ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

Separable Data Hiding Using AES Algorithm

Sneha Rankhambe[1], Shital Sasane[2], Kirti Shelke[3]

Electronics and Telecommunication Department, Savitribai Phule Pune University,

Bharati Vidyapeeth's college of Engg Women's, Pune, Maharashtra, India[1,2,3]

Abstract: In this paper, the problem of transmitting redundant data over an insecure, bandwidth-constrained communications channel is discussed. A content owner encrypts the original uncompressed image using an encryption key. Then, a data-hider may compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodate some additional data. Using data hiding key the receiver can extract additional data even the receiver has no information about the original image content. Using the decryption key the receiver can extract data to obtain an image similar to the original one, but cannot extract the additional data. If the receiver has both the data-hiding key and the encryption key, the receiver can extract the additional data and the original image without any loss.

Keywords: Author Guide, Article, Camera-Ready Format, Paper Specifications, Paper Submission.

1. INTRODUCTION

As an effective and popular means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption. However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource.

The source is first compressed to its entropy rate using a standard source code. Then, the compressed source is encrypted using one of the many widely available encryption technologies. At the receiver, decryption is performed first, followed by decompression. Compression of encrypted data has attracted considerable research interest. The traditional way of securely and efficiently transmitting redundant data is to first compress the data to reduce the redundancy, and then to encrypt the compressed data to mask its meaning. At the receiver side, the decryption and decompression operations are orderly performed to recover the original data. However, in some application scenarios, a sender needs to transmit some data to a receiver and hopes to keep the information confidential to a network operator in provides the channel resource for the transmission. That means the sender should encrypt the original data and the network provider may tend to compress the encrypted data without any knowledge of the cryptographic key and the original data At receiver side, a decoder integrating decompression and decryption functions will be used to reconstruct the original data.

2. SYSTEM BLOCK DIAGRAM

This section describes the block diagram of the system which includes Encryption and Decryption of the image & Encryption and Decryption of the data.

The separate algorithm for encryption and decryption is given below.

2.1 Encryption

1. Image is encrypted using AES algorithm.
2. The secrete data is separately encrypted by AES algorithm and then we get the encrypted data.
3. The Advanced Encryption Standard (AES) specifies a cryptographic algorithm that can be used to protect electronic data.
4. The data hiding operation is done by using LSB replacement
5. Least Significant Bit (LSB) embedding is a simple strategy to implement steganography
6. Then we get the final encrypted stego image.

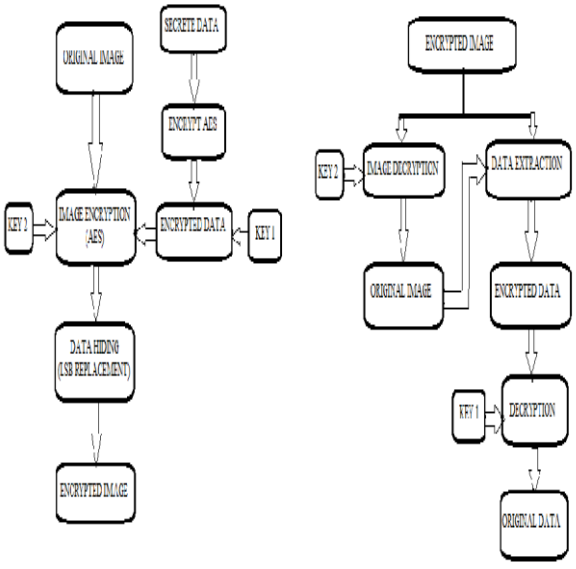

Figure 1. System Block Diagram

2.2 Decryption

1. Encrypted stego image is taken & after that image is decrypted by AES algorithm.
2. From that decrypted image we get the original image.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

3. Then the data extraction is done from that original image, i.e image and data both are separated.
4. Hence we get the encrypted data.
5. Then encrypted data is decrypted using AES algorithm and we get the original data.
6. At receiver side, a decoder integrating decompression and decryption functions will be used to reconstruct the original data.

3. SOFTWARE SPECIFICATIONS

3.1 AES Algorithm

-It is used for encryption & decryption.

3.2 LSB Replacement algorithm

-It is used for Steganograph

4. DESCRIPTION

4.1 AES Algorithm:

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called cipher text; decrypting the cipher text converts the data back into its original form, called plaintext. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits. AES is an iterated symmetric block cipher, which means that:

· AES works by repeating the same defined steps multiple times.
· AES is a secret key encryption algorithm.
· AES operates on a fixed number of bytes

AES as well as most encryption algorithms is reversible. This means that almost the same steps are performed to complete both encryption and decryption in reverse order. The AES algorithm operates on bytes, which makes it simpler to Implement and explain.

This key is expanded into individual sub keys, a sub keys for each operation round. This process is called KEY EXPANSION, which is described at the end of this document.

As mentioned before AES is an iterated block cipher. All that means is that the same operations are performed many times on a fixed number of bytes. These operations can easily be broken down to the following functions:

- ADD ROUND KEY
- BYTE SUB
- SHIFT ROW
- MIX COLUMN

4.2 LSB Replacement

Least Significant Bit (LSB) embedding is a simple strategy to implement steganography. Like all steganographic methods, it embeds the data into the cover so that it cannot be detected by a casual observer. The technique works by replacing some of the information in a given pixel with information from the data in the image. While it is possible to embed data into an image on any bit-plane, LSB embedding is performed on the least significant bit(s). This minimizes the variation in colors that the embedding creates. For example, embedding into the least significant bit changes the color value by one. Embedding into the second bit-plane can change the color value by 2. If embedding is performed on the least significant two pixels, the result is that a color in the cover can be any of four colors after embedding.

Steganography avoids introducing as much variation as possible, to minimize the likelihood of detection. In a LSB embedding, we always lose some information from the cover image. This is an effect of embedding directly into a pixel.

To do this we must discard some of the cover's information and replace it with information from the data to hide. LSB algorithms have a choice about how they embed that data to hide.

They can embed lossless, preserving all information about the data, or the data may be generalized so that it takes up less space.

Detection of LSB embeddings is done through two types of attacks. The first is a simple visual attack, which relies on the human eye to evaluate an image. The second is a statistical attack, which analyzes images in a statistical manner.

While LSB embeddings may be attacked visually and statistically, other techniques may only be attacked using statistical analysis. Jsteg is an example of such an algorithm where embedding does not take place in the pixels of an image, and instead the effects of small changes to the DCT coefficients must be detected. In such cases a statistical attack is the only viable option.

Again, these attacks are typically tailored to an embedding strategy, as is evidenced by Westfield and Pfitzmann's success attacking several steganographic algorithms with their Chi-square attack by measuring the expected distribution of values with those observed in a given image Embedding strategies may be easily derived and implemented to complicate detection and inhibit the retrieval of the message by a third party, while still allowing easy retrieval by the intended recipient.

LSB Embeddings may be detected simply through visual inspection of an image and its bit-planes, or more reliably through methods which use statistical metrics to identify the likelihood an image contains hidden data. While an embedding may be detected, it may not be easily decoded, nor may a stego object be discovered due to the sheer number of images available.

Steganography proves to be a significant technique for evading detection when communicating. The detection issues with steganography create challenges for security systems in attempting to prevent the transmission of steganographic content.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 3, March 2015

5. TABLE

	Key Length *(Nk words)*	Block Size *(Nb words)*	Number of Rounds *(Nr)*
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Figure 2. Key-Block-Round Combinations

6. FLOWCHART

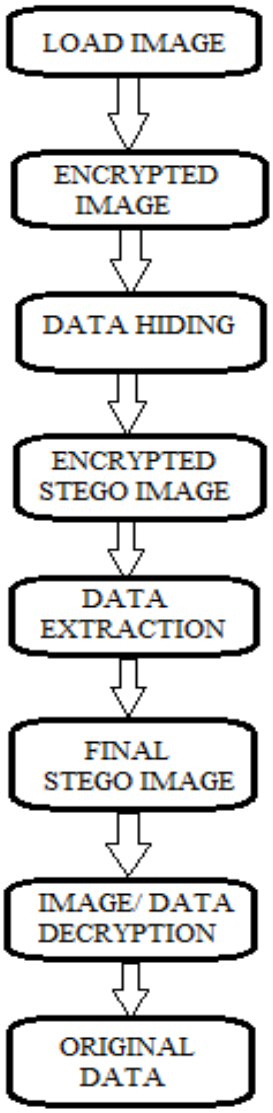

Figure 3. Flowchart

7. CONCLUSIONS

Thus, we have successfully encrypted the image using ADVANCED ENCRYPTION STANDARD (AES) algorithm. This standard specifies the encryption of the image followed by secrete data using the key decryption of data is done successfully. The secrete message which is send by the sender is received by the receiver using the secret key only. Hence we successfully demonstrate the project.

ACKNOWLEDGMENTS

With great pleasure and satisfaction we present a seminar on "Separable Data Hiding Using AES Algorithm."
We would like to take this opportunity to express true sense of gratitude towards our project guide **Prof. P. R. Yawle** for valuable co-operation and guidance that she gave us throughout the project and also the personal involvement and constructive criticism provided beyond technical guidance during project.
Finally, we thank all the people who have directly or indirectly help us through the course of our project.

REFERENCES

[1]. X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inform. Forensics Security, vol. 6, no. 1, pp. 53–58, Feb. 2011.
[2]. W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," IEEE Trans. Image Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.
[3]. T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," IEEE Trans.Inform. Forensics Security, vol. 5, no. 1, pp. 180–187, Feb. 2010.
[4]. M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A Neri, "A commutative digital image watermarking and encryption method in the tree structured Haar transform domain," *Signal Processing:*