Data Embedding Method Using Video Pixel Pair Technique

Saleena Thongam[1], Manjima R.L[2]

[1]M. Tech CSE, CMRIT Dept of CSE, Imphal, Manipur, India

[2]Asst Professor CMRIT, Dept of CSE, Imphal, Manipur, India

Abstract: ***Data hiding is a kind of secret sharing scheme. This paper proposes a new data hiding method using Pair Pixel technique which mainly focuses on Videos. The values of pixel pair will be used as a reference coordinate and search the coordinates in the neighborhood set of the pixel pair according to a message digit. The existing pixel pair will be replaced by the search coordinates to conceal the digit. This paper also proposes the idea of the visual cryptography scheme (VCS) which is a secret sharing scheme. Comparison for security level of the existing system and the proposed system is done and proves that the proposed system is more secure than the existing one.***

Keywords: LSB, Splay tree, Data hiding

1. Introduction

Data embedding involves the idea of data hiding. Data hiding is a technique for conveying secret message confidentially by concealing the data into a carrier [1] [2]. Let us consider internet as an example as most data transfer in this generation takes place through it. Images in data hiding can be classified as two types namely cover images and stego images. After embedding, distortion and the modification of the pixels of the cover images can occur which is referred to as the embedding distortion [3] Virtual Cryptography scheme is also a kind of secret sharing scheme [5] [6]. It can also be considered as the steg analysis scheme is also a kind of VCS which contains only the meaningful shares [7].

The Least Significant Bit (LSB) substitution method is also one of the data hiding method which can be implemented easily with low cost CPU. It is one of the most popular embedding techniques. Generally it can be define as the bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. The LSB is sometimes referred to as the right-most bit, due to the convention in positional notation of writing less significant digit further to the right. A trivial modification of LSB replacement [4] is LSB matching (also called embedding), which randomly increases or decreases pixel values by one to match the LSBs with the communicated message bits.

LSB matching and EMD are the two embedding method which offers no mechanism to increase the payload. In 2008, Hong [8] presented a data-hiding method based on Sudoku solutions to achieve a maximum payload of bpp. In 2009, Chao *et al.* [9] proposed a diamond encoding (DE) method to enhance the payload of EMD further. DE employs an extraction function to generate diamond characteristic values (DCV), and embedding is done by modifying the pixel pairs in the cover image according to their DCV's neighborhood set and the given message digit.

This paper proposes a new data embedding method known as APPM to reduce the embedding impact by providing a simple extraction function and a more compact neighborhood set. It also make used of the new encryption algorithm called KIST (key Insertion and splay tree). Making used of this algorithm can make the encryption fast, uses small space and good for message integrity. The other section of this paper explains about the methodologies and the comparison of the experimental result.

KIST algorithm which is abbreviated as Key Insertion and Splay tree algorithm is a new algorithm which uses an asynchronous key sequence and a splay tree. Some of the characteristics and the advantages of the KIST algorithm is as follows

- An asynchronous key sequence is used, which depends on an initial key and plaintext encrypted.
- A splay tree is used so that the substitution is dynamic.
- The encryption is fast and uses small space.
- Cipher texts are compressed in most cases.
- The block size of the plain text and key size are flexible.
- It is good for massage integrity.

2. Related works

OPAP effectively reduces the image distortion compared with the traditional LSB method. This section will briefly discuss about the OPAP and DE.

2.1 Optimal pixel adjustment process

The OPAP method proposed by Chan et al. in 2004 greatly improved the image distortion problem resulting from the LSB replacement. The OPAP method is described as follows [10], [11].Suppose a pixel value is v, the value of the right-most LSBs of v is $v^{(r)}$. Let v' be the pixel value after embedding r message bits using the LSB replacement method and s be the decimal value of these r message bits. OPAP employs the following equation to adjust v' so that the embedding distortion can be minimized

$$v'' = \begin{cases} v' + 2^r, v^{(r)} - s > 2^{r-1} \\ \quad and\ v' + 2^r \leq 255 \\ v' - 2^r, v^{(r)} - s < 2^{r-1} \\ \quad and\ v' - 2^r \geq 0 \\ \quad v', otherwise \end{cases}$$

where v'' denotes the result obtained by OPAP embedding. Note that v'' and v' have the same right-most r LSBs and thus, the embedded data can be extracted directly from the right-most r LSBs.

2.2 Diamond Encoding

In 2009, Chao *et al.* proposed a DE method based on PPM. This method conceals a secret digit in a ß-ary notational system into two pixels, where ß = $2k^2 + 2k + 1$, k≥ 1 . The payload of DE is (1/2)$\log_2(2k^2 + 2k + 1)$ bpp. Note that when k =1, DE is equivalent to EMD in which both methods conceal digits in a 5-ary notational system. The DE method is briefly described as follows. Let the size of m bits cover image be M× M, message digits be S_B , where the subscript represents is in a ß -ary notational system. First, the smallest integer k is determined to satisfy the following equation:

$$\left\lfloor \frac{M \times M}{2} \right\rfloor \geq |S_B|$$

where $|S_B|$ denotes the number of message digits in a ß-ary notational system. To conceal a message digit s_B into pixel pair (x.y), the neighborhood set $\phi(x.y)$ is determined by $\phi(x.y) = \{(a,b)||a-x| + |b-y| \leq k\}$ where $\phi(x.y)$ represents the set of the coordinates (a,b)'s whose absolute distance to the coordinate (x,y) is smaller or equal to k. A diamond function is then employed to calculate the DCV of (x,y), where f(x,y) =((2k+1)x+y) mod ß. After that, the coordinates belong to the set $\phi(x.y)$ are searched and DE finds a coordinate (x', y') satisfying f(x', y') = s_B, and then (x,y) is replaced by . Repeat these procedures until all the message digits are embedded. In the extraction phase, pixels are scanned using the same order as in the embedding phased. The DCV value of a pixel pair (x', y') is then extracted as a message digit.

This paper proposes the encryption algorithm which consists of three parts which can be described as follows:

1. Key generation algorithm.
2. Encryption algorithm.
3. Key injection algorithm.

2.2.1 Key Generation Algorithm

As the name suggest key generation is the process of generating a key for the process of cryptography. This key generation process is necessary for the data to encrypt or decrypt. Our method of key generation is as follows. After a key is generated, the algorithm will check the layer of the inner node. The key is ignored if the inner node is less than N (N is some parameter). Using this node some keys are discarded randomly. Thus the security can be increased by injecting only some random keys.

The algorithm of *injection function* **injection (byte I, integer N)** is as follows. In this algorithm, a byte is expressed as an integer between 0 and 255.

Algorithm: INJECTION (i ,N)
Comment: i is a byte from input
j ← i
if (i = 0)
then quit
s ← 1
N ← n
while (up[j] 6= 0)
j ← up[j], s ← s + 1
if (s < N)

$$\text{then} \begin{cases} if\ (j = right[0]) \\ then\ t \leftarrow left\ [0] \\ else\ t \leftarrow right[0] \\ exchange\ links\ of\ i\ and\ t\ (swap\ up[i] and\ up[t]) \end{cases}$$

else quit

At the beginning of the encryption the injection function will be called. At the same time, 16 bytes will be injected to the initial tree.

Now let us consider the key generation algorithm. The key sequence is generated from the initial key and the plain text. For this algorithm "cyclic" array key with length 16 (or the length of the key is used). Here the cyclic key is represented as *key[j]= key[j-16]for j≥16* which is represented in the array key.

Algorithm: KEY GENERATION (P,K)
Comment: P is the plain text and K is the initial key

For i=1 to 16
do key(i)←K_i
c ← 17
for j= 1 to m

$$\text{do} \begin{cases} \{key(c) = key(c) \oplus up[p_j + 255] \\ output\ (key\ (c)) \\ c \leftarrow c + 1 \end{cases}$$

The ith key is generated from the initial key and p_j, where p_j contains first j bytes of plain text.

2.2.2 Encryption algorithm

The encryption algorithm we are using here is the splay tree algorithm plus the kye injection algorithm. The algorithm of the splay function Splay (byte i) is as follows.

Algorithm : SPLAY (i)
Comment: i is a byte from input

$i \leftarrow i + 255$
J ← $up[i]$
While (j ≠ 0 $and\ up[j] \neq 0$)

$$\begin{cases} s \leftarrow up[j] \\ if\ (j = left[s]) \\ then\ t \leftarrow right\ [s] \\ else\ t \leftarrow left[s] \\ exchange\ links\ of\ i\ and\ t \\ i \leftarrow up[i] \\ j \leftarrow up[i] \end{cases}$$

The encode is proceeded byte by byte. The algorithm for the encode function encode (byte i) is as follows.

Algorithm: ENCODE (i)
Comment: i is a byte from input.
j ← i
i ← i + 255
while (i ≠ 0)

$$\begin{cases} if\ (i = left[up[i]]) \\ then\ push\ bit\ 0\ to\ stack \\ else\ push\ bit\ 1\ to\ stack \\ i \leftarrow up[i] \end{cases}$$

while (stack is not empty) pop a bit and output it Splay(j)

The algorithm first encodes the byte and the access node is splayed. A key sequence is created by the key generation algorithm to perform the encryption. The key sequence is $(K_1, K_2, \ldots\ldots\ldots K_i, \ldots.$, where $0 \leq K_i \leq 255$ for each i and the initial key is given as K= (K_1,K_2,……,K_{16}). The plain text is in bytes given as P= $(p_1, p_2, \ldots\ldots\ldots, p_m)$.

Algorithm 5: Encryption (P, K)
Comment: P is plain text and K is key sequence
for i = 1 to 16
do injection(K_i, N)
key ← 17
for j = 1 to m

$$\text{do} \begin{cases} encode\ (p_j) \\ injection\ (K_{key}, N) \\ key \leftarrow key + 1 \end{cases}$$

2.2.3 Decryption algorithm

The decoding process is performing bit by bit following a path from root to leaf. The parameter N is same as that in the encryption used. Assume the cipher text is a bit string C= $(c_1, c_2, \ldots\ldots, c_s)$. After proceeding a bit will be deleted from the bit string and gives the output of the function is a byte. The algorithm of the decode function decode (bit string C) is as follows.

Algorithm 6: DECODE (C)
Comment: C is a bit string from input
node ← 0
c ← first bit of C
while (node ≤ 254)

$$\begin{cases} if\ (c = 0) \\ then\ node \leftarrow left\ [node] \\ else\ node \leftarrow right\ [node] \\ C = C\ \{c\} \\ c \leftarrow first\ bit\ of\ C \end{cases}$$

Output (node – 255)
Splay (node -255)

The algorithm is decrypted as follows. If the cipher text is a bit string C and the key sequence is generated by K . As in encryption, the algorithm first inject 16 keys and the decode function is called. The key injection is called for every decoding.

Algorithm: DECYPTION (C, K)
Comment: C is bitstring and K is key sequence
For i=1 to 16

do injection(K_i, N)
key ← 17
j ← 0

$$\text{while (C} \neq \emptyset) \begin{cases} decode\ (C) \\ j \leftarrow j + 1 \\ injection\ (K_{key}, N) \\ key \leftarrow key + 1 \end{cases}$$

2.2.4 Key Generation Algorithm

Comment: P is Plain text and K is initial key
For i = 1 to 16
do *key(i)* ← K_i
c ← 17
for j =1 to m

$$\text{do} \begin{cases} key\ (c) = key\ (c) \oplus up\ [p_j + 255] \\ output\ (key(c)) \\ c \leftarrow c + 1 \end{cases}$$

The ith key is generated from the initial key and P_j, where P_j contains first j bytes of plain text. The parent P_j is not fixed and depends on previous plain text and the initial key.

3. Comparison

Four images Lena, Jet, Boat, Elain each sized 512× 512, are taken as test images to compare the MSE obtained by APPM, OPAP, and DE. The payloads were set to 400 000, 650 000, and 1 000 000, respectively. Message bits were generated by using a pseudorandom number generator (PRNG).

Figure 2: Cover image and stego images under various payloads. (a) Cover image. (b) Stego image, 2 bpp at 46.86 dB. (c) Stego image, 3 bpp at 40.97 dB. (d) Stego image, 4 bpp at 34.90 dB. The simulation results of the comparison of the system is as shown

Figure 3 shows the comparison of the exiting and the proposed system

4. Conclusion

This paper proposed a new data hiding method a new data hiding method based on the pair pixel matching mostly concentrated on the videos. Two pixels are scanned as an embedding unit and a specially designed neighborhood set is employed to embed message digits with a smallest notational system. APPM allows the user to select digit in any notational system for data embedding, and thus achieve a better quality image. The proposed method not only resolves the low payload problem in EMD but also offers smaller MSE compared with OPAP and DE. It also offers secure communication under adjustable embedding capacity.

The future enhancement of this project includes more Encoding techniques can be used to give the host computer a wider choice of selecting the same. Software can be enhanced to support more than one client with the same command terminal. High compression algorithms can be used to capture videos from the server computer.

References

[1] J. Fridrich, Steganography in Digital Media: Principles, Algorithms,and Applications. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[2] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," IEEE Security Privacy, vol. 3, no. 3, pp. 32–44, May/Jun. 2003.
[3] Cheddad, J. Condell, K. Curran, and P. McKevitt, "Digital image steganography: Survey and analysis of current methods," Signal Process., vol. 90, pp. 727–752, 2010.
[4] Wien Hong and Tung-Shou Chen A "Novel Data Embedding Method Using Adaptive Pixel Pair Matching" IEEE Traans on Information Forensic and Security Vol.7 No.1 Year 2012
[5] Shamir, "How to share a secret," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.
[6] G. R. Blakley, "Safeguarding cryptographic keys," in Proc. National Computer Conf., 1979, vol. 48, pp. 313–317.
[7] Feng Liu and Chuankun Wu, " Embedded Extended Visual Cryptography Schemes" IEEE Transaction on Information forensic and security ,Vol. 6, No. 2, June 2011
[8] W. Hong, T. S. Chen, and C. W. Shiu, "A minimal Euclidean distance searching technique for Sudoku steganography," in Proc. Int. Symp. Information Science and Engineering, 2008, vol. 1, pp. 515–518.
[9] R.M. Chao, H. C. Wu, C. C. Lee, and Y. P. Chu, "A novel image data hiding scheme with diamond encoding," EURASIP J. Inf. Security, vol. 2009, 2009, DOI: 10.1155/2009/658047, Article ID 658047.
[10] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognit., vol. 37, no. 3, pp. 469–474, 2004.
[11] C. H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution," Pattern Recognit., vol. 41, no. 8, pp. 2674–2683, 2008.

Author Profile

Saleena Thongam has received her BE degree in Computer Science and Engineering and pursuing her MTech in Computer Science in CMR Institute of Technology.

Manjima R.L has received her M Tech Degree in Computer Science and is currently working as an assistant professor in CMR Institute of Technology.