Image Steganography embedded with Advance Encryption Standard (AES) securing with SHA-256

Vikas Singhal, Yash Kumar Shukla, Navin Prakash

Abstract- The proposed paper, works upon the idea of securing the classified information. This is achieved by using steganography which is an approach to hide classified information into some other file while maintaining its visual aids and secondly is cryptography which works upon textual data and transform it in a way that no one can comprehend it. The proposed method secures the weaker section which is the key in Advance Encryption Standard using hashing technique. The proposed work enhances the level of concealment of information from unauthorized access and for covert information exchange by encrypting the data and hiding it into a multimedia file known as image. The Secure Hash Algorithm 256 generates a hash key of 256 bits which is an unbreakable hashing technique after that the key is used in the process of encrypting the text with Advance Encryption Standard 256 which is an unbreakable encryption technique till this time and a cipher text is obtained. The cipher text is embedded into a target image using Least Significant Bit method which make changes in image that cannot be understand by naked eyes. The change in byte is 0.000002%. It ensures the visual quality of an image remains intact. The distortion or change in the image remains intermittent to human eyes. The major issue concerned for the government and security agencies such as were to exchange highly classified information in a secure and undetectable manner and abide the notion of hacker to comprehend any such information.

Keywords- AES, Cryptography, Image steganography, LSB, SHA-256.

I. INTRODUCTION

Data plays a vital role in merely each vertical or domains. Data provides the next level of insight into different verticals. The data has a versatile nature from personal to economic, governmental, national security, technology and so on. As we all know the data is the new oil so when storing, exchanging, and using data is done then arrives a high responsibility to maintain its confidentiality, integrity, availability of the data. The higher the confidential information, higher the security is needed. From several years data hiding has captured the minds of security agencies and researchers. Digital watermarking, filtering & masking and other steganographic techniques are used to hide confidential data into digital images. The cryptography converts the plain text into an incomprehensible cipher text whereas steganography conceals the traces of the data that could be ever existed.

II. LITERATURE REVIEW

The evolution in technology changes the life of mankind. As the technology cases mankind life, it also brings a need for additional security. Steganography is a mechanism of hiding confidential and sensitive information in images to have secure and untraceable communication between multiple individuals. Steganography ensures the exchange of data covertly using any digital communication medium, even the toughest condition of constant monitoring. There is a possibility of doing covert communication and exchange of data without being get noticed and no one can imagine the existence of a secret message inside an image. The embedding is done by weakening some of the traits of other media such as image, audio, video files which are also known as cover. The Final resultant has the same properties of cover media. The cover contains the crucial secret data in it. When a secret data is concealed inside a cover image, then it results in a stego image. This paper depicts the concept of steganography and proposed methodologies of steganographic images.
SHAs-256 was developed by United States National Security Agency and then got recognized as well as published in the year 2001. SHA-256 is a cryptographic hash also known as the digest. It generates a unique 256-bit (32 bytes) code of a text. A hash is not an encryption technique so, it cannot be decrypted back to original data. It creates a fixed size of code from the source text. It is a single way cryptographic function. The hash version of the text is always gets compared only, which enhances security.

It avoids transmitting password in original view format, even if the message gets intercepted between the medium or channel then also interceptor can never reach to the original password. The NIST (National institute of standard and technology) approves that it has never been compromised [2]. The Advance Encryption Standard (AES) is a cryptographic algorithm which is used to encrypt the textual data in an incomprehensible manner. And it is a most secured encryption standard till now where no signs of its breakage been reported till yet. It is a symmetric key algorithm, which means the same key is used for both encryption and decryption both. It uses a block cipher of sizes 128, 192, 224 and 256 bits. It uses several ways in AES like XOR operation, substitution, permutation operation, rows and column shifting [3].

III. PROPOSED METHOD

The Proposed methodology is to create a system which could be used for covert communication without any hint that some classified data is being exchanged via using image steganography and securing that data with cryptographic technique and even securing its vulnerable area that is key by using hashing.

A. Least Significant Bit (LSB)

The steganography technique embeds the data into the cover image or target image. This can be achieved by using various steganographic techniques. From a pool of steganographic techniques I will be accomplishing this task with the usage of LSB. This technique is implemented by making changes in the least significant bit (LSB) of a byte. LSB is a most promising steganographic technique which conceals the data in the image and no one can get a whisper that the image accommodates some crucial data within it. LSB provides the least change in the image which is far from persistent of a human naked eye. The change in a byte is 0.000002% which is almost negligible.

The least significant bit of the cover image byes is changed by a bit of the classified data. LSB functions finest with 24-bit map (BMP) and PNG images in which each pixel is constituted of 3 bytes (red, green, blue) color respectively. LSB provides a method to store 3 bits of information in a single pixel, by making a change in one bit of each color component. Therefore, by using a 256*256 image, LSB can store 196608 bits (i.e. 24576 bytes) of data inside this cover image.

| AUTHORS | APPROACHES/ALGORITHMIC TECHNIQUES | OUTCOME |
|---------|-----------------------------------|---------|
| United States National Security Agency (NSA), 2001 | Secure Hash Standard | Hash value is obtained containing 256-bit hash value. |
| Charles O. Bravely, Jr., Newark, DE (US); Lisa M. Marvel, Churchville, MD (US); Charles T. Tenero, Belcamp, MD (US), 2003 | Spread spectrum and image steganography | It generates a spreading Sequence with a pseudorandom noise generator by a key, modulates the encoded text by the Spreading Sequence to obtain an embedded Signal and merging the embedded signal with a cover Signal to generate a StegaSignal. |
| Po-Yueh Chen* and Hung-Ju Lin, 2006 | A DWT based approach for image steganography | Data is embedded in the image with a secure key matrix. |
| Donnecio Bristo and Luca Iacobi, 2007 | Image based steganography and cryptography | A unified approach using ISIC as in cryptography and steganography. |
| Ali Abo-Andi and Fawzi Al-Nuami, 2010 | A modified high capacity image steganography technique based on wavelet transform | Data is hidden in the image with wavelet transformation in message as well as image. |
| Shailesh Gupta, Ankit Goyal, Bharat Bhasin, 2012 | Information hiding using least significant bit steganography and cryptography | RSA and Diffie Hellman algorithm has been used as cryptography technique to generate cipher text and LSB is used to embed it into image. |
| Sathif Islam*, Manger R Modi and Photogri Gupta, 2014 | Edge-based image steganography | Hiding of data into the image edges is achieved. |
| *Khan Muhammad, Jamiyl Ahmed, Halseen Farnan, Muhammad Zubair, 2015 | A novel image steganographic approach for hiding text in color images using HSI color model | It contains larger Peak Signal to Noise Ratio (PSNR) value. |
| Khan Muhammad, Jamiyl Ahmed, Muhammad Sajjad, Muhammad Zubair, 2015 | Secure image steganography using cryptography and image transposition | Multiple encryption (hash operation, bits shuffling, and stego key-based encryption) and Image transposition is done for data hiding. |
| Muhammad Abu-Albahah, 2019 | Crypto-Steganographic LSB-based System for AES-Encrypted Data | Stega image is obtained with cipher text by AES-128-bit encryption. |

Based on table 1 and after doing a detailed study, the proposed system technique has been chosen in order to have a robust system which in all aspect can shield the classified information and data within it.
The table 2 contains the pixel grid of a image where each pixel flourishes some colors. Whereas in table 3 it can be clearly seen that the pixel grid of the cover image has been changed in order to embed data into it. And the change in each pixel is so nominal that it cannot be traced. LSB makes use of lossless compression provided by PNG images. consecutive bytes of the image data are used to embed the information, initially from the first image pixel of each byte of the message in respective order. Consequently, LSB is the best way to hide data and communicate covertly but if any hacker thinks that the image contains confidential data, and he will try to extract the crucial data from it by use of steganalysis tools and if he is accomplishing his goal then our crucial data is in wrong hands. It orders to achieve a higher level of security this proposed system is created which add on advance level of security. This add on intricate multiple layers of imperishable security. The image constitutes of pixels that do not engage changes in the anatomy of the message in order to accomplish secure and imperceptible communication. A steganographic technique could embed an encrypted data in the cover image. The proposed method acquires LSB technique in order to replace the least significant bits in the cover image which contains bits of the hidden message. In succession to enhance the security level of the proposed system, the covert data should be encrypted using AES encryption using a key which is generated using the SHA-256 hashing technique. The proposed system ensure the difference between an original image and a cover image which remains imperceptible to human senses, meanwhile sustaining the integrity of the confidential data without any loss.

B. Secure Hash Algorithm (SHA-256)

The Secure Hash Algorithm (SHA-256) is a part of family. The family of cryptographic hash functions which was published by the National Institute of Standards and Technology as a U.S. Federal Information Processing Standard. SHA-256 is patented in US patents 6829355. Hash is not ‘encryption’ because it can never be decrypted back to the initial text. It is of fixed length for any varying length of the source text. SHA-256 is achieved by the following steps like padding, append length, divide the input into 512bit blocks, add chaining variable, process block respectively. SHA-256 generates a unique 256-bit or we can say 32-byte of hash code. It does 64 rounds in order to reach to the final hash value. The text length up to $2^{64}$ bit i.e (2.3 exabyte) will also be transformed into a digest size of 256 bits (32 bytes). 16.80 cycles per byte on 64-bit processor x86 architecture. It has 64 iterations in one cycle. Which makes it more safe and secure.

For example: if we want to embed “A” in a image. ASCII code of “A” is 65. The binary value of “A” is 01000001.

| RED | BLUE | GREEN |
|-----|------|-------|
| 1000000 | 10100100 | 11101011 |
| 10110100 | 11100111 | 10101111 |
| 11001110 | 10101001 | 00100011 |

For example: Hash value of “abc” is “ba7816b9f01cfe414140de5ae2223b06361a396177a9cb410ff061f0015ad”. It is of 256 bits (32 bytes). The hash value is obtained by the respective hashing operations mentioned above which are performed in SHA-256.

C. Advance Encryption Standard (AES-256)

AES is a symmetric key encryption technique, which use only one secret key to cipher(encrypt) and decipher(decrypt) data. Advanced Encryption Standard (AES) is only publicly accessible encryption technique authorized by the US National Security Agency (NSA) for securing top classified secret data. AES was earlier known as Rijndael after its two amazing developers known, Belgian cryptographers Vincent Rijmen and Joan Daemen. AES-256, uses a key length of 256 bits, it uses the largest bit length and is practically adamantine by brute force attack on the basis of current computing power, which makes it the strongest encryption standard. The AES encryption method first performs substitution of data using a substitution table; which is followed by shifting of data rows, then after that mixing of the columns takes place, and finally the last transformation with a simple exclusive (XOR) operation performed on each column using a different part of the encryption key. The following table shows that possible key combinations exponentially increase with the key length. And it also makes it hard to crack it.

| Key Length | Possible Combinations |
|------------|-----------------------|
| 1 bit      | 2                     |
| 2 bits     | 4                     |
| 4 bits     | 16                    |
| 8 bits     | 256                   |
| 16 bits    | 65536                 |
| 32 bits    | $4.2 \times 10^3$     |
| 64 bits    | $7.2 \times 10^4$     |
| 128 bits   | $1.8 \times 10^5$     |
| 256 bits   | $1.2 \times 10^6$     |
PHASES OF THE PROPOSED METHOD

The first phase involves key generation which is done by using SHA-256 a hashing technique. Hash is not ‘encryption’ because it can never be decrypted back to the initial text. It is of fixed length for any varying length of the source text. SHA-256 is achieved by the following steps like padding, append length, divide the input into 512bit blocks, add chaining variable, process block respectively. SHA-256 generates a unique 256-bit or we can say 32-byte of hash code. It does 64 rounds in order to reach its final hash value. SHA-256 is a hashing technique which secure weaker encryption as compared to SHA-256 encryption.

The AES encryption method first performs substitution of data using a substitution table; which is followed by shifting of data rows, then after that mixing of the columns takes place, and finally the last transformation with a simple exclusive (XOR) operation performed on each column using a different part of the encryption key. As AES is a symmetric data encryption technique or method, the same key is being used for both the operations encryption as well as decryption. As a result, cipher text is obtained which is incomprehensible. There has not been any report of breakage of AES encryption till now. By which it adds on a one more secure layer over the system.

Figure 2: The Diagrammatic representation of AES-256

IV. PHASES OF THE PROPOSED METHOD

The length of the key is of 256 bit (32 byte) so, 14 rounds of AES encryption have been performed and as a result an incomprehensible cipher text is generated. Same key is used in decryption process also and as a result the text is obtained by running 14 rounds of decryption. All the process executes in a reverse order in a way to get the plain text.

A. Algorithm For Embedding Data Into Image And Creating Stego-Image.

Step 1 Enter the text (T) of any length.
Step 2 Enter the key (k) of any length.
Step 3 The user need to enter a password to protects its data via encryption and decryption is done using it. Enter the Cover image (I) with extension and path.
Step 4 Enter the path of image file with its name which will be hiding the classification information in it.
Step 5 The key (k) undergoes SHA-256 hashing technique and a hash value (H) is obtained.
Step 6 Hash value (H) is used as a key in AES-256 encryption and plain text (T) is converted into cipher text (CT) by using AES-256 encryption algorithm.
Step 7 Cipher text is embedded into the image (I) using LSB a steganographic technique and a stego-image (ST) is obtained.
// Stego-image can be shared now using any communication channel.
Step 8 End
B. Algorithm For Extracting Data From Stego-Image

Step 1 Start
Step 2 Enter the Stego image (ST) with extension and path.
//Enter the path of image file with its name which is having classified information in it.

Step 3 Steganalysis on the Stego image (ST) is performed and cipher text (CT) is obtained.
//Steganalysis is a process to recover data hidden data from the image.
Step 4 Enter the key (K) that undergoes SHA-256 hashing techniques and a hash value (H) is obtained.
//A secure key is obtained by which data can be only retrieved.
Step 5 Hash value (H) is used as a key in AES-256 decryption and cipher text (CT) is converted into plain text (T) by using AES-256 decryption algorithm.
//Original classified information is obtained.
Step 6 End

Fig 3. Flowchart of proposed system (Interface Diagram)
V. EXPERIMENTAL RESULT

The proposed method has been implemented in order invent a secure and impermeable system. And to do an analysis of the result in order to find out whether the proposed system is working accordingly as per design in consideration of requirements or not.

Table 5: Original and Stegno images

| Image Name | Original Image Size | Original Image Dimension | Size of the Data Embedded | Stegno Image Size | Stegno Image Dimension |
|------------|---------------------|--------------------------|---------------------------|-------------------|------------------------|
| Butterfly  | 176 KB              | 385 * 593                | 350 B                     | 176 KB            | 385 * 593              |
| Penguin    | 221 KB              | 395 * 599                | 4 KB                      | 497 KB            | 395 * 599              |
| Snake      | 221 KB              | 219 * 350                | 2 KB                      | 22 KB             | 219 * 350              |

The above table 5 depicts that no change can be seen or persistent to human naked eyes and we cannot find out the difference between both the images i.e. original and stego image. So, by this confidentiality and covertness of the crucial data remains stagnant. Now, if we move forward to table 6 it depicts that no change in the dimension of the image is seen no matter how much amount of data been input into the image. Apparently, the data should always be less than the capacity of the image to hold it. The major difference is that if the data i.e. embedded in the original image comprises less than 1 KB then we cannot see the difference in the size of the image. Otherwise, the size of the image gets slightly increase as per the size of the data.

I have used python (Programming language) in order to implement the algorithm and to create a data securing system. And the libraries which I have used are:

1. *pycryptodome* in order to perform AES (Advance Encryption Standard) encryption and decryption.
2. *Stegano* in order to implement image steganography via LSB (Least Significant Bit).
3. *Hashlib* in order to perform hashing via SHA-256 (Secure Hashing Algorithm).

The screenshot of the Proposed working system
The above screenshot depicts that the working of the proposed system is efficient and accomplish the task and functionally robust. If the key is wrong the data cannot be extracted from it, even if the 1st layer of protection is been breached by any technique and the hacker identifies the image contains data. Then also the data is armored with other layer of protection and it cannot extract data from it as it is still being protected from cryptographic technique. The proposed system is robust and if a hacker tries to extract the data from the image, it is shielded with hashing, encryption, and image steganography. In which each technique is impermeable and combination of all the three make it extremely secure. Hence, the proposed method manifest to provide robustly and secure image steganography as well as a secure and inoculative system against data hacking.

V. CONCLUSION

The proposed system accomplishes the task of securing the classified information. Primarily no one gets a hint that some crucial data is being exchanged, and secondly it also gets shielded by using cryptography. The proposed system successfully hides the data using image steganography via LSB and surplus the protection of data using cryptographic technique AES-256 as well as the weak point which a hacker could target that is key, it has also been armored via the use of hashing technique SHA 256. And an impenetrable system is created. The experimentation performed assures the concreteness of the proposed system. The results acquire a camouflaged image, a non-noticeable distortion making it almost non-viable to attract the consciousness of attackers and even after getting it tranced without an authentic key no one was able to access that information. Hence, proposed system proves to be a highly secure system in the terms of securing classified information and exchanging it and it also seals up with no vulnerability in it.

VI. CODE

1) KeyCreation and Hide Method

```
def key_creation(key):
    block_size = 32
    # Getting a hash value of 256 bit (3 bytes).
    key = hashlib.sha512(key.encode()).digest()
    return key

def hide(input_filename, output_filename, data, key):
    block_size = 32
    # Get a random initialization vector
    iv = Random.new().read(AES.block_size)
    # Use cipher block chaining (CBC) mode.
    encryption_suite = AES.new(key, AES.MODE_CBC, iv)
    # If it is string convert to byte string before use it.
    if isinstance(data, str):
        data = data.encode()
    # Encrypt the random initialization vector added with the passed data
    cipher_data = encryption_suite.encrypt(iv + pad(data, block_size))
    # Convert the cipher byte string to a base64 string to avoid decode padding error
    cipher_data = base64.b64encode(cipher_data).decode()
    print("Cipher Text: ", cipher_data)
    # Hide the encrypted data in the image via LSB technique.
    secret = lab.hide(input_filename, cipher_data)
    secret.output(output_filename)
```

2) Retrieve Method

```
def retrieve(input_image_file, key):
    block_size = 32
    cipher_data = lab.reveal(input_image_file)
    if not cipher_data:
        return None
    cipher_data = base64.b64decode(cipher_data)
    # Retrieve the dynamic initialization vector saved
    iv = cipher_data[:AES.block_size]
    # Retrieve the cipher data
    cipher_data = cipher_data[AES.block_size:]
    try:
        decryption_suite = AES.new(key, AES.MODE_CBC, iv)
        decrypted_data = unpad(decryption_suite.decrypt(cipher_data), block_size)
    except UnicodeDecodeError:
        # Binary data - returns as it is
        return decrypted_data
    except ValueError:
        return None
```

VII. RESULT ANALYSIS

By the combination of all three methods we created a robust system. The data gets encrypted by most secure cryptographic technique and then this data gets hided inside an image which leaves no mark of any classified information communication taking place. As the visual aids remains intact. Its results can be seen in table 5 & table 6. And the weak point that a hacker targets is also been procured which is a key by hashing technique and the utilization of that key is done in encrypting and decrypting the data. Which adds on the layer of protection to the classified data. So, our goal gets achieved firstly no sign of secret communication and for safety adding more security layers on it. The data as encrypted and embedded in the image at the time of retrieval it gives the same data while using authentic secure key.
REFERENCES

1. Provos, Niels, and Peter Honeyman. "Hide and seek: An introduction to steganography." IEEE security & privacy 99.3 (2003): 32-44.
2. Henri Gilbert and Helena Handschuh, “Security Analysis of SHA-256 and Sisters**”, 2003.
3. Selent, Douglas. "Advanced encryption standard." Rivier Academic Journal 6.2, ISSN (Online): 2319-7064 Index Copernicus Value (2015): 78.96 (2010): 1-14.
4. United States National Security Agency (NSA). U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory (ITL), “Secure Hash Signature Standard (SHS) (FIPS PUB 180-2)”, National Institute of Standards and Technology (NIST), 2001 : 9-22.
5. Charles G. Bonecelet, Jr., Newark, DE (US); Lisa M. Marvel, Churchville, MD (US); Charles T. Retter, Belcamp, MD (US). “Spread spectrum and image steganography”, 2003.
6. Po-Yueh Chen* and Hung-Ju Lin, “A DWT based approach for image steganography”, DOI:10.6703/IJASE.2006.4(3).275, 2006.
7. Domenico Bloisi and Luca Iocchi, “Image based steganography and cryptography”, 2007.
8. Ali Al-Ataby and Fawzi Al-Naima, “A modified high capacity image steganography technique based on wavelet transform”, Vol. 7, No. 4, October 2010.
9. Shailender Gupta, Ankur Goyal, Bharat Bhushan, “Information hiding using least significant bit steganography and cryptography”, DOI: 10.5815/ijmecs.2012.06.04, 2012.
10. Saiful Islam*, Mangat R Modi and Phalguni Gupta “Edge-based image steganography”, DOI: 10.1186/1687-417X-2014-8, 2014.
11. *Khan Muhammad, Jamil Ahmad, Haleem Farman, Muhammad Zubair, “A novel image steganographic approach for hiding text in color images using HSI color model”, DOI: 10.5829/idosi.meir.2014.22.05.21946, 2015.
12. Khan Muhammad, Jamil Ahmad, Muhammad Sajjad, Muhammad Zubair, “Secure image steganography using cryptography and image transposition”, October 2015.
13. 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.
14. 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.

Authors Profile

Mr. Vikas Singhal, Head of the Department of Computer Applications, JSS Academy of Technical Education, Noida. He received his Master of Computer Applications from M.S. Ramaiah Institute of Technology Bangalore. He completed his M.Tech in Information Technology from USIT, GGSIPU, New Delhi. He is pursuing Ph.D from IFTM University, Moradabad. His research interests are Image processing and Machine Learning.

Mr. Yash Kumar Shukla is a bachelor’s in computer application from Integral University. He is currently pursuing master’s in computer application in JSS Academy of Technical Education Noida, a prestigious institute affiliated to Dr. APJ Abdul Kalam Technical University, Lucknow, UP, as a final year student, he is also associated with an Artificial Intelligence and software based service/product company for his industrial internship and is working as a developer intern.

Dr. Navin Prakash, received his Ph.D from IFTM University Moradabad. His research interests are Image processing and Machine Learning.