Key generation technique by password hashing using AES encryption for hiding cipher text within image

Karthikeyan. B
School of Computing
SASTRA Deemed University
Thanjavur, India
mbalakarthi@gmail.com

Kishore. M. S
School of Computing
SASTRA Deemed University
Thanjavur, India
mskishore.03@gmail.com

Sri Hari. R
School of Computing
SASTRA Deemed University
Thanjavur, India
sriharionahigh@gmail.com

Jaya Prakash. T
School of Computing
SASTRA Deemed University
Thanjavur, India
jai82889@gmail.com

Abstract— ***The process of concealing data in the form of text, images, or videos behind a cover picture, video, QR code, or even audio so that it is invisible to the naked eye is known as steganography. This paper gives a detailed literature review of hiding text within a PNG cover image by integrating Advanced encryption standard (AES) algorithm and to ensure highest visual clarity, the embedding process prioritizes least significant bit (LSB) modification for intensity per pixel. Along with the methodology, experimental set ups have been conducted to quantitatively assess the security and quality aspects of the proposed Steganography encryption technique by applying Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) scales for varying character sizes. The investigation showcases the effectiveness of the AES-based steganographic approach in ensuring powerful encryption while maintaining the original image quality. In addition to the methodology and results, a table summarizing the details and a proper flow diagram has been included for better understanding for future researchers.***

Keywords—*Steganography, AES Algorithm, PBKDF2 Password encryption, Steganalysis, LSB Embedding, XOR operation*

I. INTRODUCTION

In the contemporary era, information plays a pivotal role in various domains such as banking, sports, and education, necessitating effective information security measures for secure communication. Cryptography serves as a crucial component in safeguarding information content, permitting only the intended sender and recipient to access the communication's contents. Information hiding techniques, including steganography and watermarking, are employed to conceal sensitive information within seemingly innocuous media like images, videos, audio, and text [1]. Steganalysis, the process of detecting concealed information in digital media, is irreversible, ensuring that the original cover image cannot be retrieved once private data is embedded. The term "stegno-image" denotes an image containing hidden communication, and the practice of embedding secret information within cover media is referred to as "steganography." Image steganography stands out as a prominent method for securely concealing information and digital data. The Advanced Encryption Standard (AES), specifically the 256-bit version, has emerged as a cornerstone in cryptographic algorithms, providing a robust foundation for secure data communication and storage. This paper explores the integration of AES-256 and Least Significant Bit (LSB) steganography to enhance data security significantly. While attackers typically target the weakest link in a system, which is often not the encryption itself, AES-256 is considered highly secure. However, the vulnerability lies in the key, which utilizes the Grounded Key Derivation Function (PBKDF2) [2]. The proposed system in this paper advocates for the use of PBKDF2 over other key derivation methods to reduce susceptibility to brute force attacks. With PBKDF2, hackers are limited to making a relatively small number of guesses per second, depending on the configuration, significantly enhancing security. Strengthening passwords is crucial for thwarting attacks such as dictionary and brute force attacks. Normal users can take an equal amount of time to deduce a password by applying a Key Derivation Function (KDF) to a well-chosen password, while the system applies CPU-intensive procedures on the attacker's side. The AES encryption generates a key using PBKDF2, and the data is translated with this hexadecimal key. The resulting cipher text, an encrypted and non-understandable text, is then embedded into an image using LSB steganography. The alteration in bytes is minimal, at 0.0002%, rendering it nearly imperceptible. The final steganographic image conceals secret data effectively [3]. To enhance word protection, a 256-bit encryption key is employed, and an exclusive OR operation is performed on each bit of the plaintext and key to generate the cipher text. This encryption method, incorporating PBKDF2, segregates odd and even characters for word protection, ensuring robust data security with 256 bits. In summary, this paper proposes a comprehensive approach to information security by combining AES-256 encryption, LSB steganography, and PBKDF2 key derivation, thereby fortifying data protection against potential threats and attacks.

II. LITERATURE SURVEY

Steganography facilitates the covert exchange of data through digital communication channels, even in the challenging scenario of continuous surveillance. This method enables the concealed communication and transfer of data without detection, making it possible to hide a secret message within an image [4]. The process of embedding

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

involves modifying specific attributes of alternative media forms like images, audio, or video files, commonly referred to as the cover. The resulting output maintains the characteristics of the original cover media, concealing essential secret data within it. When confidential information is hidden within a cover image, the outcome is termed a steganographic image. This article delves into the dominion of steganography, introducing proposed techniques for creating steganographic images. The Advanced Encryption Standard (AES) functions as an encryption algorithm, employed for encrypting text data in a manner that renders it unintelligible [5]. Acknowledged as the most secure encryption standard to date, with no reported breaches, AES operates as a symmetric key algorithm, applying the same key for both encoding and decoding processes. It employs a “BLOCK CIPHER” with variable sizes, extending from 128 to 256 bits, incorporating various techniques such as XOR operation, substitution, permutation operation, as well as row and column shifts. In the realm of safeguarding personal information and accessing specific resources, user-selected passwords play a crucial role. Consequently, passwords need to be robust enough to withstand known attacks like dictionary and brute force attacks. To enhance security, a key derivation function (KDF) is applied to a user-selected password. This allows valid users to derive keys within a reasonable timeframe while transferring CPU-intensive operations to potential attackers through the PBKDF2 hash.

A. Least Significant Bit (LSB)

Several steganographic techniques are available, with a commonality in the direct alteration of specific bits within image pixel values to conceal information. LSB, a type of Adaptive Spatial Steganography, stands out as one of the simplest approaches. It discreetly hides confidential messages within the least significant bits (LSBs) of pixel values without causing noticeable distortions [6]. The LSB is an ideal choice for concealing information due to the generally imperceptible differences in its value to the human eye. This strategic placement ensures that changes made to the LSBs do not result in conspicuous alterations to the original object. Communication bits can be embedded in a sequential or random manner, providing flexibility in the concealment process. In summary, LSB-based steganography proves to be a straightforward yet effective method, leveraging the imperceptibility of changes in the least significant bits to conceal information within pixel values. The embedding process, whether sequential or random, allows for the covert integration of data without introducing noticeable deformations to the original object. Embedding operation of LSB steganography may be described by the following equation.

RED	BLUE	GREEN
10000000	10100100	10110101
10110101	11110011	10110111
11100111	10110011	00110011

Table 1: Pixel Grid of 24-Bit

RED	BLUE	GREEN
10000000	10100101	10110100
10110100	11110010	10110111
11100110	10110010	00110011

Table 2: Pixel Grid after applying LSB

$$y_i = 2\left[\frac{x_i}{2}\right] + m_i \qquad (1)$$

where x_i , y_i and m_i are the i^{th} message bit, the i^{th} selected pixel value before embedding and that after embedding respectively.

The Least Significant Bit (LSB) method introduces minimal changes to the image, making it imperceptible to the naked eye. The alteration in a byte is a mere 0.000002%, rendering it nearly negligible. As outlined in Table 1, LSB operates seamlessly with PNG images, where each pixel consists of three bytes (red, green, blue) independently. Using LSB allows the storage of 3 bits of data in a single pixel by modifying one last bit in each color, as illustrated in Table 2. Consequently, LSB proves to be an efficient method for concealing and communicating data covertly. However, if a hacker suspects that the image contains sensitive information, they may attempt to extract the crucial data using steganalysis tools. In such cases, if successful, the pivotal data falls into the wrong hands [7]. To enhance security, the proposed system introduces an advanced level of security to safeguard against such risks.

B. Advanced Encryption Standard (AES-256)

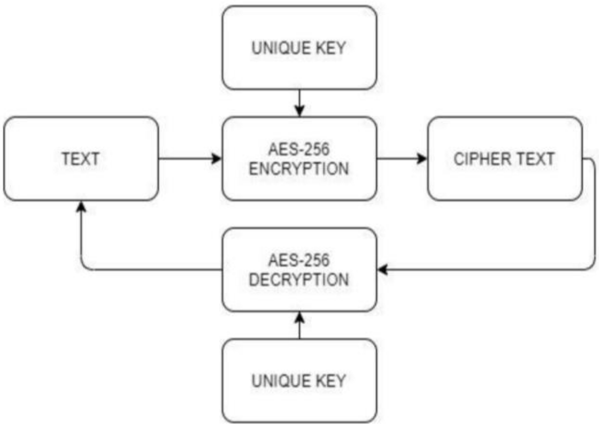


Fig 1: Diagrammatic Representation of AES-256

Using a single secret key for both data encryption and decryption, AES functions as a symmetric key encryption method. The only encryption technique that the US National Security Agency (NSA) has authorized for the protection of highly secret material is the Advanced Encryption Standard (AES) [8]. Advances in steganographic and cryptographic techniques have been produced in response to changing concerns about data security. These developments have made it possible to transmit data in a safe manner while also hiding text and image data. This is accomplished by combining LSB-based image steganography with the AES algorithm. Data substitution using a table shown Fig. 1, followed by row shifting, column mixing, and ultimately, a transformation employing an exclusive (XOR) operation on each column using a unique segment of the encryption key. To convert plaintext into ciphertext, the proposed algorithm makes use of the Sub Bytes, Shift Rows, and Add Round Key technique [5]. After 14 rounds of AES encryption, the technique produces an unreadable ciphertext with a key length of 256 bits, or 32 bytes. Using the same key, the decryption method recovers the original text after 14 rounds of decryption [7]. The length of the hexadecimal key for a 256-bit encryption should be 32 Bit.

C. Combining Steganography with Cryptography

The approach presented in reference [9] suggests enhancing message security by employing a combination of HUFFMAN coding and the AES algorithm. The primary objective is to assess the performance aspects of image visibility using MSE and PSNR values, demonstrating that the entire process is less susceptible to noise compared to alternative algorithms employed for data protection. It is important to use the Modified Least Significant Bit (MLSB) technique in reference [10] in order to conceal messages in images. AES encryption is used to demonstrate dual-layer security for sensitive data by making it difficult for unauthorized parties to decrypt the original data. The authors also give a general introduction to cryptography and survey previous works that use different methods for image encryption. According to simulation results, every algorithm for photo processing has advantages and disadvantages, but overall, all techniques successfully secure images and encrypt them to prevent unwanted access on public networks. In reference [11], four Artificial Neural Networks (ANN) are utilized for implementing steganography in concealing sensitive information within images. The choice of ANN over other neural networks is motivated by the implementation of the Scaled Conjugate Gradient (SCG) technique, ensuring seamless hiding of secret images within container images without introducing errors. Reference [12] addresses a significant drawback of cryptography and steganography techniques. Cryptography's limitation is highlighted as the conversion of encrypted text into a random stream of alphabets, numbers, and symbols. Meanwhile, steganography typically encrypts data in plaintext. The proposed technique suggests using AES encryption to encrypt data and then concealing the encrypted text within an image through Pixel Value Differencing, achieving a double layer of security. It's important to note that this technique is specifically implemented for grayscale images.

D. Data Security using hashing mechanism

Reference [13] illustrates the utilization of the salt hash approach for storing passwords and other confidential information. The researchers assert that salting is the most effective method for enhancing data security. The paper discusses how employing hashed passwords can reduce the vulnerability to dictionary attacks. To overcome hashing, attackers often resort to dictionary attacks. In [14], researchers conducted an analysis comparing the effectiveness of hashing algorithms, including PBKDF2, “BCRYPT”, and “SCRYPT”. The study evaluates the time and complexity associated with password hash generation. Notably, “BCRYPT” is identified as the slowest due to its use of blowfish, while PBKDF2 is the fastest but susceptible to being broken. However, the authors conclude that the “BCRYPT” algorithm or “SCRYPT” is highly resistant to breaking due to their computational strength.

III. PROPOSED METHODOLGY

The proposed methodology is to hide information within an image using AES and securing the shared key for AES using pbkdf2.The key is hidden with the image making it more secure. And on top of that pbkdf2 uses SHA 256. Here we generate keys using the password given by user. We have two separate passwords for both odd and even characters and we will secure the salt and we got another key which is random for each process regardless of password to add more security we used AES 256 i.e. 32-bit keys and AES double encryption. We hided both the salt and entry level encryption key in the encrypted image for more security. We hided the data using LSB 1-bit steganography.

A. Significance of LSB Steganography

LSB steganography is a type of image steganography where messages are hidden inside an image by substituting the bits of the hidden message for the least significant bit in each pixel.

B. Usage Of PBKDF2

PBKDF2 (Password-based key derivation function 2) is used to make key protection more secure. As compared to sha 256 pbkdf2 comes with sha 512 prebuilt. Pbkdf2 gives key based on the salt and our password. As we can randomize the salt and password for key encryption. We can give the desired length for key and we can vary the salt length too. As this is random it’s more secure. And pbkdf2 it uses hash-based message authentication code (HMAC).
It uses key stretching as given in Fig. 2 which makes password cracking more difficult. DK = PBKDF2(PRF, Password, Salt, c, dkLen) As we are using salt it adds another layer of security as they can’t retrieve the password just by using ciphertext. PRF is a pseudorandom function.

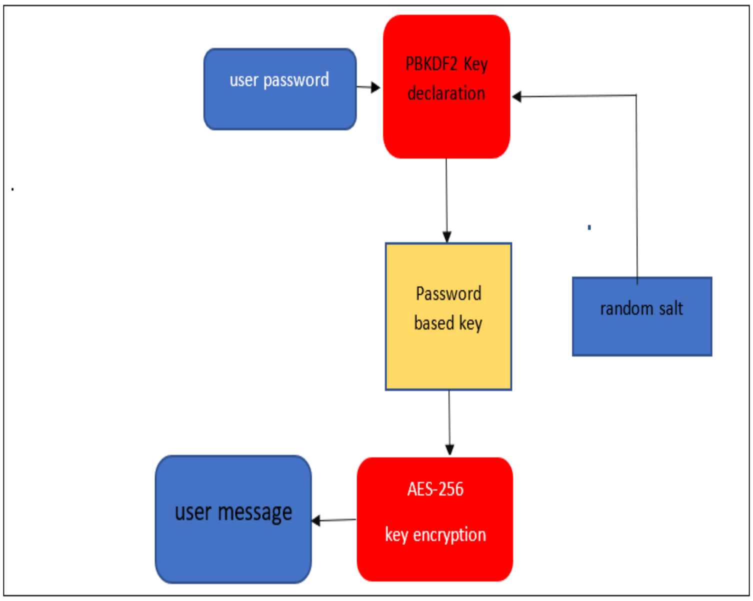


Fig 2: Flow Diagram of PBKDF2 Hashing Technique

C. Usage of Double AES

AES is the advanced encryption standard it's used to encrypt more securely it uses a key to encrypt the given data. Single level of AES encryption is difficult to crack with AES 256. Here we are adding another layer of security by separating what user can achieve. User can decrypt one level by retrieving ciphertext and salt from the encoded image [10]. But the user can't decrypt the text just by that as we have another level of encryption. The rest of the decryption can be done only if the user knows the structure of how the encryption is done where the second key is stored. So, it increases the security.

D. Usage of Two Passwords

We are separating both odd and even characters and we are giving different passwords so we have two different keys for encryption and we have another key to add another encryption layer to the cipher text we got from both odd and even characters. In the study [14], it is discussed how to use the double Advanced Encryption Standard (AES) to convert alternating plaintext into a cipher text and how to use steganography to embed the cipher text and plaintext into a picture by removing the ODD and EVEN characters for more secure hashing. On top of AES 256 we are using double AES and in one AES we are using 2 passwords one level of encryption to separate both odd and even characters and adding them in image in separate places in an image. Odd part of data is stored in second half of the image in reverse order with it's key. Even part of the image is stored in first half of the image with it's key. So, it's difficult to distinguish between key and cipher text. And we have a separate key hidden in image using LSB for second layer of encryption. In this encryption password technique, as given in Fig. 3, the password for odd and even characters are received from the user and by using a random salt 2 32-bit keys are generated and a random 32-bit key is generated to encrypt on top of this which is system generated. Where odd characters are stored in the second half in reverse order with a special delimiter added to the key of odd characters for example [end]. The data will be like [end]+ key+ ciphertext. And this will be stored in reverse order. We will use the delimiter to retrieve both key and encrypted text as we are using a 32-bit key, we don't need another delimiter. Similarly, for even characters the data is stored as [ciphertext+key2+[end]]. The cipher text in both odd and even are encrypted using double AES. Following that the data is fetched from both beginning and end simultaneously by checking the delimiter if found the data exists else there is no hidden data in the image. After fetching the data which is a combination of both key and ciphertext for odd and even. Use the key from the image to decrypt the first layer by using the salt which is also hidden in the image itself. As the image contains most of the secure data in encrypted manner it's more secure. We can also use database where we can store the key and salt for authentication for a particular user and only certain users can access certain data. Even if they know the password, they can't understand anything as the salt is different for every access. Finally, both odd and even characters are merged together to give the end user a desired output from the output image. Next, the user can enter both original and embedded image path to get both PSNR and mean square values and these are entered in a csv file. It contains input image output image data hidden length of hidden data MSE and PSNR values. The flowchart process is depicted in Fig. 3.

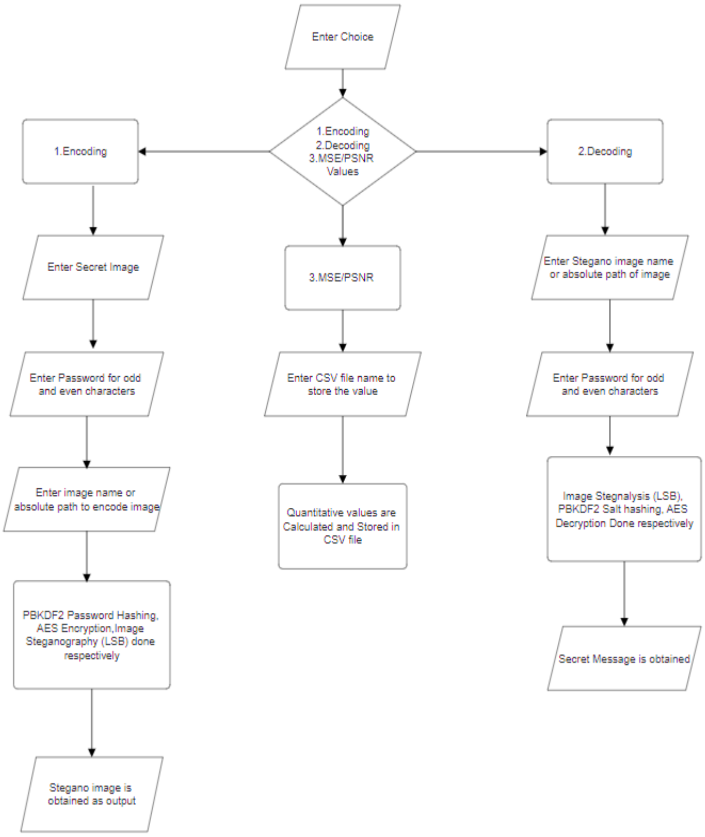

Enter Choice
1.Encoding
2.Decoding
3.MSE/PSNR Values
1.Encoding
2.Decoding
Enter Secret Image
3.MSE/PSNR
Enter Stegano image name or absolute path of image
Enter Password for odd and even characters
Enter CSV file name to store the value
Enter Password for odd and even characters
Enter image name or absolute path to encode image
Quantitative values are Calculated and Stored in CSV file
Image Stegnalysis (LSB), PBKDF2 Salt hashing, AES Decryption Done respectively
PBKDF2 Password Hashing, AES Encryption,Image Steganography (LSB) done respectively
Secret Message is obtained
Stegano image is obtained as output

Fig 3: Flow Chart of Proposed Embedding and Decryption Process

IV. RESULT AND ANALYSIS

A. Subjective Evaluation

For the Experiment, Colored PNG pictures were utilized as the cover image for the suggested algorithm's trials. The secret data is successfully encrypted using the Advanced Encryption Standard (AES) method once the original cover picture was examined, guaranteeing a high level of security. PNG pictures of various sizes were tested to see how authentic they were. The information in Table 3 indicates a lack of noticeable distinctions between the two images, posing a challenge for the human eye to discern any variations or changes when comparing the original image with the steganographic image.

Original Image			
Embedded Image			

Table 3 : Comparison of Cover Image and Stegno Image

In Table 4, it is illustrated that encoding various characters into the image does not result in any changes in the image dimensions. Despite the size of the embedded data needing to be smaller than the image size, they maintain the exact dimensions. This suggests that the image size experiences only a marginal increase, even when the size of the embedded data is larger.

Image Name	Original Image Size	Original Image Dimension	No of characters embedded	Stegano Image Size	Stegano Image Dimensions
Dice	221 KB (2,26,933 bytes)	800 x 600 pixels	50	238 KB (2,44,104 bytes)	800 x 600 pixels
Bird	267 KB (2,74,375 bytes)	840 x 859 pixels	10	275 KB (2,81,786 bytes)	840 x 859 pixels
Art	257 KB (2,63,267 bytes)	800 x 600 pixels	100	286 KB (2,64,109 bytes	800 x 600 pixels

Table 4: Result after performing the experiment

B. Objective Evaluation

Performance Metrics like Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error (MSE) are used to evaluate the fidelity and quality of stegano image with RGB pixel and frequency scale. The MSE and PSNR values are defined by the formula in equations (2) and (3).

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{i=1}^{N}(p_{ij} - q_{ij})^{2}, \qquad (2)$$

M = Height of Cover Image, N = Width of Cover Image
pij = Pixel Value before embedding characters,
qij = Pixel Value after Embedding Characters

$$PSNR = 10 \times \log_{10}\frac{Cmax^{2}}{MSE} \qquad (3)$$

In our case, c_{max} will be 255. The stegno-image quality is rather acceptable if the PSNR value is in the range of 30 to 40 decibels. A stegno-image considered to be very good has a PSNR value more than 40 dB, and any alterations are barely perceptible [10]. The quality of the steganography increases with the PSNR value.Fig.

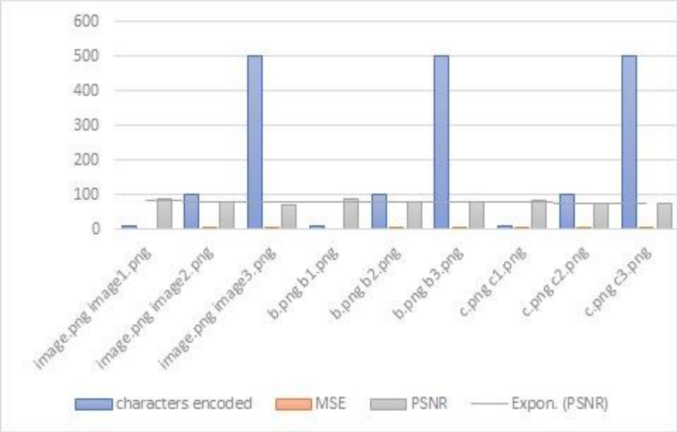


Fig 4: Exponential PSNR Values for 3 Sample Images

4. represents the experiment's findings based on MSE and PSNR values.

C. Histogram analysis

The PSNR values show the RGB scale for the original cover picture when they are plotted against the image's pixel values and expressed in decibels. The results from Fig. 5 indicate that there is very little difference between the two image's histograms [15]. This alignment is attributed to the Adaptive LSB technique, which minimizes the visibility of data to the naked eye by altering a limited number of pixels. These metrics serve to evaluate both the efficacy of the proposed embedding algorithm and the quality of the steganographic image.Visually, the steganographic image closely resembles the cover image, affirming the effectiveness of the suggested technique for image steganography across diverse image sizes and formats. The process involves encrypting and embedding the data in the image, and upon retrieval using a

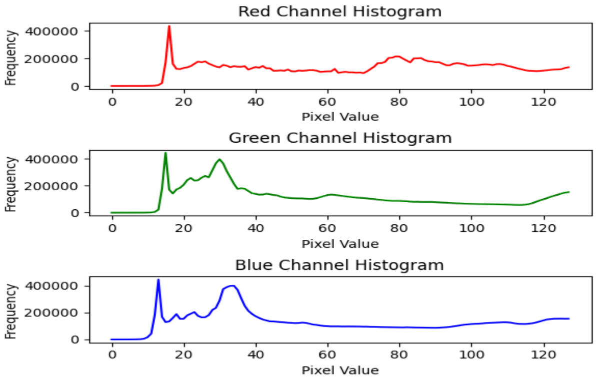
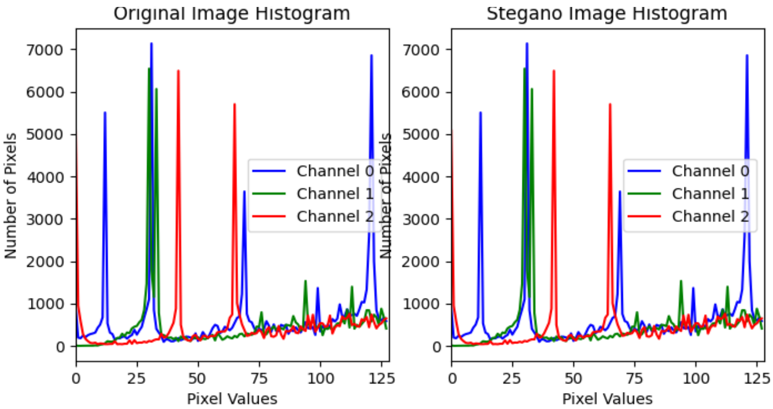


Fig 5: PSNR Value for Sample RGB Image

Fig 6: Histogram Analysis with different RBG Channel



hashed key via PBKDF2, the original data is reproduced Fig. 6 reprsents the Pixel Variation for orginal and stegano image in 3 different rgb channels.

V. CONCLUSION

The research explores a collaborative integration of AES 256 encryption and LSB steganography, unlocking a new area for high-security and secret data transmission. By imposing the strong encryption of AES 256 and the concealed embedding capabilities of LSB, we have demonstrated a powerful approach for Hexadecimal Key generation and Password Hashing by Separating the odd and even characters that overcomes the limitations of previous techniques. By utilizing the combined power of AES 256 encryption and LSB steganography armored with PBKDF2 Password Hasher. The results show a camouflaged stegno image which is exactly the same as the original cover image, with only a significant Peak Signal Noise Ratio (PSNR) distortion making it non-viable to attackers. As a result, this system exhibits a strong security architecture for classifying and distributing data.

REFERENCES

[1] V. Singal, Y. K. Shukla and N. Prakash, “Image Steganography embedded with advanced encryption standard (AES) with SHA-256”, (IJITEE) International Journal of Innovative Technology and Exploring Engineering, Vol. 9, No. 10, 2020

[2] T. Bhuiyan, A. H. Sarower, R. Karim and M. Hassan, “An image steganography algorithm using LSB replacement through XOR substitution”, 2019 International Conference on Information and Communications Technology (ICICT), pp. 44-49, 2019

[3] Mwaffaq Abu-Alhaija “Crypto-Steganographic LSB-based System for AES-Encrypted Data”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 10, 2019.

[4] B. Lin, J. He, J. Huang and Y.Q. Shi, “A survey on image steganography and steganalysis”, Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 2, pp. 142-172, April 2011.

[5] Y.Jinachu, K.M Singh and T.Tuithung, “Image Steganography and Steganalysis: A Survey”, International Journal of Computer applications, vol 2, no 52, 2012

[6] Visconti, M. Brosz, O Mosnacek and V Matyas, “A case study of LUKS examining PBKDF2 security Margin”, Journal of Information Security and Applications, vol 46, pg 296-306, 2019

[7] M. Rahul, M. Malathi, N. Satish Kumar, R. Thamaraiselvan, “Enhanced Image Steganography Using AES & SPIHT Compression”, International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS), March 2017, DOI: 10.1109/ICIIECS.2017.8276029

[8] J. GNDU RC, “Dual-layer security of data using LSB Image steganography method and AES encryption algorithm”, International Journal of Signal Processing Image Processing and Pattern Recognition, vol. 8, no. 5, pp. 259-266, 2015.

[9] B. Karthikeyan, R. Phani Teja, G. V. Gowtham, “Conglomerate Encipher”, International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 6, 2019

[10] Oad, H. Yadav, A. Jain et al., “A review: image encryption techniques and its terminologies”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN, pp. 2249-8958, 2014.

[11] Hussain, I. et al. “A survey on deep convolutional neural networks for image steganography and steganalysis”, KSII Transactions on Internet and Information Systems, 14(3), pp. 1228–1248. 2020.

[12] Atee, H. A., Ahmad, R. and Noor, N. M. “Combining Cryptography and Steganography for Data Hiding in Images”, conference of Applied Computer and Applied Computational Science (ACACOS), 5(12), pp. 128–134, 2014

[13] Pritesh, P. “A Cryptography Application using Salt Hash technique”, International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 2, no. 6, 2013

[14] B. Karthikeyan, K. S. Krishna, D. Chandrasekaran, R. Seethalakshmi, “A Combination of Plain Text and Cipher Text Based Steganography Through Advanced Encryption Standard”, International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 1, 2019

[15] R. Jaiswal, A. G. Rao, and H. P. Shukla, “Image Enhancement Techniques Based on Histogram Equalization”, Int. J. Adv. Electron. Eng., vol. 1, no. 2, pp. 69–78, 2010