] depended on the encryption of private information using the encryption key and the XNOR gate, followed by the LSB algorithm hiding the encrypted information in a colour picture. The concealing approach is based on extracting three RGB chromatic channels for each pixel and setting the channel in which the encryption message's bit will be concealed, [9] proposed the combination between both steganography and cryptography techniques to come up with a system that is more robust, hence resistible to attacks. The technique of transforming the information or message into a non-readable file format so that a third party or intruder, excluding the intended recipient, is uninformed of the message's contents is known as cryptography. Steganography is also the act of disguising the message inside a cover media from an unauthorized person's view. This work makes use of the best steganography and cryptography available. Inserting or embedding a message in a cover object, the Least Significant Bit (LSB) method is employed. The encryption employed in this work is known as RSA, which is asymmetric cryptography, [3] suggested a method in which a cover item, especially an image, is used to disguise the message to be delivered. The message is first encrypted using the RSA encryption technique before being inserted in the picture. Following the encryption of the message, the process of embedding or concealing the message in the picture continues. The message is embedded in the movie using the Least Significant Bit (LSB) method. The Peak-Signal-to-Noise-Ratio method was used to analyse the performance (PSNR).

3. Cryptography algorithm

The encryption algorithm of the proposed system explains in this section.

3.1. Advanced Encryption Standard (AES)

AES (Advanced Encryption Standard) is a block cipher having a 128-bit input block and a key length of 256, 192, or128 bits. The number of rounds needed in encryption is determined by the key length, although it has no influence on the overall structure of every round. AES is fundamentally a permutation-substitution network, unlike DES, which is completely reliant on the Feistel structure. At the AES operations, the state array is a 16 byte (4×4) array that is inserted and updated in a series of rounds at AES procedures. The state is comprised of splitting the 128-bit entry block into 16 dividers, and each of which is 8 bits long (16 bytes). From the time the plaintext is introduced till the ciphertext is accessible, the following actions are performed on the state in each round:

1. AddRoundKey: Each round of the AES algorithm includes XOR operation among the array of state and (128-bit) the round's assigned key, with a round key generated from the main key for each round, that is only used in the first round.
2. SubBytes: Every byte in the state array is replaced by a new byte at this point, which would be achieved with the help of a customized table.
3. ShiftRows: With the exception of the initial row bytes, which remain unchanged, shifts all state array's bytes in shift-rows. Once in the second row, twice in the third row, and three times in the fourth row, the bytes move to the left. Because the conversion is done on a regular basis, the majority of the state array's bytes are altered.
4. MixColumns: In each round of the AES algorithm, one of the most significant steps is MixColumn, where every column of the state array is subjected to a transformation (This technique produces a linear transformation as a result of the operation) [4].

3.2. Galois Counter Mode (GCM)

Galois Counter Mode (GCM) is a block cipher mode of operation for the AES algorithm that produces authenticated encryption by using universal hashing over a binary Galois Field (GF) [13]. GCM performs encryption using a counter mode (CTR), GCM ensures data confidentiality by utilizing a variant of the Counter mode of operation for encryption. GCM also uses a binary Galois hash function defined over a universal hash function (i.e., finite) field to protect the integrity of sensitive data (up to 64 terabytes per invocation). GCM may also offer non-encrypted data with an authentication guarantee (with essentially unrestricted lengths for each invocation) [5].

3.3. AES-GCM

Galois Counter Mode (GCM) with Advanced Encryption Standard (AES), it introduced by National Institute for Standard and Technology (NIST).

Plaintext P, initialization vector IV or nonce, AES key K, and additional authenticated data (AAD) or associated data are encryption inputs. The decryption data produced by AES-GCM is Authentication Tag T and ciphertext C.

Tag lengths range from 128,120,112,104, and 96 bits, in our research we chose length of Authentication Tag 128 bits. The AAD isn't encrypted but it's used to compute Authentication Tag.

The AES-GCM algorithm is a two parts Encryption and Authentication, in the encryption part is AES encryption using counter mode (CTR), and authentication part is Galois Hash (GHASH) to calculate Authentication Tag.

Figure 1. display the implementation of AES-GCM [15].

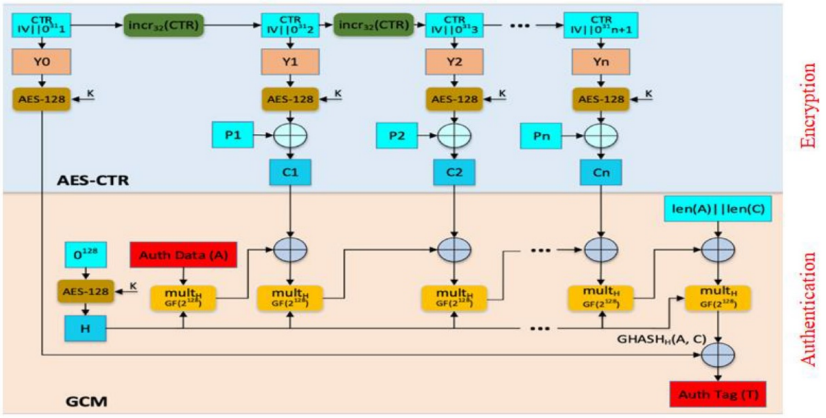

Figure 1: AES-GCM Implementation [15].

4. Steganography algorithm

4.1. Least Significant Bit (LSB)

The Least significant bit is the simplest hiding method, it converts the data to be hidden into a binary representation, then writes the binary representation in the LSB of the carrier's bytes. it results in a very slight change in the image that is not visible to the naked eye.

for example, we'll be working with 24-bit PNG images, which include one byte for each of the red, green, and blue channels.

the digital image is a matrix of the small element named pixels, In a true-color image, each pixel use three bytes (eight for red, green, and blue) and each byte represents the intensity of the RGB colors (0-255).

Then each character in the message is converted to binary. The final bit on the right side of the cover image is substituted with the secret message bit to be disguised by LSB, resulting in the secret message bits being in the 8th bit of each byte of the cover image, as illustrated in (Figure 2).

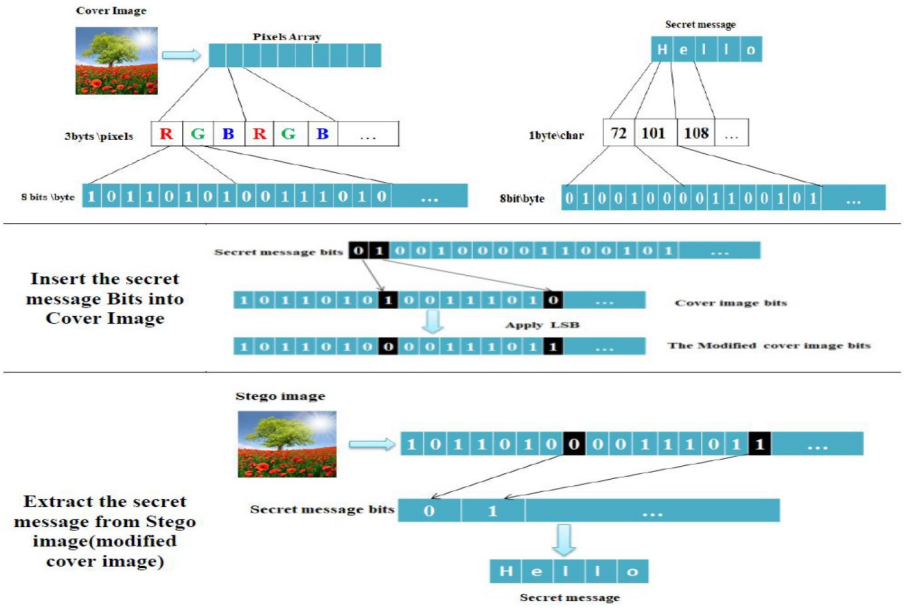

Figure 2: LSB Steganography Implementation

5. Mobile Computing

Mobile computing is a modern technology that has a wide range of applications.; It's also a popular issue in the field of computer science research. It's about how to provide high-quality information services (data storage, query, calculation, and so on) to mobile users (including laptop, mobile phone, and pager users) in a variety of locations. It's a brand-new technology that allows computers and other information devices to communicate and receive data without having to be physically connected to another device. Mobile computing is rising, and it's already being used in a variety of fields, thanks to the rise in mobile device usage [10].

6. Android

Android is an open-source operating system means that its available, and anyone can use or modify it, it was developed by Google. It is a Linux-based operating system. It basically designed for mobile devices (e.g., smartphones and tablet computers) [14]. But is currently used in different devices such as televisions, watches, and cars, etc. It supports a large number of applications on mobile devices. Java programming language is mostly used to write an android cod [14], In addition to Kotlin programming language.

7. Proposed Work

The proposed system presented in this paper is a critical part after the implicit authentication process, this application is divided into two categories: Cryptography and Steganography. It's an android based application that works on smartphones, and it is the second level of security after the implicit authentication usage, meant If implicit authentication allows an unauthorized person to access sensitive data, the proposed system will work as follows:

Initially, the sensitive data are split into three parts as follows:

- Personal information (such as phone number, age, weight, height, gender).
- Privacy information (such as user location and Wi-Fi connections).
- Profile information (such as name, email, and user id).

Then bring sensitive data mentioned above, Profile information bring from Google Sign-In, user privacy information bring from the actual location of the user using (GPS), while Wi-Fi connections bring programmatically from an android phone, and the Personal information bring from TEXT file or SQLite database named "Users.DB"

This SQLite database is associated with an activity that appears only once when the application is launched for the first time. It is used registration to fill the user information such as user name, password, phone number, and another email to send an alert when someone tries to violate privacy.

When a user opens the application for the first time, the registration activity appears to insert the user information. If implicit authentication specifies that the authorized user is using the phone, they will continue to use the phone as normal and can use the application containing the sensitive data as normal, while if the Implicit authentication failed and allow an unauthorized user to access the sensitive data in this phone the application moves the unauthorized user to the login activity to insert the user name and password, the user has three attempts to enter the password if these attempts fail, it sends an alert message to the backup email(from User. DB) and then exits the application, if the password matched, move the user to the next activity that takes the user to either the display

activity or the hiding activity, the display activity displays the sensitive data (privacy, personal, profile information) when the user chose one of the sensitive data you must enter the password to check the user identity, the user has three attempts to enter the password if these attempts fail, it sends an alert message to the backup email and then exits the application, if the password matched, It will display the information associated with that choice.

In hiding activity contained encode and decode, in encode function the sensitive data are encrypted using AES-GCM algorithm with its key then hiding the encrypted message using LSB algorithm as in the following algorithms:

Algorithm 1: "Encrypt and hiding the sensitive data"
Input: Cover Image Img (H,W)
Output: Steganography Image Img_Stg (H,W).
begin
Step 1: bring the sensitive data in the current mobile, the sensitive data split as below:

- Personal information (such as phone number, age, weight, height, gender).
- Privacy information (such as user location and Wi-Fi connections).
- Profile information (such as name, email, and user id).

Step 2: Convert all data in Step 1 as a single Plain text variable **P1**.
Step 3: Generate in a random manner the **AES-GCM** encryption key **K**.
Step 4: Encrypt **P1** using **AES-GCM** algorithm with its key **K** as below:
C1=AES-GCMK(P1).
Step 5: Hide the ciphertext **C1** in cover image **Img** using **LSB** algorithm to produce **Img-Stg**, Img was picked as the cover image from the mobile gallery.
Step 6: Save **Img_Stg** in the mobile gallery.
End

Then, in the decode function, Using the LSB approach, the hidden data extract from the cover image, then decrypt with the AES-GCM technique to acquire and expose the sensitive data. As seen in Algorithm 2.

Algorithm 2:"Extraction and decryption of hidden text"
Input: Steganography Image Img_Stg (H,W).
Output: Sensitive data.
begin
Step 1. Choose Steganography image **Img_Stg** from the gallery
Step 2. Extract bits of the hidden message **C1** from Steganography image **Img_Stg**.
Step 3. Decode it using **LSB** algorithm.
Step 4. decrypt using **AES-GCM** algorithm using its Key **K** as below:
P1= AES-GCMK(C1).
End

Figure 3 displays the block diagram for the suggested system's main phases as below:

Figure 4 displays the implementation of the proposed system as below:

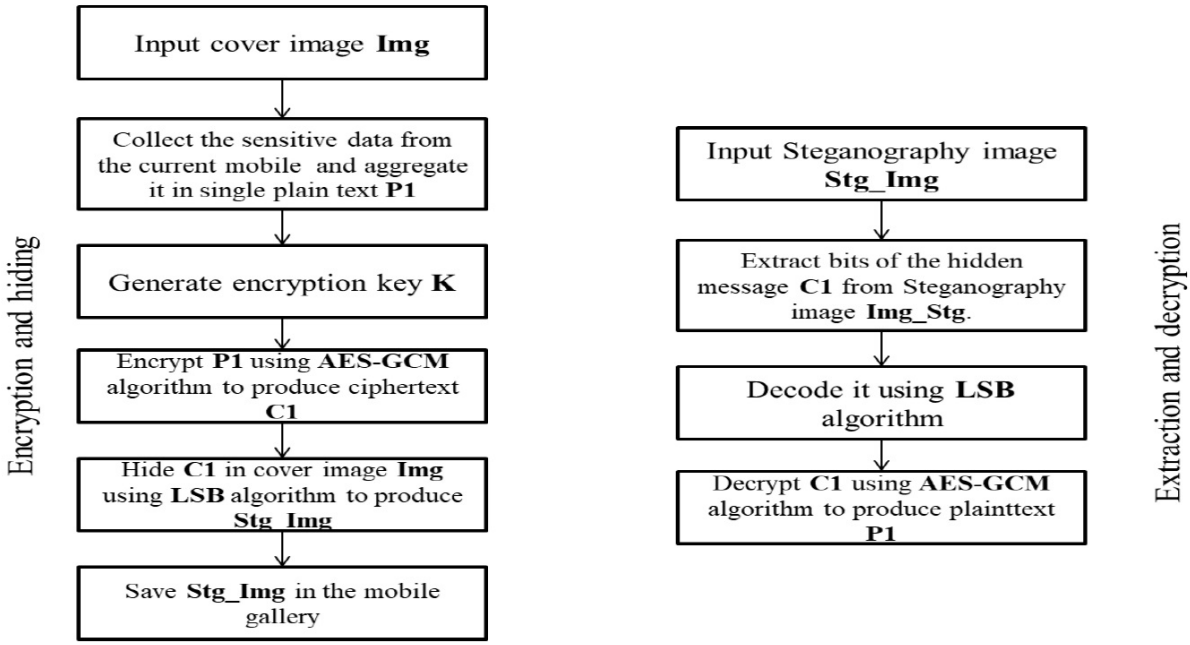

Figure 3: A Block diagram of the suggested system's key steps.

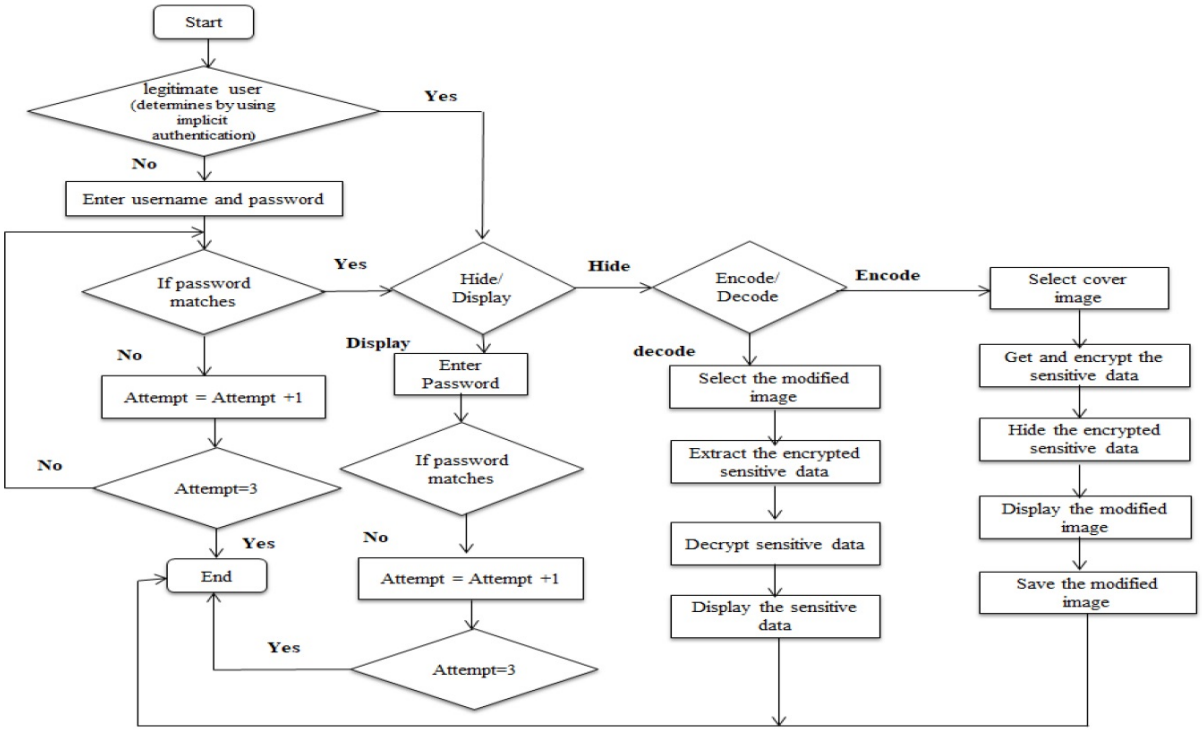

Figure 4: Details for the proposed system implementation

8. Result Analysis and discussion

This application develops and implements using Software Development Kit (SDK) and Integrated Development Environment (IDE), it was developed by using a java programming language (it's an object-oriented language and used to develop the application in mobile platforms), It's compatible

with any device that runs the Android operating system. This application was developed using the Android Studio application development tool and tested using Android Virtual Device (AVD). Next, the physical and software requirements are shown sequentially:

Android Studio 4.1.2 (JDK Toolkits)
16.0 GB RAM
PC processor: Intel(R) Core(TM) i7-10750H @ 2.60 GHz 2.59 GHz
JDK Version: Java Development Kit (JDK) 8 or higher
System type: x64-based processor, 64-bit operating system
Windows 10 is the operating system.

As a cover image, many images were utilized, the resulting Steganography image was almost similar to the cover image as shown in (Figure 5).

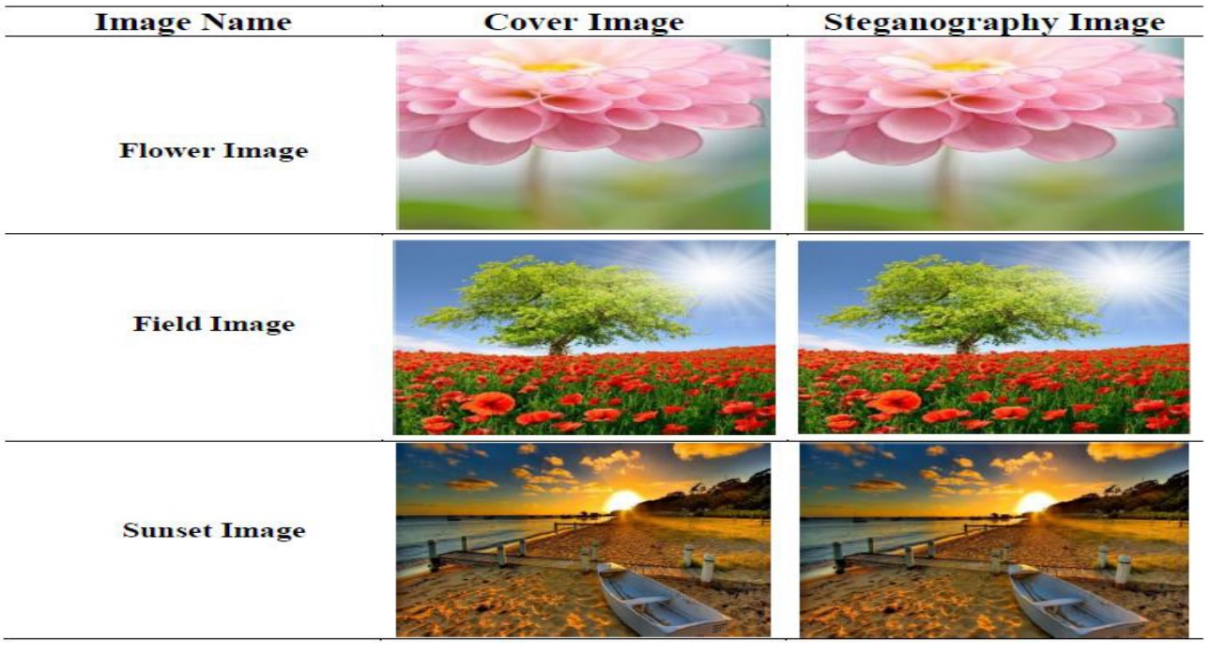

Figure 5: Cover and Steganography images used in this experiment

To evaluate the efficiency of the suggested approach, Peak Signal-to-Noise Ratio (PSNR), Pearson Correlation coefficients (PCC), and Means Square Error (MSE) are employed as metrics for evaluation. PSNR and MSE are used to determine the distortion ratio between the original image and steganography image.

- Means Square Error (MSE) is the simplest estimator to quantify image quality, which is one of the "full-reference" evaluation metrics The cumulative squared intensity error discrepancies among the original image and a reconstructed image are averaged to calculate MSE. When MSE is little, the reconstructed picture quality is better; when MSE is large, the reconstructed image quality is poor. It's calculated through the following equation [8].

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} | \sum_{j=1}^{N} (X(i,j) - \hat{X}(i,j))|^2 \tag{8.1}$$

- Peak Signal-to-Noise Ratio (PSNR) is the ratio of the highest signal strength in the original image to the power of distorting noise calculated using MSE. PSNR is a metric that measures how humans react to image quality [8].

When the PSNR number is higher, the quality of the steganography image improves, PSNR is calculated through the following equation [8]:

$$PSNR = 10 \log 10 \left(\frac{L^2}{MSE} \right) \tag{8.2}$$

- Pearson Correlation coefficients (PCC) the strength of the purported linear relationship between the variables in the issue is represented by the correlation coefficient. It's a quantity factor with a range of values ranging from -1 to +1. A correlation coefficient of -1 or +1 implies the existence of a perfect linear relationship, whereas a correlation value of zero denotes the lack of a linear link between two continuous variables. The strength of a relationship can range from -1 to +1. As the association grows stronger, the PCC approaches one. If the coefficient is positive, the variables are directly related. As a coefficient is negative, the variables are inversely related (that is, when one variable's value rises, the other tends to decline) [11].

Pearson Correlation coefficients is calculated through the following equation [11]:

$$r = \frac{\sum_{i=1}^{n}(X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X - \bar{X})^2 \sum_{i=1}^{n}(Y - \bar{Y})^2}} \tag{8.3}$$

The results from the experiment were deduced and tabulated as follow:

Table 1 presents MSE and PSNR values of the Cover and Steganography Images in order to measure the distortion ratio between the cover and steganography images.

According to the results, the PSNR value between the cover image and the Steganography image is considerable. The quality of the steganography image improves as the PSNR number increases. In addition, the MSE value between the cover and the Steganography images is low (approaching zero), suggesting that the Steganography image is of high quality. According to the data in this table, a Sunset image has the best MSE and PSNR.

Table 1: MSE and PSNR Results.

Image Name	**Size**	**MSE**	**PSNR**
Flower Image	69 KB	0.008075966042154567	69.05885876715166
Field Image	81 KB	0.015822230014025246	66.13812667134025
Sunset Image	27 KB	0.005245395127748069	70.95472149737718

Table 2 shows the correlation coefficient between plaintext and ciphertext. To ensure the success of the encryption process, the PCC used for measuring the scattering ratio between the plaintext and the ciphertext, and the results were always zero, indicating no correlation between the original text and the ciphertext. And the encryption process's success utilizing the AES-GCM algorithm

9. Conclusions

To evaluate the system's performance, the proposed system was tested on numerous images. The proposed system's aim is to protect sensitive data by hiding it within a cover image after implicit authentication failure and allowing an unauthorized user to access sensitive data. The PSNR, MSE, and PCC values demonstrated the effectiveness of the suggested method, PSNR, and MSE values have been measured between the cover and steganography images to measure the steganography images quality, when the PSNR value is as large as possible and the value of MSE is as small as

Table 2: Presents the Pearson correlation coefficient between plaintext and ciphertext.

Plaintext (The sensitive data)	**Ciphertext**	**Pearson Correlation Coefficients**
Baghdad/Iraq/Iraq/44.3094354 Network is:"AndroidWifi" SSID is:""AndroidWifi"" Mousa Ahmed Hassan M.A.Hassan1995@gmail.com 107962208469343656598 07505623610 Health information: name: Mousa Ahmed age: 26 weight: 65 length: 170 gender: male CreditCard Number and Passwords: Masterdcard 5412 7512 3412 3456 facebook password fb123595 instagram password in1234595 email password gm1234595	WgmFrPqAwOJk2/QXVoY9LjnEwc0MzJ aw8PmzWP3R/eKK4inlXojTk6zg5Re/Mri Bpc3Ne/HMxvI+Rrm8OPF4H7jx1noMIA oX0sREiX5Ie6w4OrPFsGM1V7lW0QB1 PMmWM7TdioGPWEzQLhKB8dotEZDL voeg9td1SEramUg9CxF9cjeDvwyAmtrDz ICDWl8KdMYeI2wPSLSVwuoDegVUEf vohGHi5UWqqnzOCCMScpgQKHH9Qq DhGWvpDRE0x9vuZylWl8P88IQ5AmD Kwi0kKMc01OCPkPtAgjUCLoMRPQvP pmlq0bt4q3uBDrkhZ27JsGdZWpqLc2S/F bL5PSg2JaBsjwkrMdgV8BCvS+RDs240 wQcTmSY51wqWy1lyPnxTxlHeI91BOB ZNVJDF56AgMlmIzB0r7ss3Y5CgYmnqf taKgN1cJegVFc+ppRxhdSYoo5C1qFpXI aywqT4p9rbBVRZi3pe/qt30VGBqT2A/os X3PNITMGtod0ObLo+kiI/5zCrzP7OiV2 V3HY2clBhq31t0cKss3uUlZ+vqqO8=	0

possible, which means that the distortion ratio of steganography image is small. The whole encrypted sensitive data was also retrieved from steganography images. and use the PCC between plaintext and ciphertext to measure the ratio of scattering and to show that there is no correlation between the original and ciphertext, Because IV and AES-GCM key change, the ciphertext value changes, but the resulting PCC values are always zero, indicating no correlation between the original text and the ciphertext and the scattering ratio is high, so the encryption process's success utilizing the AES-GCM algorithm with key length 256 bit. Therefore, the proposed method is considered as second-level security that provides protection to the sensitive data from unauthorized access by another person's and it provides adequate protection without affecting image quality. And also it's used to achieve confidentiality, integrity, and authentication using the AES-GCM algorithm, and it does not consume the phone's memory and is relatively fast. In summary, the best results were achieved for the test images (Sunset Image and Flower Image), where PSNR was high and MSE was low.

References

[1] N.H.M. Ali, A.M.S. Rahma and A.S. Jamil, *Text hiding in color images using the secret key transformation function in GF (2 n)*, Iraqi J. Sci. 56(4B) (2015) 3240–3245.

[2] A. Ali, A.-H.S. Saad and A.H. Ismael, *VRNFC-Stego: Data hiding technique based on VR images and NFC-enabled smartphones*, Procedia Comput. Sci. 171 (2020) 1551–1560.

[3] R. Apau and C. Adomako, *Design of image steganography based on RSA algorithm and LSB insertion for android smartphones*, Int. J. Comput. Appl. 164(1) (2017).

[4] A.S. Bader and A.M. Sagheer, *Modification on AES-GCM to increment ciphertext randomness*, Trans. Int. J. Math. Sci. Comput. 4 (2018) 34-40.

[5] M.J. Dworkin, *Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac*, NIST Special Publication 800-38D, (2007).
[6] R.J. Essa, N.A. Abdullah and R.D. AL-Dabbagh, *Steganography technique using cenetic algorithm*, Iraqi J. Sci. 59(3A) (2018) 1312–1325.
[7] E.S.I. Harba, *Advanced password authentication protection by hybrid cryptography & audio steganography*, Iraqi J. Sci. 59(1C) (2018) 600–606.
[8] A.A. Ibrahim, L.E. George and E.K. Hassan, *Color image compression system by using block categorization based on spatial details and DCT followed by improved entropy encoder*, Iraqi J. Sci. 61(11) (2020) 3127–3140.
[9] R.K. Kyei, J.K. Panford and J.B. Hayfron-Acquah, *Enhancing data security in android smartphones using image steganography, RSA encryption with LSB insertion*, Int. J. Comput. Sci. Info. Secur. 17(2) (2019).
[10] X. Ma, Z. Wang, S. Zhou, H. Wen and Y. Zhang, *Intelligent healthcare systems assisted by data analytics and mobile computing*, Int. Wireless Commun. Mobile Comput. Conf. 2018, pp. 1317–1322.
[11] M.M Mukaka, *Statistics corner: A guide to appropriate use of correlation coefficient in medical research*, Malawi Med. J. 24(3) (2012).
[12] R.M. Neamah, J.A. Abed and E.A. Abbood, *Hide text depending on the three channels of pixels in color images using the modified LSB algorithm*, Int. J. Electr. Comput. Eng. 10(1) (2020).
[13] M. Rodríguez, A. Astarloa, J. Lázaro, U. Bidarte and J. Jiménez, *System-on-programmable-chip AES-GCM implementation for wire-speed cryptography for SAS*, Conf. Design of Circuits and Integrated Systems (DCIS), IEEE, 2018, pp. 1–6.
[14] R. Singh, *An overview of android operating system and its security features*, J. Eng. Res. Appl. 4(2) (2014) 519–521.
[15] Y. Sovyn, V. Khoma and M. Podpora, *Comparison of three CPU-Core families for IoT applications in terms of securityand performance of AES-GCM*, IEEE IOT J. 7(1) (2020).