Research paper © 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019s

An improved steganography model for efficient data integrity and confidentiality

V Bhavani,

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. vasanthabhavani@kluniversity.in

A.Roshini

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India. vasanthabhavani@kluniversity.in

Abstract:

In the contemporary era, ensuring secure and confidential data transmission poses a significant challenge, especially in the realm of networking. This paper presents a methodology aimed at bolstering data integrity and confidentiality by extending existing steganography techniques and leveraging encryption and decryption methods such as AES128, SHA256, and Image Steganography Cryptographic Methods. The proposed approach entails combining encrypted plain text and a hash code within an image to secure data before transmission to the end-user. In the event of an attack, even if successful, the attacker would only gain access to a RAW message without knowledge of the plain text. This insight opens avenues for further advancements in various research areas.

Keywords: Encryption, SHA256, AES128, Image Steganography, Confidentiality, Integrity, Cryptology

1. Introduction

Data confidentiality is paramount in today's interconnected world, particularly during data transmission over networks. Encryption[1] algorithms like AES play a crucial role in safeguarding sensitive information, ensuring that only authorized parties can access it. However, ensuring data integrity, the authenticity of the transmitted information, is equally important[9]. Hashing algorithms like SHA256 and SHA512 excel in this regard, providing robust verification mechanisms. Traditionally, encryption algorithms and hash functions have been employed independently. Plaintext is encrypted using encryption algorithms, generating ciphertext, while the hash function produces a hash value based on the original plaintext. Both the ciphertext and hash value are then combined and transmitted to the receiver[5].

Image steganography, a technique for hiding data within images, offers additional layers of security. By embedding plaintext within an image, the steganographic image becomes the transmission medium. While this approach enhances confidentiality, it introduces the risk of data leakage if an attacker intercepts and analyzes the steganographic image.

To address these challenges, integrating image steganography with a hash function provides a comprehensive solution for secure data transmission and integrity enhancement. This involves encrypting the plaintext using an encryption algorithm to obtain ciphertext and simultaneously generating a hash value from the original plaintext using a hash function. The ciphertext and hash value are then combined and embedded within an image using a steganographic algorithm, resulting in a steganographic image transmitted to the receiver.

This combined approach fortifies both data confidentiality and integrity. If an attacker intercepts the steganographic image, steganalysis techniques would likely only yield either a portion of the hash message or a fragment of the ciphertext, but not both. Decrypting[4] the complete message from the image alone would be highly improbable, safeguarding confidential communication between parties without the knowledge of a third party.

The hash function plays a dual role in this model, ensuring both confidentiality and integrity. The hash value serves as a unique identifier for the original plaintext, enabling message integrity verification at the receiver's end. Even if the attacker successfully decrypts the ciphertext, modifying the message without altering the hash value would be impossible. Upon receiving the steganographic image, the receiver decrypts it, extracts the concatenated message (ciphertext and hash value), and passes it through the hash function. If both hash codes match, it signifies that the message has remained unaltered during transmission.

This integrated approach has wide-ranging applications across various sectors, including the military, where secure data transmission is crucial, and healthcare, where patient details are often transmitted through patient images. This project distinguishes itself by offering a unique approach that simultaneously enhances both confidentiality and integrity, making it a valuable tool for safeguarding sensitive information in diverse domains.

The second module of the project focuses on decryption, where the concatenated message, consisting of ciphertext and hash value, is decrypted. The decoding process differs from conventional approaches in that the decrypted message is not the plain text itself but the concatenated message. The decoded image yields the concatenated message, the output from the initial module. The concatenated message is then split into its two components: the ciphertext and the hash value. The ciphertext[3] is subsequently decoded and sent to the hash-finding file, where the corresponding hash is obtained and saved. Comparing these two hash values, one derived from the decoded plaintext and the other obtained from the plaintext generated in the encoding module, indicates whether the message has been altered between the encryption and decryption phases.

2. Literature Survey

Cryptography encompasses the encryption or decryption of plain text through cryptographic algorithms [2], playing a pivotal role in secure communication between two parties over an unsecured network [11]. The effectiveness and confidentiality of message transmission hinge on the robustness and security of cryptographic functions. These algorithms often employ intricate mathematical operations to establish a high level of security challenging for attackers to breach. Presently, cryptographic algorithms fall into two categories: symmetric-key and asymmetric-key cryptography [2], each serving distinct purposes and providing varying levels of security for data transmission.

To execute encryption or decryption, possession of the corresponding key is imperative. The sender acquires this key during the encryption of plain text using a specific key, and the key transfer can occur in person or through a secure channel [1]. In asymmetric-key cryptography, the challenge posed by variable-length keys is addressed through techniques such as hashing, which converts variable-sized data into a fixed-size output. Various hashing algorithms, including SHA1, SHA12, SHA0, among others, serve this purpose, generating a unique message digest applicable as a password or for data authorization [6]. Hashing finds widespread applications, contributing an additional layer of security that significantly impedes progress if an intruder attempts to break the hash algorithm.

The expansion of computer usage has paralleled the growth of cyberspace, influencing the development of information theory, logical prowess, and steganography [14]. The adoption of steganography has led to a proliferation of compelling applications [2,3], presenting a range of intriguing possibilities. While cryptography primarily secures text-based data susceptible to interception, steganography, with its data embedded in pixel form, poses a formidable challenge to decoding, thereby enhancing the security of the encoded image [8].

Steganography is a technique employed to conceal data within various formats, including images, audio, videos, and more, with its primary security function being data hiding [4]. Additionally, steganography ensures confidentiality by utilizing a specific key known only to the sender and receiver, preventing any potential data leakage [13]. These security measures underscore the inherent protective advantages associated with the use of steganography.

3. State of Art and Algorithm

In this proposed approach, a secure graphical user interface (GUI) is introduced to encrypt data prior to transmission over the network. Initially, the plaintext undergoes encryption using a robust algorithm such as AES 128-bit, resulting in ciphertext. Simultaneously, the same plaintext is subjected to a hash algorithm, producing a distinctive message digest through SHA-256. The combined message, comprising the ciphertext and the unique message digest, is then fed into a steganography tool. This tool prompts the selection of an image to store the information, ultimately generating an encrypted Stego Message. Through this process, we enhance security, achieving both confidentiality and integrity safeguards. Our project enables efficient data concealment and transmission through the incorporation of data into an image. This is achieved seamlessly through a unified application that securely embeds data into images, thereby enhancing both the integrity and confidentiality of the image.

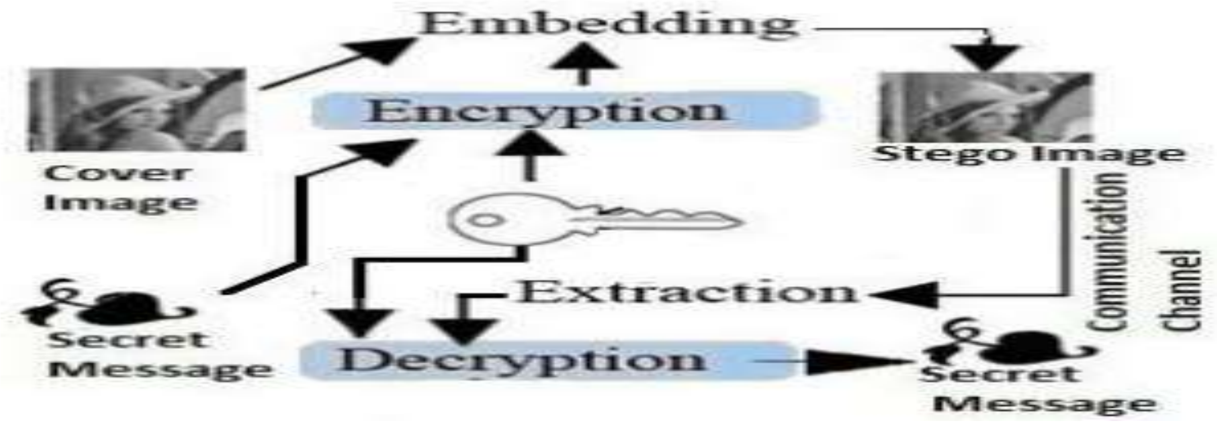

Fig.1. Plain text encryption using asymmetric technique

In the illustrated block diagram, we can observe the improvement of data integrity achieved through the utilization of encryption algorithms. The initial block illustrates the plaintext and the key, which undergo encryption using the Advanced Encryption Standard 128-bit algorithm, resulting in the creation of ciphertext. Concurrently, the plaintext undergoes the SHA-256 hash algorithm, producing a unique message digest. The second block outlines the combined message of the ciphertext and a distinct message digest. This amalgamated message is then processed through a steganographic tool alongside an image, generating the binary code corresponding to the combined message.

In the illustrated diagram, the image undergoes decoding, and the combined message is forwarded to another file for further processing and separation. Subsequently, the combined message is split into two parts: ciphertext and hash value. The ciphertext is decoded and directed to the hash-finding file, where the hash is extracted and stored in the hash file. Following that, the hash value obtained from the decoded plaintext in the decryption module is compared to the hash value derived from the plaintext in the encoding module. If the two hash values match, it indicates that the message remains unchanged from the encryption phase to the decryption phase.

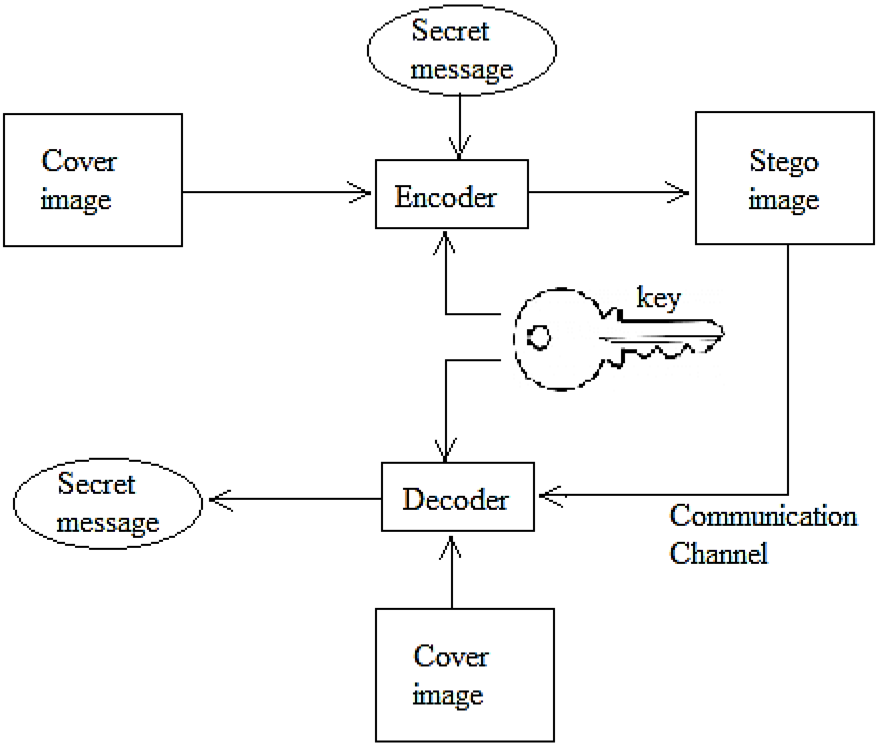

Fig.2. Decryption of Cipher text using stego image

While steganography is generally considered a secure method for data hiding with no known breaches, current practices involve sending plaintext directly to a steganographic tool and embedding it in an image or encrypting the plaintext before embedding it. The first approach lacks confidentiality, and though the second ensures confidentiality, it lacks an integrity check [12]. To address these issues, our proposed system offers enhanced performance by achieving both confidentiality and integrity in data hiding. Our approach becomes particularly relevant in scenarios like WhatsApp backup, where despite end-to-end encryption between the sender and receiver, there is no encryption during user data or chat backups. This creates a potential risk of data exposure during backup processes. Our data hiding method provides a secure procedure for users to transmit important chats, generating a secure image for data backup. This mitigates the risk of data exposure and ensures confidentiality. In this way, our system proves beneficial to users by enhancing data security during backup procedures.

Algorithm.1: Enhanced steganography algorithm for image encryption and Decryption

Input Plain Text: Enter the plain text that needs to be encrypted.

Enter Encryption Key: Provide a key for encrypting the entered plain text.

AES Encryption: Navigate to the encryption button to initiate the AES encryption algorithm with a 128-bit key.

Generate Message Digest: Enter a plain text identical to the one used for encryption to create a unique message digest.

Hash Generation: Click on the hash button to generate a unique message digest.

Steganography Encoding:

Concatenate the ciphertext and the unique message digest for steganography encoding.

Access Steganographic Tool:

Click on the "Stego" button to access the steganographic tool.

Choose Image: Select an image by clicking on "Choose File."

Paste Concatenated Message: Paste the concatenated message (ciphertext and hash value) into the designated text area.

Hide Data: Click on the "Hide" button to generate the binary representation of the data.

Embed Data in Image: The binary representation of the data will be embedded into the image.

Obtain Output Images: Obtain three images as output: the original image, normalized image, and the encoded image.

Download Encoded Image: Click on the encoded image to download it.

Steganographic Decryption: Use the encoded image for steganographic decryption by uploading the image and clicking on the "Decode" button.

Retrieve Concatenated Message: The concatenated message (ciphertext and hash value) is obtained as output.

Advanced Decryption: Process the ciphertext using an advanced decryption algorithm to obtain the plaintext.

Generate Message Digest (SHA-256): Pass the obtained plaintext through the Secure Hash Function SHA-256 algorithm to generate a unique message digest.

Hash Value Comparison: Compare this hash value with the existing hash value sent to the receiver.

Verification: If both hash values match, it confirms that the message has not been altered, ensuring integrity and confidentiality.

4. Results

Initially, we acquire the plaintext and key via the encryption-specific GUI. Utilizing the Advanced Encryption Standard (AES) 128-bit symmetric encryption algorithm, encryption is carried out, resulting in a variable-length ciphertext that is robust and resistant to tampering. This ciphertext is crucial as it needs to be combined with the hash value generated in the subsequent step.

Next, we utilize the same plaintext employed in the first step to generate a unique message digest. This involves applying the secure hash function SHA-256 hash algorithm, which accepts variable-length input and yields a fixed-size output. The resultant unique message digest is secure, irreversible, and ensures a distinct hash.

The output ciphertext obtained in step 1 and the unique message digest obtained in step 2 are combined and sent to the image steganographic tool.

Upon accessing the steganographic tool, select the image by clicking the "Choose File" button. Choose the image where you intend to conceal the concatenated message of the ciphertext and the unique message digest. Once the image is selected, paste the concatenated message. Paste the ciphertext and unique message digest into the provided text area. Subsequently, click the "Hide" button to conceal the concatenated message within the selected image. Following the hiding process, the binary representation of the image will be generated.

Three images will be generated: the original image, normalized image, and encoded image. Download the encoded image, which contains embedded binary data. This encoded image can then be transmitted to the receiver for decryption. The encoded image serves as the primary input for the decryption process. Use the steganographic decoder tool by clicking the "Choose File" button, selecting the encoded image, and then clicking the "Decode" button. The initial concatenated text of the ciphertext and the unique message digest is then produced as the output.

The ciphertext undergoes decryption through the AES decryption algorithm to obtain the plaintext. Subsequently, the plaintext is processed using the secure hash function SHA-256 algorithm to generate a unique message digest. The obtained hash value from the concatenated message is then compared with the hash value generated from the plaintext. Confirming their match assures that the message remains unaltered, thereby enhancing both confidentiality and integrity of the message.

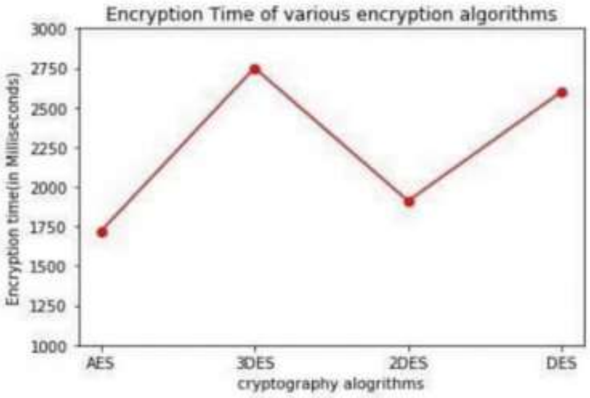

Fig.3. Encryption Algorithms for varied timeslots

5. Conclusion

Cryptography stands as a robust security mechanism providing comprehensive security services. Within this proposed methodology, we have successfully achieved both confidentiality and integrity for plain messages before their transmission over an open network. Additionally, we have identified and addressed the drawbacks of existing systems. The introduced system has the potential to contribute to further advancements in various research areas within Cryptology.

References

[1] Singhal, Manav, and AnupamShukla. "Implementation of Location based Services in Android using GPS and Web Services." IJCSI International Journal of Computer Science Issues 9, no. 1 (2012).

[2] Kushwaha, Amit, and VineetKushwaha. "Location Based Services using Android Mobile Operating System." International Journal of Advances in Engineering & Technology 1 (2011): 14-20.

[3] Rani, ChRadhika, A. Praveen Kumar, D. Adarsh, K. Krishna Mohan, and K. V. Kiran. "Location Based Services In Android." International Journal Of Advances In Engineering & Technology (2012).

[4] Ejiagha, Ifeanyi R., Johnbusco C. Ojiako, and Chijioke G. Eze. "Accessibility Analysis of Healthcare Delivery System within Enugu Urban Area Using Geographic Information System." Journal of Geographic Information System 4 (2012): 312-321.

[5] C.Prabha,R.Sunitha,R.Anitha. "Automatic Vehicle Accident Detection and Messaging System using GSM and GPS Modem." International Journal of Advances in Engineering & Technology Issues 7, no. 3 (2014).

[6] Chenshu wu, Zheng Yang, Yu Xu, Yiyang Zhao, Yunhao Liu. "Human Mobility Enhances Global Positioning Accuracy for Mobile Phone Localizaion. " IEEE (2015).

[7] Shanu Agrawal, Pradeep Majhi, Vipin Yadav." Fingerprint Recognition based Electronic Voting Machine." International Journal of Engineering & Technical Research.

[8] A.Xavier Raj." Saving lives through rural ambulance services: Experiences from Karnataka and Tamil Nadu states, India. " Transport and Communication Bulletin for Asia and the Pacific (2014).

[9] Jie Liu, Bodhi Priyantha, Ted Hart, Yuzhe Jin, Woosuk Lee, Vijay Raghunathan, Heitor S. Ramos. [11]"Co-GPS: Energy Efficient GPS Sensing with Cloud Offloading." IEEE (2015).

[10] Augusto Luis Ballardini, Lorenzo Ferretti, Simone Fontana, Axel Furlan, Domenico Giorgio Sorrenti. "An Indoor Localization System for Telehomecare Applications." IEEE (2015).

[11] Sri Krishna Chaitanya Varma, Poomesh, Tarun Varma, Harsha. "Automatic Vehicle Accident Detection and Messaging System using GPS and GSM Modems." International Journal of Scientific & Engineering Research, Issues 8, no. 4 (2013).

[12] S. L. R, S. M, S. K, and A. Thilagavathy, "AI-Powered Smart Glasses for Blind, Deaf, and Dumb," 2022 5th International Conference on Advances in Science and Technology (ICAST), Dec. 2002, doi: 10.1109/icast55766.2022.10039557.

[13] V. Rashmi et al., "Experimental Investigations to Fault Reduction System for Software Applications," Ingénierie des systèmes d information, vol. 28, no. 3, pp. 567–573, Jun. 2003, doi: 10.18280/isi.280304.

[14] G. Cao, "Acoustical Measurement and Fan Fault Diagnosis System Based on LabVIEW," Practical Applications and Solutions Using LabVIEW&amp;#8482; Software, Aug. 2011, doi: 10.5772/22216.