Enhancement of Data Hiding Process in Encrypted Image Using Advanced Encryption Standard

Deepthi Barbara Nickolas*, Sindhuja.B, Sivasankar.A

*Department of Electronics and Communication Engineering, Anna university, Regional centre, Madurai, Tamilnadu, India.

Accepted 29 March 2013, Available online 1 June 2013, Vol.3, No.2 (June 2013)

Abstract

In this paper a scheme known as separable reversible data hiding is proposed for the secured transmission of data. This scheme uses Advanced Encryption Standard for secured data transmission. The basic technique used here is image encryption, data embedding along with data extraction and image recovery phases which uses same keys for both encryption and decryption. To make the data transmission process secure additional data is added using the data hiding key so that original data can’t be extracted without both encryption and data hiding key. The data embedding technique uses data compression which is a lossless technique creating space for the additional data to be added. This technique using the AES algorithm improves the PSNR ratio.

Keywords: Image encryption, data hiding, image recovery.

1. Introduction

In recent days security is a big threat in the transmission medium due to the development of the Internet and multimedia contents such as audio, image, video, etc. It enables us to easily purchase digital contents via the net. However, it causes several problems, such as violation of ownership and illegal distribution of the copy. The basic idea followed is based on cryptography technique. Here symmetric encryption technique is used, which uses the same keys for both encryption and decryption.

Data hiding is referred to as a process to hide data into cover media. This implies that the data hiding process links two sets of data, a set of embedded data and another set of cover media data. These two sets of data have different applications. In covert communications, the hidden data may often be irrelevant to the cover media. In authentication, the embedded data is closely related to the cover media. In these two types of applications, the invisibility of hidden data is an important factor. In some cases of data hiding, the cover media will experience some distortion due to data hiding and cannot be inverted back to original media. That is, some permanent distortion has occurred to the cover media even after the hidden data have been extracted out. In other applications, such as remote sensing and high-energy particle physical experimental analysis, it is also desired that the original cover media can be recovered because of the required high-precision. The marking techniques satisfying this requirement are referred to as reversible, lossless and distortion-free data hiding techniques. Reversible data hiding is a technique to embed additional message into some distortion free cover media, such as military or medical images, in a reversible manner so that the original cover content can be perfectly restored after extraction of the hidden message. As a popular and effective means of privacy protection, encryption process converts the ordinary signal into unintelligible data, so the traditional signal processing steps takes place before encryption or after decryption.

In [1], an interactive buyer–seller protocol for invisible watermarking, the seller does not know the exact watermarked copy that the buyer receives. So the seller cannot create copies of the original content containing the buyer’s watermark. If the seller finds an unauthorized copy, the seller can identify the buyer from the watermark using the unauthorized copy, and hence the seller can prove this fact to a third party using a dispute resolution protocol. In [2] it is desired to transmit redundant data over an insecure and bandwidth-constrained channel, it is desired to first compress the data and then encrypt it. Traditional techniques used to compress the data first and then encrypt the data. But we are reversing the order of these steps, thereby first encrypting and then compressing the data. A significant compression ratio can be achieved if compression is performed after encryption. Compression is performed by standard source coder and decryption by decompression.

In [3] we propose a new fingerprinting protocol applying additive homomorphic property of Okamoto–Uchiyama encryption scheme. Exploiting the property, the enciphering rate of our fingerprinting scheme can be
improved. The security can also be protected for both a buyer and a merchant since the protocol is performed between only two parties, a buyer and a merchant, which is similar to a real-world market. In [4] reversible data hiding algorithm, it can recover the original image without any distortion from the marked image after the hidden data have been extracted. This algorithm utilizes zero or minimum points of the histogram of an image and modifies the pixel grayscale values to embed data into the image.

In [5] this scheme is proposed to implement commutative video encryption and watermarking during advanced video coding process. In [6] signal-processing modules works directly on encrypted data to provide a solution to scenarios where valuable signals must be protected from a malicious processing device. Here, we investigate the implementation of the discrete Fourier transform (DFT) in the encrypted domain by using the homomorphic properties of underlying cryptosystem.

In [7] signal processing tools works directly on encrypted data, where sensitive signals must be protected from an untrusted processing device. Here, we consider the data expansion required to pass from the plaintext to the encrypted representation of signals. A general composite signal representation allows us to pack together a number of signal samples and process them as a unique sample. In [8] the lossy compression method presented in an encrypted gray image can be efficiently compressed by discarding the excessively rough and fine information of coefficients generated from orthogonal transform. When having the compressed data, a receiver may reconstruct the principal content of original image by retrieving the values of coefficients. In [9] the work proposes a reversible data hiding scheme for encrypted image. After encrypting the entire data of an uncompressed image the additional data can be embedded into the image by modifying a small portion of encrypted data. With an encrypted image containing additional data, one may first decrypt it using the encryption key, and the decrypted content is similar to the original image. According to the data-hiding key, the embedded data can be successfully extracted and the original image can be perfectly recovered. In [10] hi-tech secure products need more services and more security. In this paper, we have implemented Advanced Encryption Algorithm for ARM based platforms. AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. AES requires best exploitation of architectural parameters in order to reach the optimal performance on specific architectures.

2. Proposed Methodology

The proposed scheme uses AES encryption algorithm which uses the same keys for both encryption and decryption. AES is an iterated block cipher with a fixed block size of 128 and a variable key length. AES uses variable number of rounds, which are fixed. The key expansion algorithm is based on the assumption of a 128 bit key. AES calls for a larger number of rounds when we use a key length other than 128 bits. The key expansion algorithm must obviously generate a longer schedule for the 12 rounds required by a 192 bit key and the 14 rounds required by a 256 bit keys. If a change in one bit of the encryption key occurs it will affect the round key for several rounds. The key expansion algorithm ensures that AES has no weak keys During each round, following operations are performed on each state

1. Sub Bytes: every byte in the state is replaced by another one, using the S-Box.
2. Shift Row: every row in the 4x4 array is shifted a certain amount to the left.
3. Mix Column: a linear transformation on the columns of the state.
4. Add Round Key: each byte of the state is combined with a round key.

2.1 The Substitute Bytes Step

This is a byte-by-byte substitution and the substitution byte for each input byte is found by using the lookup table. The size of the lookup table is 16 × 16. To find the substitute byte for a given input byte, we divide the input byte into two 4-bit patterns, each yielding an integer value between 0 and 15. We can represent these integer values by their hex values 0 through F. One of the hex values is used as a row index and the other as a column index for reaching into the 16×16 lookup table. The entries in the lookup table are constructed by a combination of GF (2^8) arithmetic and bit mangling. The goal of the substitution step is to reduce the correlation between input bits and output bits at the byte level. The bit mangling part of the substitution step ensures that the substitution cannot be described in the form of evaluating a simple mathematical function. We first fill each cell of the 16 × 16 table with the byte obtained by joining together its row index and the column index. We next replace the value in each cell by its multiplicative inverse in GF (2^8) based on the irreducible polynomial x^8+x^7+x^6+x+1. The value 00 is replaced by itself since this element has no multiplicative inverse. Let’s represent a byte stored in each cell of the table by b_7b_6b_5b_4b_3b_2b_1b_0 where b_7 is the MSB and b_0 the LSB. Therefore, the bit pattern stored in the cell with row index 9 and column index 5 is 10001010, showing that b_7 is 1 and b_0 is 0.

This byte-by-byte substitution step is reversed during decryption. The 16×16 lookup table for decryption is constructed by starting out in the same manner as for the encryption lookup table.

2.2 The Shift Row Step

The Shift Row transformation consists of

1. Not shifting the first row of the state array at any condition.
2. Circularly shifting the second row by one byte to the left side.
3. Circularly shifting the third row by two bytes to the left side.
4. Circularly shifting the last row by three bytes to the left side.

The input block is written in column-wise manner. That is the first four bytes of the input block fill the first column of the state array and the next four bytes the second column and so on. For decryption, the encryption steps are shifted in exactly the opposite fashion. Initially the first row is left unchanged. The second row is shifted to right by one byte and the third row to right by two bytes and the last row to right by three bytes. All shifts performed are circular manner.

2.3 Mix Column Step

This step replaces each byte of a column by a function of all the bytes in the same column. More precisely, each byte in a column is replaced by two times that byte, plus three times the next byte, plus the byte that comes next, plus the byte that follows. The words ‘next’ and ‘follow’ refer to bytes in the same column, and their meaning is circular, in the sense that the byte that is next to the one in the last row is the one in the first row. This stage known as Mix Column is basically a substitution but it makes use of arithmetic of GF ($2^8$). Each column is operated individually. Each byte of each column is mapped into a new value that is a function of all four bytes in the column.

2.4 Add Round Key Transformation

This stage known as AddRoundKey uses the 128 bits of state and is bitwise XORed with the 128 bits of the round key. This operation is performed in column wise manner. This transformation is as simple as possible which helps in efficiency but it affects every bit of state.

3. Experimental Results

As per the AES Encryption process first the owner sends an image, it is then encrypted and then additional data is added to it using the data hiding key. The data is also encrypted with the encrypted image and at the decryption side the data along with the image can be extracted successfully without any distortion if the receiver has both data hiding and encryption key. The output images obtained are displayed below.

Conclusion

In this paper a separable reversible data hiding in encrypted image is proposed which uses the AES Algorithm to encrypt and decrypt the data along with the image. This can be further improved using AES_CCMP (Advanced Encryption Standard – Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) with non-linear arrangement of S-box.

References