Indian Journal of Science and Technology, *Vol 8(16), 51269, July 2015*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Novel Data Hiding Approach by Pixel-Value-Difference Steganography and Optimal Adjustment to Secure E-Governance Documents

Tanmoy Halder[1*], Sunil Karforma[2] and Rupali Mandal[3]

[1]Department of Computer Application, Dr. B.C. Roy Engineering College, Durgapur - 713206, West Bengal, India; tanmoyhalder@gmail.com
[2]Department of Computer Science, The University of Burdwan, Burdwan - 713104, West Bengal, India; sunikarforma@yahoo.com
[3]Department of Computer Application, Bengal College of Engineering and Technology, Durgapur - 713212, West Bengal, India; rupa.mandal23@gmail.com

Abstract

In this paper, we have proposed a new data hiding approach using Pixel Value Difference (PVD) seganography for digital image. PVD steganography, proposed by Wu and Tsai, use non-overlapping block of two pixels to find the edge areas to hide secret message by adjusting the pixel pairs. In PVD method only the edge between two pixels within a block is detected. In order to find out more edges and to increase the capacity of PVD method we have used non-overlapping blocks of three pixels and an optimal adjustment process. The proposed method compared with different PVD approaches with respect to capacity and PSNR. This novel method could be applied to hide E-Governance related data for secure transfer.

Keywords: E-Governance, Optimal-Adjustment, Pixel-Value-Difference, Steganography

1. Introduction

E-Governance is an application of G2C E-commerce through which government can communicate with consumers by internet. Lots of E-Governance projects are now on a way to successful implementation in different sectors, from rural areas to corporate billing and Tax collection. The main threat over these E-Governance projects is insecurity of online document transaction. Important documents like voter card, PAN card may be misused by unauthorized hackers on the way to transfer, due to the reason it became necessary to protect important documents transferring through insecure channels like internet. Steganography algorithms are potential methods to prevent unauthorized access by unknown people and attest the original documents.

Steganography is an old art of data hiding. It differs from cryptography; in cryptography presence of hidden data is noticeable, whereas in steganography nobody expects the sender and receiver could be able to know about hidden message. Modern Digital Steganography uses a covering media (like image, video, sound etc). To hide secret information using some secret process.

The best-known image steganographic method that works in the spatial domain is the LSB (Least Significant Bit) replacement[1–3]. Kevin Curran and Karen Bailey[4] analyze seven different image based steganography methods. Some of these methods are Stego1bit, Stego2bits, Stego3bits, Stego4bits and Stego Color Cycle[4]. Chan et al.[3], proposed a method, which substitutes the same number of bits of each and every pixel of the cover images for hiding the secret message[3]. Chang et al.[5], Thien et al.[6]

**Author for correspondence*

proposes that LSB substitution is the most commonly used method directly replacing the LSBs of pixels in the cover image with secret bits to get the stego-image. LSB substitution algorithm is the simplest scheme to hide message in a host image[3,5,6]. T. Halder and S. Karforma introduced indexing over the existing LSB matching scheme, which does not store the bits directly within the LSB positions, instead bits are distributed by matching/replacing bits within the whole image and use the LSB bits as an index to remember the positions[7].

Recent steganolysis found out that, if an image is processed with simple LSB substitution the histogram of the image will be showed in a "pair-wise" manner. These pair-wise blocks are known as Pairs of Values (PoV) which can be identified by Chi-square Test given by[8,9]. Another steganalysis method RS steganalytic algorithm by Fridrich et al.[10] is able to detect the existence of hidden data at LSB positions. The detection capability of the RS steganalytic algorithm depends on the capacity of the hidden message. Specifically, the algorithm can detect the existence of the LSB scheme with high precision when the hidden capacity is more than 0.005 bits per pixel[10,11].

Other than LSB methods Wu and Tsai[12] proposed a method known as 'Pixel value differencing' where a non-overlapping block of two pixels are chosen from the cover image. Difference between the pixels point out smooth and edge areas. More number of bits could be hidden in edge areas rather than smooth areas. Applying this methodology PVD method can hide more amount of data than LSB method and more susceptible against visual attack because rather than hiding secret bits throughout the image PVD finds out edge areas to hide large amount of data. Wu et al.[13] proposed another method by combining PVD and LSB replacement scheme which shows better capacity than PVD method. Later this paper modified and analyzed by Yang et al.[14] using R-S diagram.

Cheng-Hsing Yang et al.[15] proposed another data hiding scheme based on PVD method, in this scheme varieties of pixel-value differencing by pair-wise grouping of blocks, shows greater hiding capacity than the PVD method proposed by Wu and Tsai, and also have better capacity than the method proposed by Yang et al.[14] and Wu et al[13].

In the proposed approach in order to increase capacity of the PVD method introduced by Wu and Tsai[12] we have chosen a non-overlapping block of three pixels in a zigzag order to find more edge areas, after finding out smooth and edge areas, as we know edge area is capable of hiding more secret data rather than smooth area, secret data is embedded according to that. The pixels are adjusted using a process to reflect minimum distortion in original image. Distortion in pixels is calculated by comparing the difference values between original pixel and modified pixel by applying different methods. We have also check the values after altering that they must fall within the range of 0 to 255. The proposed method also has greater capacity than the method proposed by Cheng-Hsing Yang et al[15].

Rest of this paper is organized as follows: Sec 2 gives a brief introduction about PVD method, sec 3 is about proposed method, in Sec 4 result and discussion, sec 5 is about conclusion.

2. Review of Wu and Tsai's Scheme

The PVD method proposed by Wu and Tsai takes two consecutive pixels (P and Q) from cover image, the difference (d) between the pixels are calculated as

$$D = |P - Q| \quad (1).$$

Value of D will be within the range from 0 to 255. In a smooth region value of D is less and in a sharp-edge region value of D is high, depending on the value of D number of bits could be chosen from the secret message, The range table Ri (i = 1, 2…..n) calculates the range of D, main purpose of the range table is to calculate the capacity of that pair. Each range has a lower and upper limit says Li and Ui, the width (Wi) of the range is a power of two and is calculated as

$$Wi = (Ui - Li + 1) \quad (2)$$

This restriction of width facilitates the embedding of binary data. If D falls in smooth area, less secret data will be hidden in the block. On the other hand, if D falls in sharp area, then the block has higher tolerance and thus more secret data can be embedded inside it. In this way hiding capacity (Ti) is calculated as

$$\mathrm{Ti} = \left[\lg\left(Wi\right)\right] \quad (3)$$

Here Ti is the number of bits that could be fetched from secret message. Now convert the secret bits in corresponding decimal number (Ti′). The new difference between the pixels P and Q is calculated as

$$D' = Li + Ti' \quad (4)$$

The secret data could be hidden by adjusting P and Q as P′ and Q′, the adjustment is as follows.

$$P', Q' = \begin{cases} (P + \left\lceil \frac{m}{2} \right\rceil, Q - \left\lfloor \frac{m}{2} \right\rfloor) & \text{if } P \geq Q \text{ and } D' > D \\ (P - \left\lfloor \frac{m}{2} \right\rfloor, Q + \left\lceil \frac{m}{2} \right\rceil) & \text{if } P < Q \text{ and } D' > D \\ (P - \left\lceil \frac{m}{2} \right\rceil, Q + \left\lfloor \frac{m}{2} \right\rfloor) & \text{if } P \geq Q \text{ and } D' \leq D \\ (P - \left\lfloor \frac{m}{2} \right\rfloor, Q - \left\lceil \frac{m}{2} \right\rceil) & \text{if } P < Q \textit{ and } D' \leq D \end{cases} \quad (5)$$

Where m = |D′-D|.

3. The Proposed Method

Choosing a block of two pixels (PVD method proposed by Wu and Tsai) will miss some edges, there may be a chance of hiding large data. In our approach a non-overlapping block of three pixels is chosen to find out more edge areas. Figure 1 shows six pixels and edges detected by Wu and Tsai's method and ignored edges. Figure 2 shows in a sequence of six pixels edges detected by our approach and Wu and Tsai's approach.

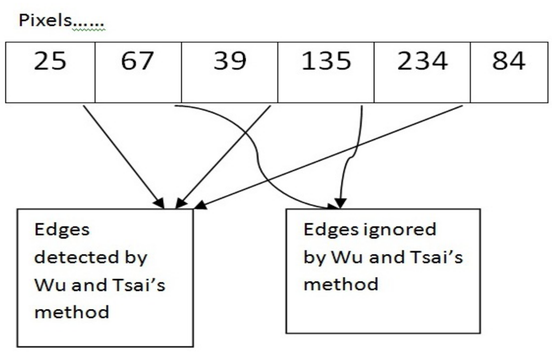

Figure 1. Edges detected and ignored by Wu and Tsai's method.

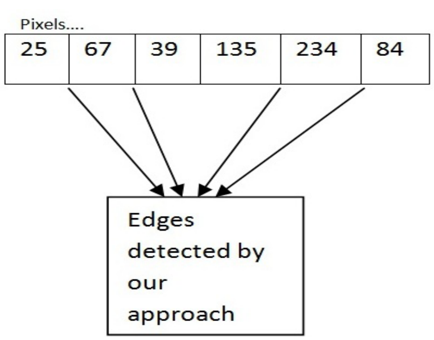

Figure 2. Edges detected by our approach.

From Figure 1 and Figure 2 it is clear that within six pixels our approach can find out four edges whereas PVD method finds out three edges. Now, the problem arises in a block of three pixels is adjustment after hiding data. In a block of three pixels (P, Q, R) there are two edges, they are between P and Q and Q and R. The middle pixel Q is the common part, so there are two choices, either Q is totally ignored only P, Q takes part in adjustment. Or Q is adjusted in a balanced way after hiding data in both parts in the group of three pixels, if the gaps are not totally balanced then rest of the adjustment is distributed among P and R. following this process it is observed that for some group the first adjustment produces optimal distortion in pixel values and for some block the second process reflects optimal distortion in pixels. In order to keep distortions minimal in original pixels and to keep PSNR high we have applied optimality checking process which calculate total distortion in pixels after hiding data then select the process producing minimum changes in pixels. The process is as follows.

3.1 The Embedding Algorithm

Step 1: Block of three pixels in a non-overlapping manner are chosen from cover image in a zigzag order, suppose the pixels are P, Q and R. Choose 1st pair as P and Q, and 2nd pair as Q and R, Obtain the difference value for 1st pair as D = |P–Q| and check the range table Ri (i = 1, 2…..n) to obtain the range where D belongs to. In the same way obtain the difference value D1 = |Q–R| and check the range table Ri (i = 1, 2…..n) to obtain range D1 belongs to.

Step 2: From the range calculate the width of the range (Wi), Wi = (Ui - Li + 1), where Li is the lower limit of the range and Ui is the upper limit of the range. Calculate number of bits to be embedded by the formula: $\mathrm{Ti} = \lfloor \lg(Wi) \rfloor$ for both pair.

Step 3: Extract Ti number of bits from secret message and convert it in decimal and name as Ti′. Calculate new difference using the equation (4), and name them as D′ for group 1 and D1′ for group 2 then apply adjustments.

Step 4: Adjustment type 1 (here Q is involved in the adjustment);

Step 4.1: Create Adjustment Factor (AF) and set their sign using the following rules:

For 1st pair: If P>Q and D′ < D mark adjustment factor (AF) as +ve. If P<Q and D′ > = D mark AF as +ve. If P<Q and D′ < D mark AF as -ve. If P> = Q and D′ > = D mark AF as –ve.

For 2nd pair: If Q>R and D1′<D1 mark AF as –ve. If Q<R and D1′ >D1 mark AF as –ve. If Q<R and D1′< = D1 mark AF as +ve. If Q> = R and D1′ > = D1 mark AF as +ve.

Step 4.2: Categorize AF in three types

- If AF for 1st pair and 2nd pair are +ve.
- If both are –ve.
- If they are of opposite sign, i.e. one is +ve other is –ve or vice versa.

Step 4.3: If AF is of type 1 does Q = Q+ (greater value between (absolute (D′–D) and absolute (D1′–D1))).

If AF is of type 2 do Q = Q– (greater value between (absolute (D′-D) and absolute (D1′–D))).

If AF is of type 3 do Q = Q+ (absolute (D′–D) + absolute (D1′–D1)).

Step 4.4: Calculate new difference between P, Q and R as:

$$D'' = (P-Q),\ D1'' = (Q-R).$$

If D″ = D′ and D1″ = D1′ then do D′ = D″ and D1′ = D1″, and go to Step 4.6 otherwise continue with Step 4.5

Step 4.5: Adjust P, Q and R and obtain new pixels P′, Q′ and R′ according the following cases:

Case 1: P′ = P- |D′– D″| if D′>D″ and P < = Q.
Case 2: P′ = P+ |D′– D″| if D′>D″ and P > Q
Case 3: P′ = P+ |D′– D″| if D′<D″ and P < Q
Case 4: P′ = P– |D′– D″| if D′<D″ and P > Q
Case 5: R′ = R+ |D1′– D1″| if D1′ > D1and Q < = R
Case 6: R′ = R- |D1′– D1″| if D1′ > D1 and Q > R
Case 7: R′= R- |D1′– D1″| if D1′ < D1and Q < R
Case 8: R′ = R+ |D1′– D1″| if D1′ < D1and Q > R
Q remains unchanged for all cases, i.e. Q′ = Q.
P′, Q′ and R′ are pixels after adjustment 1.

Step 4.6: Calculate change in pixels after embedding bits, CP = absolute (P–P′), CQ = absolute (Q–Q′), CR = absolute(R–R′). Compute total change in pixels TP = CP+CQ+CR.

Step 5: Adjustment type 2 (here Q remains unchanged only p and R is affected).

Step 5.1: From original pixels P, Q and R. choose 1st pair as P and Q and 2nd pair as Q and R, Obtain the difference value for 1st pair as D = |P-Q| and check the range table Ri (i = 1,2…..n) to obtain the range where D belongs to. In the same way obtain the difference value D1 = |Q-R| and check the range table Ri (i = 1, 2…..n) to obtain range D1 belongs to.

Step 5.2: From the range calculate the width of the range (Wi), Wi = (Ui - Li + 1), where Li is the lower limit of the range and Ui is the upper limit of the range. Calculate number of bits to be embedded by the formula: $Ti = \left[\lg\left(Wi\right)\right]$ for both pair.

Step 5.3: Extract Ti number of bits from secret message and convert it in decimal. Calculate new difference using the equation (4) and name them as D′ for group 1 and D1′ for group 2.

Step 5.4: For 1st pair adjust P using following cases.
Case 1: P′ = P– |D′– D| if D′>D″ and P < = Q.
Case 2: P′ = P+ |D′– D| if D′>D″ and P > Q
Case 3: P′ = P+ |D′– D| if D′<D″ and P < Q
Case 4: P′ = P– |D′– D| if D′<D″ and P > Q
For 2nd pair adjust R using following cases
Case 1: R′ = R+ |D1′– D1| if D1′ > D1and Q < = R
Case 2: R′ = R– |D1′– D1| if D1′ > D1 and Q > R
Case 3: R′ = R– |D1′– D1| if D1′ < D1and Q < R
Case 4: R′ = R+ |D1′– D1| if D1′ < D1and Q > R
Q remains unchanged for all cases, i.e. Q′ = Q.
P′, Q′ and R′ are final pixels obtained using adjustment type 2.

Step 5.5: Calculate change in pixels, CP = absolute (P-P′), CQ = absolute (Q–Q′), CR = absolute (R–R′). Compute total change in pixels TP′ = CP+CQ+CR.

Step 6: If the value of P′ falls out of range i.e. <0 or >255 then P is re adjusted using following calculation: if P′<0 P = P+ abs (P′) and go to Step 1. Otherwise if P′ >255 then P = P–(P′–255) and go to Step 1. Do the same for Q and R.

Step 7: Check for optimal change in pixels, if TP> = TP′ chose P′,Q′,R′ from adjustment type 2 otherwise chose P′,Q′,R′ from adjustment type 1 is optimal solution for embedding.

Step 8: Repeat Step 1 to Step 8 until bits available from secret file or end of cover image.

3.2 A Simple Example of the Proposed Method

Three pixels P, Q, R chosen from cover image and the values are P = 23, Q = 30 and R = 52, First pair is P and Q, difference between P and Q is D = |30-23| = 7, we have used range values of 8, 8, 16, 32, 64, 128. D falls in the range of 1st group where Li = 0, Ui = 7, Wi calculated as (Ui–Li+1) so Wi = 8 and $Ti\lfloor \lg(8) \rfloor$ = 3. As Ti = 3 three bits from secret message is chosen, in this case they are '0 1 0', decimal value of '0 1 0' is 2, so Ti′ = 2. New difference D′ obtained from D′ = Li+Ti′ = 0+2 = 2. Now consider second pair pixels Q and R, difference between Q and R is D1 = |30–52| = 22, from the same range values of 8, 8, 16, 32, 64, 128. D1 falls in the range of 3rd group where Li = 16, Ui = 31, Wi calculated as (Ui-Li+1) so Wi = 16 and $Ti = \lfloor \lg(16) \rfloor$ = 4. As Ti = 4, four bits from secret message is selected, in this case they are '1 0 1 0', decimal value of '1 0 1 0' is 10, so Ti′ = 10.The new difference D1′ is calculated as Li+Ti′ = (16+10) = 26. First adjust P, Q and R using adjustment type 1. AF for 1st pair is –ve and AF for 2nd pair is also –ve. So go for type 2 adjustment and we get P = 23, Q=25 and R = 52, as adjustment is not complete re-adjust pixels using rules in Step 4.5, and finally we get P′ = 23, Q′ = 25 and R′ = 51. Now calculate TP, which are 6 for 1st type adjustment. Figure 3 shows the process.

Now adjust P, Q and R according to adjustment type 2, in this case D > D′ and P < Q so P′ = P+ |D′– D| = 23+5 = 28. And Q′ = Q. R′ = R+ |D1′– D1| = (52+4) = 56, and Q′ = 30. TP′ = 9.

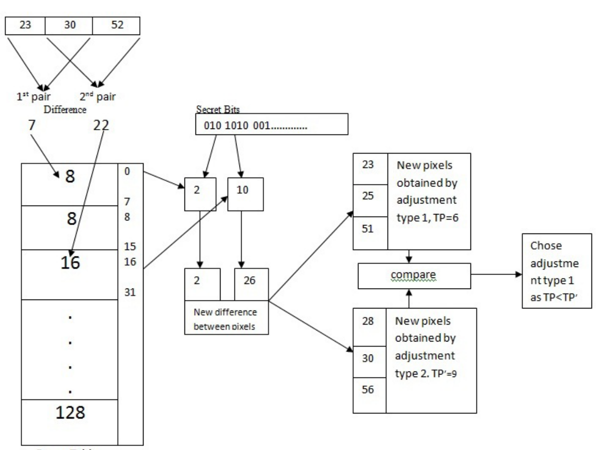

Figure 3. An example of proposed method.

Here in the example we found TP is < TP′, so type 1 adjustment is optimal so chose P′, Q′, R′ from type 1 adjustment is chosen by the program.

3.3 The Extraction Algorithm

In the proposed approach secret bits could be extracted without the original image using following steps.

Step 1: Choose blocks of three pixels from the stego image and calculate the difference between 1st pair and 2nd pair of pixels. No matters what adjustment type is applied for that block. Check the range from same range table.

Step 2: Applying equation 3 calculates number of bits to be hidden. Extract the hidden number in decimal by applying reverse rule of the embedding rules proposed in step 4.

Step 3: Convert the decimal value in Ti bits binary number. The binary number is the secret message hidden within that particular block.

4. Results

Our proposed algorithm implemented in C language. In the proposed method the stego image passes through two testing procedures. First we have compared the histograms of original image and the stego image, the histograms are almost same, this indicates that steganolytic methods could not easily find out the distortion by visual attack. Figure 4 shows detail about histogram analysis using five 512*512 images.

The statistical distortion due to data embedding in the original image is measured using Peak Signal to Noise Ratio (PSNR), which is calculated from MSE (Mean Square Error), the calculation procedure is given as follows: Let x and y arrays of size NxM, respectively representing the Y-channel frame of reference (i.e. the original copy) and Y-channel frame of the encoded/impaired copy. The MSE between the two signals is defined in equation (6) and PSNR is obtained using equation (7).

$$MSE = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left[X(i,j) - Y(i,j) \right]^2 \qquad (6)$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \qquad (7)$$

L reflects the range of values that a pixel can take: for example, if the Y channel is encoded with a depth of 8-bit, then L = 2^8 - 1 = 255. It is evident from the formula that the result is expressed in decibels. PSNR observed between original image and stego image and payload (capacity in bits) calculated using proposed approach is in Table 1. Range block used here is 8, 8, 16, 32, 64, and 128.

Our proposed scheme is compared with Wu and Tsai's PVD method[12] and the method introduced by Cheng-Hsing Yang et al.[15] using range table of 8, 8, 16, 32, 64 and 128. In order to compare we have chosen PSNR and capacity for all the three methods. Table 2 shows PSNR, payload (capacity) observed in all the three methods.

Table 1. The Capacity (payload) in Bits and PSNR Observed for 512*512 Grayscale Images Lena, Baboon, Peppers, Sailboat, Barbara, Tiffany and 480*640 Grayscale Images f–18 and f–14. Payload Observed using Random Bits

Image	PSNR	Payload (in bits)
lena	37.29	546324
baboon	31.08	657105
peppers	36.32	565101
sailboat	34.38	585969
barbara	30.67	619558
tiffany	34.50	576367
f–18	38.26	638457
f–14	38.74	620505

Table 2. PSNR and Capacity Compared between Three Methods, PSNR is in db and Capacity in Bits. Lena, Baboon, Peppers, Sailboat, Barbara and Tiffany are of Size 512*512. F18 and F–16 is of Size 480*640. Hidden Bits are from Random Bit Stream and Remain Same for Same Cover Image

Wu and Tsai's Method			Cheng-Hsing Yang's Method		Proposed Method	
PSNR		Payload	PSNR	Payload	PSNR	Payload
Lena	41.61	409019	40.47	411102	37.29	546324
Baboon	35.36	490719	33.48	520614	31.08	657105
Peppers	40.38	411615	39.86	423002	36.32	565101
Sailboat	38.18	437088	36.87	458069	34.38	585969
Barbara	34.76	461448	36.83	470586	30.67	619558
Tiffany	39.97	414402	39.05	419561	34.50	576367
f-18	42.08	464814	41.86	489563	38.26	638457
f-14	42.39	464496	41.79	480749	38.74	620505
Average	39.34	444200	38.77	459155	35.16	601173

5. Discussion and Analysis

The main objective of this paper is to increase capacity of the already proposed PVD method by finding out more edge areas. So we choose block of three pixels. The problem with choosing three pixels block is adjustment, the middle pixel takes part in adjustment of both blocks so either it is altered in a balanced way or kept unaltered, this depends on the value of the pixels. While choosing adjustment type for a particular block if we involve the middle pixel then the load will be distributed among three pixels which is a advantage for this type, on the other hand if the middle pixel does not take part in adjustment and remains unaltered, which is also good for better PSNR. That's why we choose the optimal distortion process before applying adjustment type. The optimality process check overall distortion in pixels and produces better PSNR than any of these single processes.

In the result section we have experimented on 8 images, it is observed that our proposed method finds out more edge areas than other two methods with degradation in PSNR of 4.18 units in an average while comparing with Wu and Tsai's method and 3.6 units in average while comparing with Cheng-Hsing Yang's method. This distortion is negligible in case of visual attack. While comparing capacity among these three methods, our proposed approach could store 156973 more bits i.e. 35.33% extra capacity in an average when compared with Wu and Tsai's method and 142018 more bits ie.30.93% extra than Cheng-Hsing Yang's method. The proposed method also guarantees that no value in stego image falls out of range (0-255).

6. Conclusion

In this paper we have proposed a new approach in order to increase capacity of PVD method proposed by Wu and Tsai and later modified by Cheng-Hsing Yang et al. The basic idea in our method is PVD proposed by Wu and Tsai. Our proposed method can hide much more number of bits compared to both of the methods which is shown in the result section. To ensure the quality of the stego image first we have chosen optimal adjustment technique. The histograms of both the original and stego images are compared which ensures against visual attack. Secondly to check statistical distortion we have used PSNR and that also within an acceptable range with a little degradation when compared to previous algorithms. The method also ensures that all the values are within the range of 0 to 255. Finally this proposed approach improves capacity of PVD method in a noticeable amount and produce stego image with unnoticeable distortion. This novel idea could be used to protect E-governance documents against vulnerable attack.

7. References

1. Sharp T. An implementation of key-based digital signal steganography. Proceedings Information Hiding Workshop on LNCS; Springer; 2001 Oct 24. p. 13–26.
2. Niel FJ, Zoran D, Sushil J. Information hiding, and watermarking - attacks and counter measures. Kluwer Academic Publishers; 2000.
3. Chan C-K, Cheng LM. Hiding data in images by simple LSB substitution. Pattern Recogn. 2004 Mar; 37(3):469–74.
4. Kevin C, Karen B. An evaluation of image based steganography methods. International Journal of Digital Evidence. 2003 Sep; 2(2):1–40.
5. Chang C-C, Lin M-H, Hu Y-C. A fast and secure image hiding scheme based on LSB substitution. Int J Pattern Recogn Artif Intell. 2002 Jun; 16(4):399–416.
6. Thien CC, Lin JC. A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. Pattern Recogn. 2003 Dec; 36(12):2875–81.
7. Halder T, Karforma S. A LSB-Indexed steganographic approach to secure E-governance data. Second International Conference on Computing and Systems; Burdwan, West Bengal, India: The University of Burdwan; 2013 Sep 21–22. p. 158. ISBN (10): 9-35-134273-5. ISBN (13): 978-9-35-134273-1.
8. Westfeld A, Tzmann AP. Attacks on steganographic systems - breaking the steganographic utilities Ezstego, Jsteg, Steganos, and S-tools-and some lessons learned. Proceedings of the 3rd Information Hiding Workshop on LNCS; Springer; 1999. p. 61–76.
9. Stanley CA. Pairs of values and the chi-squared attack. CiteSteer; 2005. p. 1–45.
10. Fridrich J, Goljan M, Du R. Reliable detection of LSB steganography in color and grayscale images. Proceedings of ACM Workshop on Multimedia and Security; Newyork, NY, USA. 2001. p. 27–30.
11. Ker AD. Quantitative evaluation of pairs and RS steganolysis. Proceedings of the SPIE Security, Steganography and Watermarking of Multimedia Contents; 2004 Jun 22. p. 83–97.
12. Wu DC, Tsai WH. A steganographic method for images by pixel-value differencing. Pattern Recogn Lett. 2003 Jun; 24(9–10):1613–26.
13. Wu HC, Wu NI, Tsai CS, Hwang MS. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEEE Proceedings on Vision Image and Signal Processing; 2005 Oct; 152(5):611–5.
14. Yang CH, Wang SJ, Weng CY. Analyses of pixel-value-differencing schemes with LSB replacement in steganography. The Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing; Kaohsiung, Taiwan. 2007 Nov 26–28. p. 445–8.
15. Yang C-H, Weng C-Y, Tso H-K, Wang S-J. A data hiding scheme using the varieties of pixel-value differencing in multimedia images. J Syst Software. 2011 Apr; 84(4):669–78.