Volume 5, Issue 12, December 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Combining Hash Value and Ciphertext Using Blending Algorithm in Steganography

Vimla Yadav
Gargi Institute of Science & Technology
Bhopal (M.P.) India

Pankaj Kumar Sahu
Barkatullah University
Bhopal (M.P.) India

Abstract— The word Steganography is derived from a Greek word, which means secret writing. The modern digital steganography refers to art of hiding messages inside a covered carrier file such as image, audio, video, text, or any other digital file such that the existence of the embedded messages cannot be detected easily. Steganography and cryptography are entirely different as the essence of steganography lies in hiding the secret message where as cryptography refers to scrambling of secret message. This paper presents a technique under which a plaintext message is first encrypted using AES encryption algorithm and SHA-1 algorithm is applied on the plaintext to generate its hash. The main contribution of this research is a blending algorithm that combines encrypted message and the hash value before embedding it into the cover image. We have applied LSB technique for embedding the blended message into image because with LSB, there are minor chances of degradation of original image and large amount of information can be stored in the image. The advantage of blending algorithm is that once it is identified that the image contains some secret message, the attacker cannot separate the hash and ciphertext for applying cryptanalysis.

Keywords—AES, SHA-1, LSB, Steganography, Data hiding

I. INTRODUCTION TO STEGANOGRAPHY

In modern world, almost any business is based upon communication and it is basic requirement to keep the important information to be secret and safe. As we all know that Internet is the biggest medium for communication but it is not considered safe for transferring and sharing of information. In order to keep the vital information secure, it is essential to use either Steganography or Cryptography or a combination of both. Cryptography includes modification of a message in a way which could be in encrypted form generally guarded by an encryption key which is known to sender and receiver only and without using encryption key the message couldn't be accessed. But in cryptography it is always apparent to eavesdropper that the message being transferred is in an encrypted form. This is not the case in steganography because the secret message is made to hide in cover file so that it looks like a normal message to eavesdropper and he does not even know whether there is any message hidden in the information or not, which makes the cryptanalysis further more difficult. The cover file containing the secret message is transferred to the recipient. The process of embedding message in steganography process is shown in Fig.1.

(a) Basic process of embedding secret message into cover image

(b) Basic process of extracting secret message from stego-image

Fig. 1 Basic process involved in steganography

In the past decades, the simple steganography process has evolved as a complex combination of cryptography and steganography. It makes the job of eavesdropper much more difficult as compared to time when plain old steganography techniques were used[8]. In this paper, we present a novel algorithm that combines the ciphertext generated after applying AES on plaintext, and SAH-1 message digest generated from plaintext referred to as blended code. This blended code is finally written into the least significant bits in the pixels of cover image. Since proposed work is based upon AES encryption technique and SHA-1 hash algorithm hence a brief introduction to both of these techniques without getting into their technical details is provided in next section.

This paper is organized as follow: section.2 elaborates the facts about AES encryption standard and SHA-1. Section.2 provides a brief discussion about the work already done around the world in this area. Section.3 explains the architecture and algorithms involved in proposed work. Results have been discussed in section.4 and finally section.5 concludes the paper.

II. AES AND SHA-1

Almost all cryptographic algorithms have some problem. On one hand, earlier ciphers can easily be broken using modern computation systems whereas on other hand, DES algorithm was broken in 1998 using a system that cost about $250,000. It was also found too slow in mid-1970's hardware. Triple DES has three times as many rounds as DES and is TREMENDOUSLY slow. Also 64 bit block size of triple DES and DES was not very efficient and is questionable when it comes to security [4].

As a result, the National Institute of Standards and Technology (NIST) established new Advanced Encryption Standard (AES) specification in 2002. It is a symmetric key based algorithm that allows a variety of block and key sizes unlike DES that supports just 64 and 56 bits. The AES was designed with to resist all known attacks at the same time keeping it simple so that it runs fair enough.

SHA-1 produces a message digest based on principles similar to those used in design of the MD4 and MD5 message-digest algorithms, but has a more conservative design. It is required by law for use in certain U.S. Government applications, including use within other cryptographic algorithms and protocols, for the protection of sensitive unclassified information etc.

III. PREVIOUS WORK

Abbas et al.[1] proposed a block-based image steganalysis system and conducted extensive performance evaluation of block-based image steganalysis studied the performance of the block-based steganalysis by varying different parameters, including block number, the block size, the effects of block overlapping, the class number of block, the classifier choice and the decision fusion scheme. It was practically seen that the performance of block-based image steganalysis is not as much of sensitive to the decision fusion approach but more responsive to classifier choice.

To ensure the security against the steganalysis attack, Vijay Kumar et al. [10] proposed a steganographic algorithm for 8bit (grayscale) or 24 bit (colour image) based on Logical operation. This algorithm embedded MSB of secret image in to LSB of cover image which is claimed to improve the quality of stego-image greatly with low extra computational complexity. Experimental results showed that the stego-image was visually indistinguishable from original cover-image with this technique. Hayfaa Abdulzahra Atee et al.[3] proposed an encryption method which was combined with two steganographic methods separately and claimed their performance and effectiveness to be improved. The two steganographic methods were Simple LSB and color image based data hiding (CIBDH).

In [2],Atallah et al. proposed a method that hides the secret message based on searching about the identical bits between the secret messages and image pixels values. They claimed that this technique works better as compared to LSB. They also claim that the proposed technique is efficient, simple and fast it robust to attack and improve the image quality, which obtains an accuracy ratio of 83%. Shamim Ahmed Laskar et al. [8] proposed a high capacity data embedding approach by using a combination of steganography and cryptography. They provided resistance against visual and statistical attacks as well as high capacity using LSB embedding technique. The encrypted message to be hidden was converted into its equivalent ASCII value and subsequently into binary digit before writing it to least significant bit of pixel. In the process a message is first encrypted using transposition cipher method and then the encrypted message is embedded inside an image using LSB insertion method. The authors claimed that combination of these two methods enhances the security of the data embedded, and this combinational methodology will satisfy the requirements such as capacity, security and robustness for secure data transmission over an open channel.

Sandeep et al.[7] proposed LSB & DCT based Steganography. Authors implemented LSB and DCT based steganography and calculated PSNR ratio. The result shows that PSNR ratio for DCT based steganography scheme is higher than LSB based steganography scheme for different types of images. In [6], Rinu Tresa et al. came up with a technique that combines both steganography and cryptography so that attacker doesn't know about the existence of message and the message itself is encrypted to ensure more security. The textual data entered by the user is encrypted using AES algorithm. After encryption, the encrypted data is stored in the colour image by using a hash based algorithm. This technique does not corrupt images quality in any form. The size of cover image, a large amount of information can be embedded into it. It is because only single bit of every pixel changes due to which, there is minor change in histogram hence stego-image is visually identically same as was the original cover-image.

Pragya Agarwal et al. [5] proposed a scheme under which, a SHA-1 hash code is generated from original text message, which is sent to the receiver through a secure channel. The receiver can authenticate the received hash to ensure the integrity of the original message. The message is sent to receiver by hiding it into an image using image steganography. Moreover the receiver can authenticate the received message to ensure that any intruder has not altered the original message. To facilitate authentication facility, a hash code is generated from original message, which is sent to receiver securely. The hash code is sent to receiver by hiding it into an image using image steganography. The receiver decrypts the cipher text message to obtain plaintext message. After that receiver calculates hash code from received message which is compared with the received hash code. If both the hash codes are equal, it means that the received message is the original message.

However if an attacker knows the AES key then it can beat the scheme proposed in [5] by generating ciphertext as well as hash code of an entirely new plaintext message and replace both of them inside the image. The root cause of this entire problem is that the attacker possesses both ciphertext and the hash code separately. In this paper, we come up with a solution to this problem by blending the hash code and ciphertext before embedding them into cover image which makes it very hard for attacker to separate them and replace. This proposed blending algorithm provides an extra layer of security. On one hand, proposed scheme benefits sender and receiver as they need not to send two different files thus

reducing the total data transmitted to the network. On the other hand, it will be much more difficult for the attacker to break the algorithm as he never knows the blending method used to blend the ciphertext and digest.

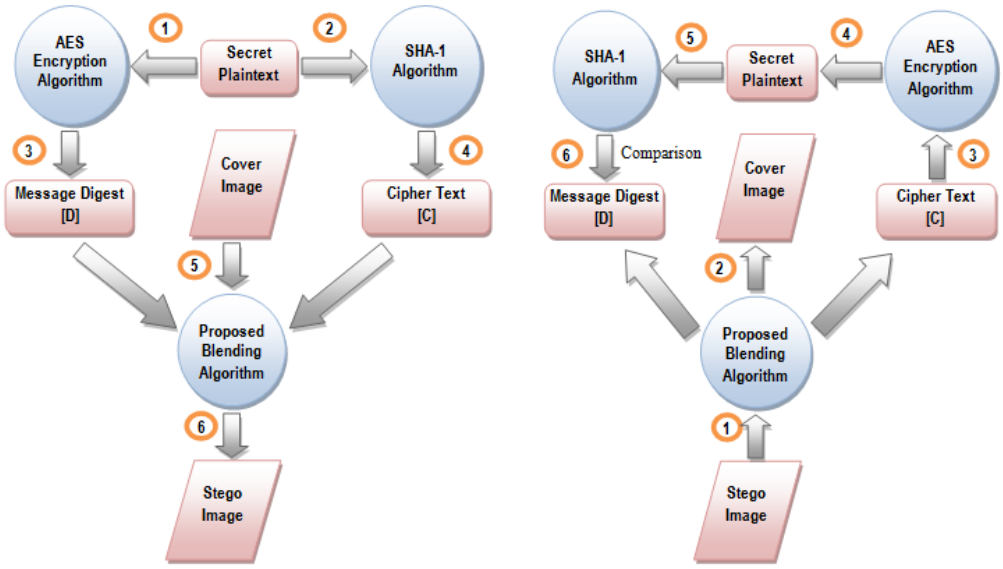
IV. PROPOSED WORK

This section discusses the proposed work in detail. First of all the proposed architecture is presented followed by the detailed depiction of operations that are carried out for combining the ciphertext and hash obtained from plaintext.

A. Architecture

We have divided the architecture of proposed steganography scheme into two parts i.e. sender side and the receiver side as shown in Fig.2(a) and (b) respectively. On sender side, AES encryption and SHA-1 are applied on plaintext (P) to obtain the ciphertext (C) and digest (D). The C, D, and cover image are provided as input to proposed combining algorithm which produces the Blended code at output. This code is finally embedded into the cover image and the stego-image is generated. This stego-image is then transmitted to receiver.

When receiver receives stego-image, it applies reverse combining algorithm on it and yields C and D. The AES decryption algorithm is applied on C to obtain plaintext P. Again SHA-1 is applied on P to obtain D'. Now D and D' are compared, if both are same then authentication of obtained message is successful otherwise the message has been compromised.



(a) Proposed sender side architecture (b) Proposed receiver side architecture

Fig. 2 Proposed architecture

B. Blending Algorithm for Sender

Let the plaintext be denoted by *P*, and *C* be the ciphertext generated from *P* using AES algorithm. Let H be the hash code generated from P after applying SHA-512, and I be the cover image and let C_i and H_i denotes the i^{th} byte of *C* and *H* respectively.

i. Reverse each byte of ciphertext *C* to obtain *C'* as shown in Fig.3.

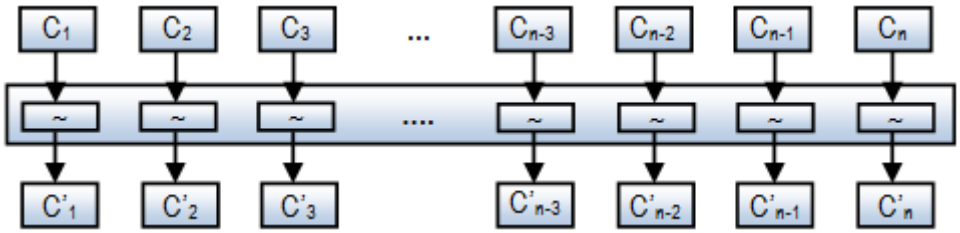


Fig. 3 Reverse the bytes in ciphertext *C*

ii. Reverse each byte of Hash H to obtain H' as shown in Fig.4.

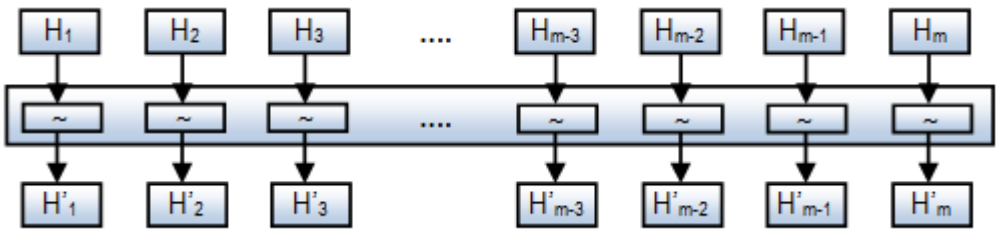


Fig. 4 Reverse every byte of hash H

iii. Write each byte from C' and H' in alternate fashion into temporary array T until all the bytes from C and H are written into it as shown in Fig.5.

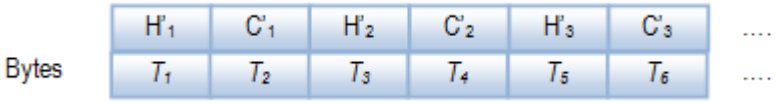


Fig. 5 Write alternate bytes from *C'* and *H'* into *T*

iv. Perform Reverse and 2-Bit Left shift operations respectively on resulting bytes.

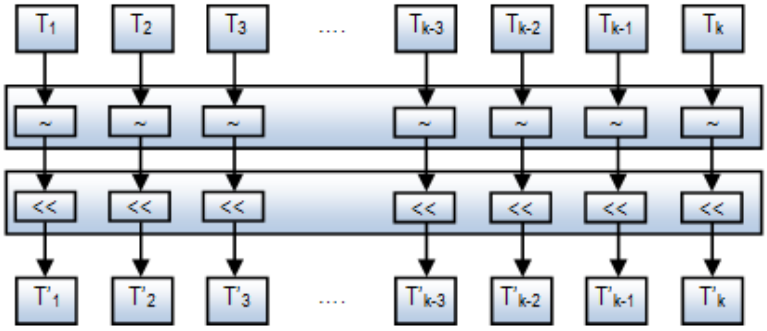


Fig. 6 Produce final mixed array *T'* and write it into image

C. Reverse Blending Algorithm for Receiver Side

i. Read T' from received stego-image. Perform reverse and 2-Bit right shift operations respectively to produce T as shown in Fig.7.

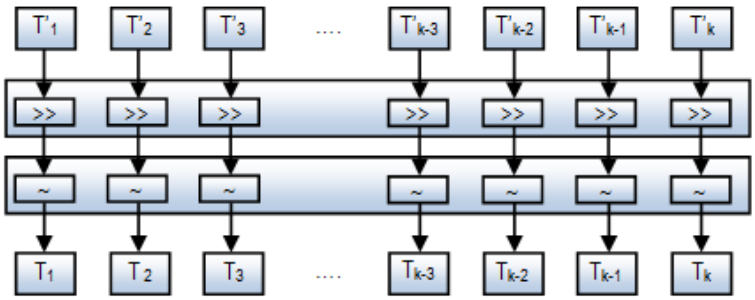


Fig. 7 Produce the temporary array T from received T'

ii. Read alternate bytes of T and obtain the array C' and H' as shown in Fig.8.

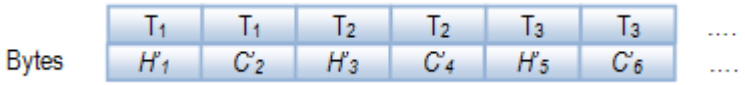


Fig. 8 Obtain C' and H' from T

iii. Reverse each byte of C' to obtain actual ciphertext C as shown in Fig.9.

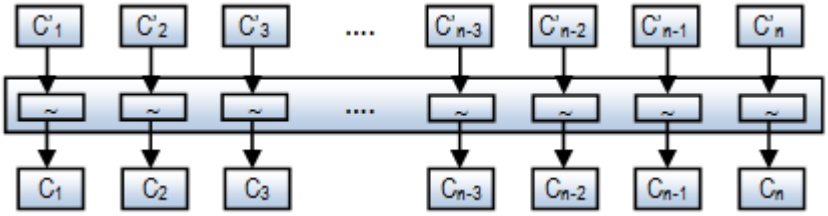


Fig. 9 Reverse every byte of hash C'

iv. Reverse each byte of H' to obtain hash value H as shown in Fig.10.

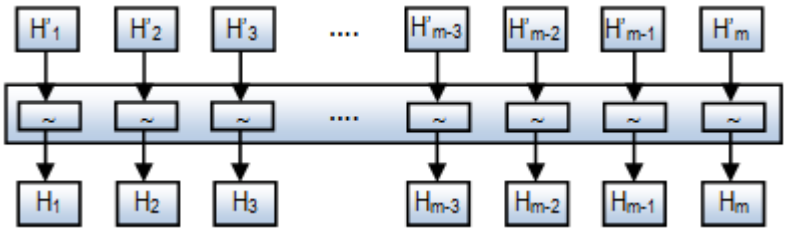


Fig. 10 Reverse each byte of hash H'

v. Apply AES decryption algorithm on C to obtain plaintext P.
vi. Apply SHA-512 on P to obtain new hash value H2.
vii. Compare H and H2, if both are same then message authentication is successful otherwise received image has been compromised.

V. RESULTS AND DISCUSSION

This section presents the results obtained from simulation. We implemented the proposed algorithm using JDK7.0 and compared the results with the algorithm presented in [5].

In the first experiment, plaintext size was set constant (5000 bytes) with varying size of cover images. Results shown in Fig.11 reveal that the size of final output of proposed scheme, which is transmitted over network using proposed algorithm is always smaller as compared to previous scheme. It shows that proposed algorithm is better regardless of cover image size, in terms of data size transmitted over network.

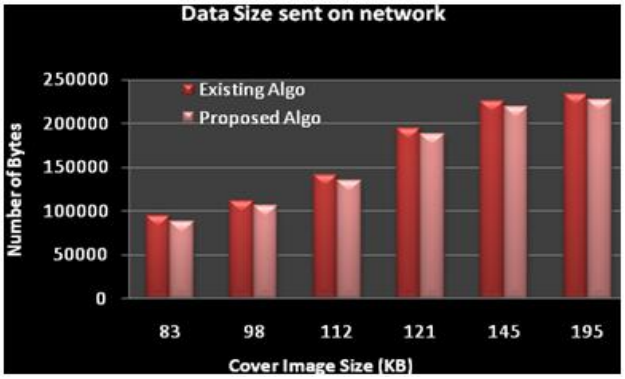


Fig. 11 Data bytes sent over network when plaintext size is constant

In the second experiment, cover images size was set constant (200704 bytes) whereas size of plaintext was varied from 400 to 2500 bytes. Results shown in Fig.12 reveal that the size of data sent over the network using proposed algorithm is consistently smaller as compared to previous scheme. It proves that proposed scheme works better than existing algorithm regardless of plaintext size, in terms of data size, which is transmitted over network.

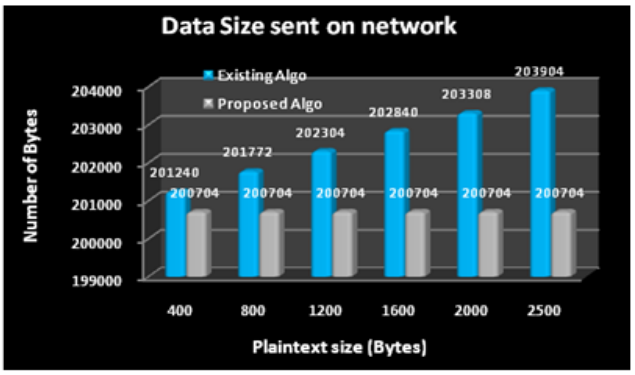


Fig. 12 Data bytes sent over network when cover image size is constant

In third experiment, plaintext size was set constant (2000 bytes) with varying size of cover images. The image utilization can be termed as the percentage of pixels in stego-image that are rewritten (utilized) by the algorithm. Results shown in Fig.13 show that when compared to existing algorithm, the image utilization was consistently significantly high when proposed algorithm is used. It proves that the proposed algorithm works much better in terms of image utilization regardless of cover-image size. It should be noted that the image utilization can further be increased by increasing the size of plaintext.

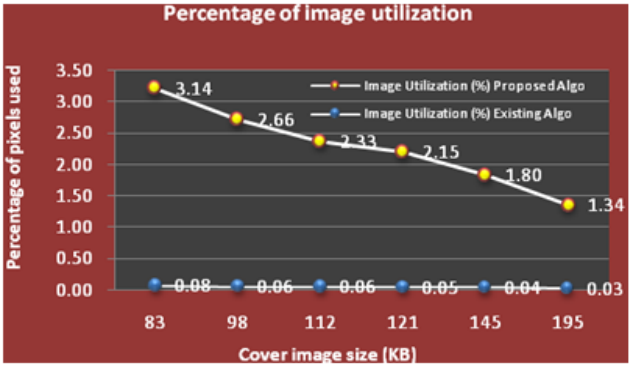


Fig. 13 Image utilization (%) when plaintext size is constant

All the above three experiments reveal that the proposed algorithm apparently outperforms existing algorithm regardless of cover image and plaintext size.

VI. CONCLUSION

The major contribution of this paper is a novel blending technique for maintaining the integrity of secret message sent inside a stego image. This work provides an extra security layer which makes the cryptanalysis harder. Results have shown that the proposed algorithm apparently outperforms existing algorithm regardless of cover image and plaintext size in terms of size of data sent over network and pixel utilization.

REFERENCES

[1] Abbas Cheddad, *"Digital Image Steganography: Survey and Analysis of Current Methods,"* Elsevier, Signal Processing, Vol.90, No.3, Mar.2010, pp.727–752.

[2] Atallah M. Al-Shatnawi, *"A New Method in Image Steganography with Improved Image Quality,"* Applied Mathematical Sciences, Vol.6, No.79, 2012, pp.3907-3915.

[3] Hayfaa Abdulzahra Atee et al., *"Cryptography and Image Steganography Using Dynamic Encryption on LSB and Color Image Based Data Hiding,"* Middle-East Journal of Scientific Research, Vol.23, No.7, 2015, pp.1450-1460.

[4] http://www.facweb.iitkgp.ernet.in/~sourav/AES.pdf

[5] Pragya Agarwal et al., *"Transmission and Authentication of Text Messages through Image Steganography,"* IJCA Proceedings on 4th International IT Summit Confluence 2013, No.2, pp.16-20.

[6] Rinu Tresa et al. *"A Novel Steganographic Scheme Based On Hash Function Coupled With Aes Encryption,"* Advanced Computing: An International Journal (ACIJ), Vol.5, No.1, Jan.2014, pp.25-34.

[7] Sandeep et al., *"A Review on the Various Recent Steganography Techniques,"* IJCSN International Journal of Computer Science and Network, Vol.2, No.6, Dec.2013, pp.142-156.

[8] Shamim Ahmed Laskar et al., *"High Capacity data hiding using LSB Steganography and Encryption,"* International Journal of Database Management Systems (IJDMS) Vol.4, No.6, Dec.2012, pp.57-68.

[9] Shamim Ahmed Laskar et al., *"High Capacity data hiding using LSB Steganography and Encryption,"* International Journal of Database Management Systems, Vol.4, No.6, Dec.2012, pp.57-68.

[10] Vijay Kumar Sharma et al., *"A Steganography Algorithm for Hiding Image in Image by Improved LSB Substitution by Minimize Detection,"* Journal of Theoretical and Applied Information Technology, Vol.36, No.1, Feb.2012, pp.1-8.

