IJE TRANSACTIONS B: Applications Vol. 36 No. 11, (November 2023) 2124-2136

International Journal of Engineering

Journal Homepage: www.ije.ir

Novel Scheme for Data Hiding in Binary Images using Cover Pattern Histogram

G. Chhajed*[a], B. Garg[b]

[a] *Department of Computer Engineering, BV(DU)COE, Pune, VPKBIET, Baramati, Pune, Maharashtra, India*
[b] *Department of Computer Engineering, BV(DU)COE, Pune, Maharashtra, India*

PAPER INFO

Paper history:
Received 13 April 2023
Received in revised form 22 July 2023
Accepted 23 July 2023

Keywords:
Decision Tree
Information Hiding
Encryption
Watermarking
Steganography
Histogram
Pattern Series

ABSTRACT

In today's digital age, security and safe communication are necessities. Applications frequently transport large amounts of private data as binary images. This research proposes a unique scheme that uses a cover pattern histogram-based decision tree for information concealment and extraction from binary images. This research aims to provide a data-hiding approach with a large capacity for data concealment, possible minor distortion, security, and difficulty discovering hidden data. This method uses high-frequency 3X3 pixel block patterns to obscure data. The two series of pattern's are identified based on sorted block pattern frequency. To construct a decision tree for embedding, these series patterns, key bits, and information bits work together as parameters. Information is encrypted using a secret key to ensure message security before being hidden. A decision tree decides the block suitability and bit embedding with or without flipping at the sender side. A histogram of 3X3 pixel block patterns gets generated for the received image containing concealed data, and two series are recognized similarly to the embedding procedure at the receiver side. A decision tree assesses whether an image block carries an information bit and decides whether the bit is "0" or "1". This decision tree extracts hidden data bits by analyzing series patterns and key bits. The secret key decodes retrieved concealed bits and reveal the original data. According to research, 50-80 % of hidden bits are transmitted without flipping the pixels, automatically reducing visual distortion. This scheme performs better than comparable methods and is applicable in steganography and watermarking.

doi: *10.5829/ije.2023.36.11b.16*

1. INTRODUCTION

Digital media has become a daily need due to its widespread. Images, movies, and audio are examples of digital media used for various applications. Transaction receipts, medical imaging, and scanned document images are typically binary. Significant concerns exist regarding the security and confidentiality of these images as data carriers. Cryptography is a mechanism that serves this purpose. The disadvantage of cryptography is that it is easily noticeable that some critical information exists in the carrier medium. Steganography and watermarking, which hide the existence of the information, are preferred solutions to this restriction. A combination of encryption and information hiding [1], i.e., steganography or watermarking, can improve further information security. Information hiding in the binary medium is complex due to using only two colors for painting binary images. For researchers in this discipline, maintaining visual quality with high-capacity embedding is a significant difficulty. Numerous data hiding techniques are developed to conceal data in black and white images. The techniques used edge pixels [2], boundaries [3], blocks [1-8], transformation functions [9], and run lengths [10] to locate the suitable pixels for hiding data. The steganalysis techniques confirm using the same to locate whether we can hide data at a particular location [11]. We can evaluate the performance of the data hiding technique using the peak signal-to-noise ratio (PSNR), Mean square error (MSE), and distortion matrix. We can also evaluate the data-hiding techniques with other parameters like hiding capacity and security [10]. In the current work, the hiding capacity is directly proportional to visual distortion. So there is scope to identify a better solution. Compared to the currently existing methods, using a decision tree for choosing data-hiding locations is not

*Corresponding Author Email: gjchhajed@gmail.com (G. Chhajed)

Please cite this article as: G. Chhajed, B. Garg, Novel Scheme for Data Hiding in Binary Images using Cover Pattern Histogram, ***International Journal of Engineering, Transactions B: Applications,*** Vol. 36, No. 11, (2023), 2124-2136

recommended. So this research proposed a scheme to determine whether to embed information bits in patterns or to bypass them using a decision tree. We have used the decision trees to retrieve exactly hidden information bits. We encrypted the message before hiding the information to increase the security of confidential data. The paper is organized as: section 1 introduces the proposed work, section 2 addresses related work; section 3 examines the details of the proposed scheme; section 4 discusses the details of block processing and decision tree; section 5 discusses performance analysis; and section 6 conclusion.

2. RELATED WORK

Grayscale and color images are among the main focus of information concealment efforts. There is ample scope to define new methods and design new techniques for grey and color media, as 256 shades for grey and 256 shares for each RGB color are available to maintain visual artifacts. However, binary medium faces a significant challenge due to its limited number of shades. Maintaining high information hiding capacity and less distortion is crucial for binary images. In studied literature, many techniques generally use pixels at edges, image boundaries, image blocks, and transformation functions to hide information in binary images. The extended boundary pairs with 5 pixels, and the center foreground pixel is added or deleted for information hiding [3]. The trading between a modified bit of the host image is applied, and the adjacent bit to the former's new value provides an invisible hiding effect [1]. To check the feasibility of watermark embedding in the discrete cosine transform (DCT) domain is used [5]. The image blocks are examined for smoothness and connection with neighboring pixels to select them for data hiding [6]. A blind data-hiding method preserves the connectivity of pixels in a local neighborhood [7]. A two-layer scheme proposed to authenticate and identify tempering locations [8]. A morphological transform domain is used for location identification for embedding watermarks for authentication purposes [12]. By creating an RL pair, a run-length (RL) histogram is used to improve reversible data concealing [10]. A spatial domain steganographic scheme considering HVS and statistics was proposed [13]. For precise authentication and the verification of modification, a fragile embedding developed [14]. A hybrid authentication scheme utilizes many good DCPLs for embedding and extraction [15]. An arbitrary additive distortion function is proposed, performing near the theoretical bound [2]. A framework is proposed for designing distortion functions [16]. A secret position matrix increases the hiding capacity, which minimizes distortion based on combination theory [17]. Confidential information is secured using a binary weight matrix and keys [11]. Pixels are manipulated in a certain way to hide a message to minimize distortion to achieve higher security without compromising visual artifacts [18]. A high-capacity scheme with a distortion function to check distortion uses pixels cluster and boundary connectivity to evaluate the flipping distortion [19]. The text data hiding in text documents discussed [20]. In contrast to state-of-the-art techniques, coding tables based on HVS were used to hide information in blocks of binary images [21]. Data identifying the best changeable pixel to hide data using the cover image's edge. Pixels are changed by changing the distance matrix dynamically and computing its changeable score by weighting mechanism [22]. The RDH schemes are applied for authentication and proving ownership of images captured by robots [23]. For binary images, a block classification identifies the complex sections to insert secret data [24]. Hamming codes-based data hiding scheme is proposed, which claims to flip a small number of pixels. An algorithm suggests the flipping of selected particular pixel to minimize visual distortion [25]. An RDH approach decides flipping pattern pairs with opposite center pixels (PPOCPs) in binary image [26]. Shared pixel prediction and halving compression propose a new reversible data hiding scheme in an encrypted binary image [27]. A 3x3 image block used to hide and extract groups of four bits of data using a decision tree technique [28, 29]. A diagonal partition patterns analysis of a 3X3 image block to hide data [30, 31]. Petri net modeling, which offers security and regulates user access to massive data databases to analyze the privacy issue [32].

As per the literature survey analysis, there is always a scope to identify a new approach to hide data in images, creating difficulty for steganalysis. For this, there is a requirement to design a technique that provides a new way for searching locations in the image to hide data so that confidential data will be unnoticeable. The design should also satisfy the characteristics such as imperceptibility, security to data, high data hiding capacity, and minimum visual distortion caused due to data hiding. The best way to make this possible is to utilize existing image patterns to their maximum for hiding data and increase the difficulty level of locating, extracting, and understanding the hidden data.

3. PROPOSED SYSTEM

In the proposed work, a novel approach is identified where frequently occurring block patterns in the image are used to hide data bits. The frequently occurring blocks supports the increased data hiding capacity.

The frequently occurring blocks supports the increased data hiding capacity. The novelty of the proposed method is that it uses a decision tree to decide whether the image block is suitable to carry information bits and whether the bits transmitted to be '0' or '1'. The utilization of decision tree also secures the hidden data as

the key bit pattern is used in making decision. Next, the encryption prior to hiding also secures the from detectability. This method has two phases: building a decision tree and applying it for embedding and extraction. The decision tree constructed using block patterns found in cover images. After creating and sorting the histogram of the pattern's occurrences in the input image, two pattern series, S1, and S2, are identified, thoroughly detailed in section four. The steps for creating and using a decision tree are about similar for the sender and receiver.

Let *I* represent the image of size m x n , and B represent the image pixel block of size 3x 3

I = {B00, B01,..., Bij } where i= 1 to m/3 and j=1 to n/3

Each Bij is represented by 9 bits as it corresponds to a image pixel block of size 3x3

B={ b0,.....b8}

Let *H* represent the sorted histogram of distinct block patterns *B* of the image *I* in descending order excluding the uniform block patterns, i.e., whole black or white.

H={h1,h2,….hn} } where n is number of distinct B pattern

Let S1 represent the set of sorted odd-numbered frequency block patterns in H

S1 = {B(h1),B(h3),B(h5)…}

Let S2 represent the set of sorted even-numbered frequency block patterns of H

S2= {B(h2),B(h4),B(h6)…}

Let F_S1 represent the set of frequency block patterns with the flipped bit for blocks of series represented by S1

F_S1={ B'(h1),B'(h3),B'(h5)…}

Let F_S2 represent the set of frequency block patterns with the flipped bit for blocks of series represented by S2

F_S2= {B'(h2),B'(h4),B'(h6)…}

Let B' represents the block after flipping the one bit in it where b' represents the flipped bit in B which results to B'

If flipping the first bit does not result in uniform pattern, then B' represented as

B'={ b'0,.....b8}

Else if flipping first bit results in uniform pattern, then B' represented as

B'= { b0,b'1.....b8 }

Figure 1 shows the process for building a decision tree. After creating a decision tree for a given input image, it is used to conceal information inside that image.

Figure 2 depicts the suggested workflow at the sender side. The input image divided into blocks of size 3X3 pixels. Then , block by block , the image scanned from left to right. As it scans, a 3X3 block gets represented as a 9-bit pattern. An image-based decision tree receives this block pattern at its root node as input. On the basis of a shared secret key and encrypted data bit, a decision tree provides the decision.The decision tree decides whether or not to embed information bits in the current pattern, along with whether or not the flipping of pixels is required to embed data bits.The hidden data is first encrypted with the help of a shared secret key between the sender and the receiver to increase security.

At the receiver side, the work flow shown in Figure 3. The first thing done to partition the obtained image into 3X3 block segments. The subsequent steps involve processing the block pattern histogram and identifying the series. The image is then scanned left to right, block by block, and each block examined using a decision tree to determine whether it contains concealed data and, if so, whether it is a "0" or "1". The sender-side flow explanation and the flow described here are about

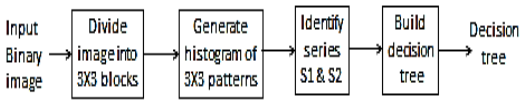

Figure 1. Decision tree building process

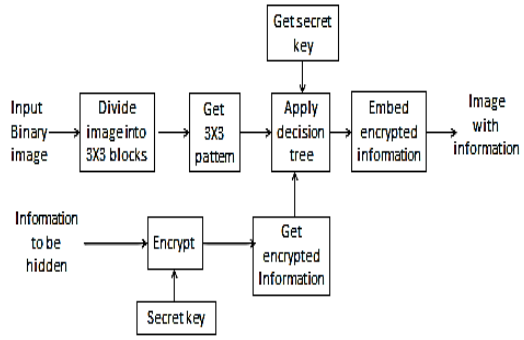

Figure 2. The process on the sender side

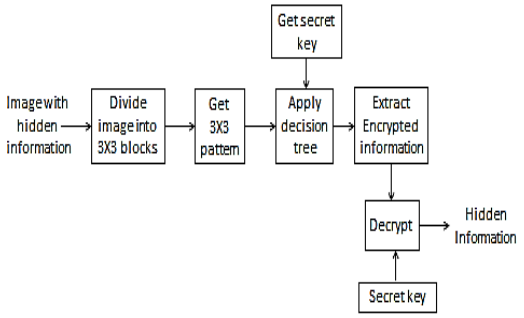

Figure 3. The process on the receiver side

TABLE 1. Sorted patterns as per their decimal values

Sr. no	Decimal value	Pattern	Pattern Frequency
1	0	000000000	748
2	1	000000001	46
3	2	000000010	34
4	4	000000100	56
5	8	000001000	41
6	32	000100000	36
7	48	000110000	23
8	64	001000000	51
9	128	010000000	27
10	219	011011011	27
11	255	011111111	36
12	256	100000000	60
13	383	101111111	31
14	438	110110110	25
15	447	110111111	44
16	479	111011111	28
17	507	111111011	42
18	509	111111101	34
19	510	111111110	45
20	511	111111111	924

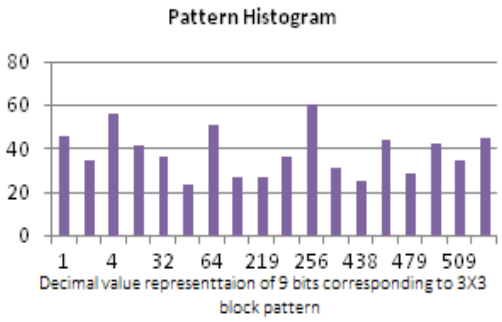

Figure 4. Histogram of top 20 patterns

similar. The decision tree only considers the current block pattern and the shraed secret key bit when determining if the current block pattern includes information and which information bits, '0', or ' 1', are embedded. Since the extracted data is encrypted, next, it is decrypted by a shared key. This information is the original hidden information sent by the sender.

4. BLOCK PROCESSING AND DECISION TREE

4. 1. Block Processing Technique The image is divided into 3X3 pixel blocks, and these blocks are processed to hide information. The following steps make up the block processing method.

Algoritm : Block Processing

1. *Get input image*
2. *Divide into 3x3 blocks and consider it as 9 bits pattern*
3. *Generate a histogram of distinct patterns*
4. *Sort the patterns in descending order according to the frequency of patterns in the image.*
5. *Ignore the uniform patterns if they are in frequent top patterns*
6. *Get corresponding patterns to the above-arranged patterns after flipping one pixel. Call it a Flipped pattern.*
7. *Sum the frequency of sorted and corresponding flipped pattern*
8. *Again arrange the patterns according to the sorted sum of frequencies in descending order.*
9. *Select a suitable number of top patterns for embedding.*

The patterns with all 9 bits 0's or 1's corresponding to a 3X3 block, i.e., all 9 pixels are black or white pixel, are called uniform pattern. The images generally have more frequency of uniform blocks patterns. However, they are not considered for data hiding because the change in a single pixel is easily noticeable in a block. The top 20 frequently occurring patterns for the test image baboon of size 200 X 200 are listed in descending order. The uniform patterns are at the top, which means they are the frequently occurring patterns in the image. The top 20 most frequent 3X3 block patterns are shown in Table 1, arranged according to their corresponding decimal values in ascending order and is depicted graphically in Figure 4.

4. 2. Series Identification The following steps explain the steps of series identification.

The outcome of applying the steps mentioned is provided in Table 2, which also includes the top 20 frequently occurring patterns with their frequency as well

Algorithm : Series Identification

1. *Get the top frequency patterns*
2. *Get the patterns after flipping one pixel of above-top frequency patterns called flipped patterns.*
3. *Get the frequency of flipped patterns.*
4. *Sum up the frequencies of patterns in step1 and step 3*
5. *Arrange the patterns in descending order of sum*
6. *Create two series called S1 and S2*
7. *S1 has patterns at odd places in ordered patterns*
8. *S2 has patterns at even places in ordered patterns*
9. *Create two more series, flipped S1 and flipped S2.*
10. *Flipped S1 has flipped patterns corresponding to patterns in S1*
11. *Flipped S2 has flipped patterns corresponding to patterns in S2*

as corresponding patterns with their initial bit flipped, i.e. changing '1' to '0' and '0' to '1'. Although uniform patterns inverted pattern's have high frequencies, they also get ignored for conveying data bits, similar to how uniform patterns are. As opposed to typical pattern's where the first bit is flipped, in such case, the second bit flipped. Pattern "10000000" has the highest frequency, as indicated in Table 2, but it is actually the inverted version of pattern "000000000". The inverted pattern "10000000" is likewise not used since uniform patterns discarded. So for this case , the second bit is flipped as "11000000". The number of information bits to be hidden dictates how many top-frequency pattern selections should be made in a series. This decision impacts the hiding capacity. The capacity to hide data would increase when more number of top-frequency patterns in a series were taken into consideration.

For the experiment, the top eight frequencies considered discarding uniform patterns. Tables 3 and 4 display the pattern analysis on the transmitter and receiver side respectively. It is observed that the sum of frequencies shown in the sum column is equal. The correct encrypted message can be decoded by adding the frequencies of the pattern and the reversed pattern at the receiver side , which is believed to be used at the sender side.

4. 3. Building Decision Tree The key component of this technique is the building of a decision tree. Two decision trees are constructed, one of which is, for embedding and the other for extraction.The sender-side decision tree uses the series pattern at the first level, a key bit at the second level, and a message bit at the third level. The tree finally decides whether the bit is to be embedded in the current testing block and as well with or without flipping.

4. 3. 1. Building Decision Tree At Sender Side The decision tree at sender side is built using following steps described using the decision table in Table 5 and the decision tree in Figure 5.

TABLE 2. Patterns in series S1 and S2

Pattern	Pattern Frequency	Flipped Pattern	Flipped Pattern Frequency	Sum	Series
100000000	60	110000000	17	77	s1
001000000	51	101000000	14	65	s2
000000100	56	100000100	6	62	s1
000100000	36	100100000	21	57	s2
000000001	46	100000001	10	56	s1
011111111	36	001111111	11	47	s2
111111110	45	011111110	2	47	s1
111111101	34	011111101	11	45	s2
110111111	44	010111111	0	44	s1
010000000	27	110000000	17	44	s2
111111011	42	011111011	0	42	s1
101111111	31	001111111	11	42	s2
111011111	28	011011111	14	42	s1
011011011	27	111011011	15	42	s2
000001000	41	100001000	0	41	s1
000000010	34	100000010	6	40	s2
110110110	25	010110110	1	26	s1
000110000	23	100110000	3	26	s2

Algorithm : Building Decision Tree At Sender Side

1. *Consider a 9-bit pattern at the root.*
2. *Patterns are categorized into Uniform patterns, S1, S2,flipped S1 , flipped S2 and other patterns at next level*
3. *Shared key bits are considered at the next level with values 0 and 1.*
4. *Information bits are considered at the next lower level and have two values, 0 and 1*
5. *The next decision level is whether to embed without flipping or after flipping.*
6. *The decision is at the leaf level, which signifies whether to discard the pattern for embedding or embed bit 0 or embed bit 1*
7. *The decision is based on a combination of the type of pattern, key bit, and Information bit.*

4. 3. 2. Building Decision Tree At Receiver Side

The receiver-side decision tree uses a series pattern at the first level and the secret key bit at the second level. This tree determines whether the current block has a hidden bit and, if so, whether it is a '0' or a '1'. The steps for building a receiver-side decision tree are explained below. The decision tree in Figure 6 and decision Table 6 are used to describe these steps.

TABLE 3. Selected series for embedding

Pattern	Pattern Frequency	Flipped Pattern	Flipped Pattern Frequency	Sum	Series
100000000	60	110000000	17	77	s1
001000000	51	101000000	14	65	s2
000000100	56	100000100	6	62	s1
000000001	46	100000001	10	56	s2
110111111	44	010111111	9	53	s1
000001000	41	100001000	7	48	s2
111111011	42	011111011	5	47	s1
111111110	45	011111110	2	47	s2

TABLE 4. Series for extraction based on sample Table 3

Pattern	Pattern Frequency	Flipped Pattern	Flipped Pattern Frequency	Sum	Series
100000000	52	110000000	25	77	s1
001000000	49	101000000	16	65	s2
000000100	47	100000100	15	62	s1
000000001	43	100000001	13	56	s2
110111111	44	010111111	9	53	s1
000001000	41	100001000	7	48	s2
111111011	42	011111011	5	47	s1
111111110	45	011111110	2	47	s2

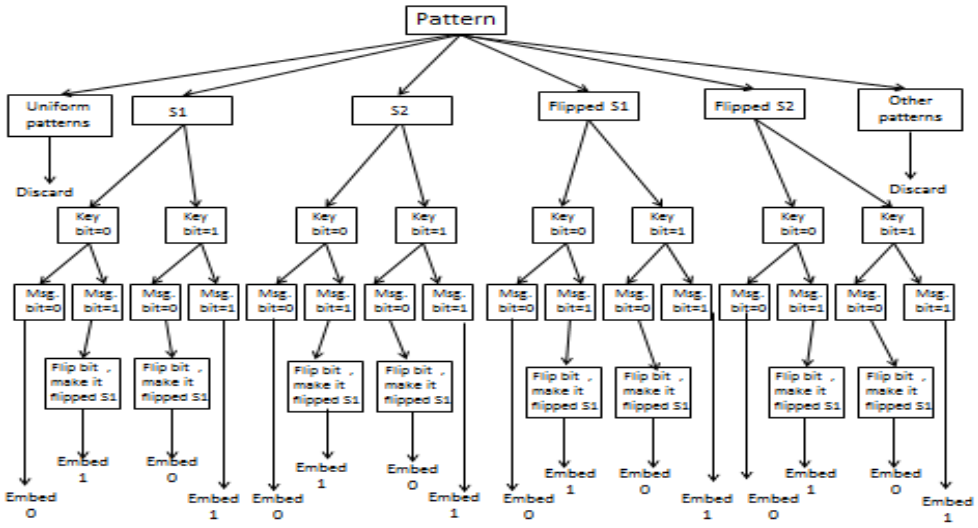

Figure 5. Decision tree for embedding at sender side

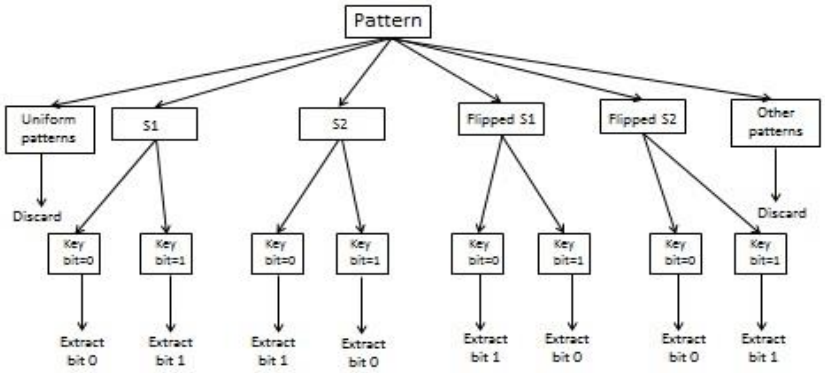

Figure 6. Decision tree for extraction at receiver side

TABLE 5. Decision table for embedding

Series	Key bit	Encrypted information bit	Decision (Embedding of encrypted information bit)
S1	1	1	Embed without flipping
S1	0	0	Embed without flipping
S1	0	1	Flip bit in S1 and make it Flipped S1
S1	1	0	Flip bit in S1 and make it Flipped S1
S2	0	1	Embed without flipping
S2	1	0	Embed without flipping
S2	0	0	Flip bit in S2 and make it Flipped S2
S2	1	1	Flip bit in S2 and make it Flipped S2
Flipped S1	1	0	Embed without flipping
Flipped S1	0	1	Embed without flipping
Flipped S1	0	0	Flip bit in S1 and make it Flipped S1
Flipped S1	1	1	Flip bit in S1 and make it Flipped S1
Flipped S2	0	0	Embed without flipping
Flipped S2	1	1	Embed without flipping
Flipped S2	0	1	Flip bit in S2 and make it Flipped S2
Flipped S2	1	0	Flip bit in S2 and make it Flipped S2

Algorithm : Building Decision Tree At Receiver Side

1. *The pattern is at the root.*
2. *Patterns are categorized into Uniform patterns, S1, S2, flipped S1, flipped S2, and other patterns*
3. *. The shared key is at the next level and has two values, 0 and 1, as it is in binary bits.*
4. *The decision is at the leaf level, which signifies whether to skip the pattern for the extraction or extract information bit 0 or 1*
5. *The decision is based on a combination of the type of pattern and key bit.*

4. 4. Embedding Using Decision Tree

Following are the steps for embedding in the binary image using a decision tree:

Algorithm : Embedding Using Decision Tree

1. *Scan the image block by block of size 3 X 3 pixels*
2. *Get the block pattern of size 9 bits*
3. *Check the category of pattern from the decision tree*
4. *Get the Key bit*
5. *Move to the branch with the combination of pattern category and key bit*
6. *Get the encrypted Information bit*
7. *Move to the branch with the combination of pattern category, key bit, and information bit*
8. *Get the decision if reaching to reach a leaf node or perform the action of flipping bit, making flipped pattern, and then get the decision reaching to a leaf node.*

Figure 7 pictorially represents the embedding of encrypted information 11010111. The sequence of

patterns belonging to series is shown cells , and the key is 11011000 in Figure 7(a). The square inside the cell represents flipping the pattern to hide data bit and making it flipped pattern is shown in Figure 7(b). In this example, S2 and S1 are converted to flipped S2 and flipped S1, respectively.

4. 5. Extraction Using Decision Tree

Following are the steps for extracting information using decision tree

Algorithm : Extraction Using Decision Tree

1. *Scan the image block by block of size 3 x 3*
2. *Get the pattern of the block of size 9 bits*
3. *Check the category of pattern from the decision tree*
4. *Get the Key bit*
5. *Move to the branch with the combination of pattern category and key bit*
6. *Get the decision reaching to leaf node whether to discard or extract 0/1 bit*

Figure 8 pictorially represents the extraction of encrypted information. The sequence of patterns in cells and the key is 11011000, is shown in Figure 8 square

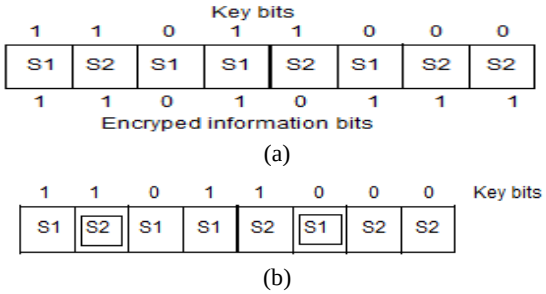

Figure 7. Embedding example (a) before embedding (b) after embedding

TABLE 6. Decision table for extraction

Series	Key bit	Decision (Extraction of encrypted information bit)
S1	1	Extract 1
S1	0	Extract 0
Flipped S1	1	Extract 0
Flipped S1	0	Extract 1
S2	0	Extract 1
S2	1	Extract 0
Flipped S2	0	Extract 0
Flipped S2	1	Extract 1

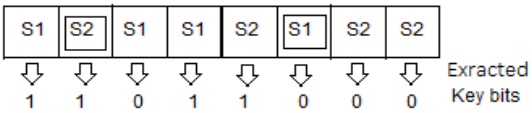

Figure 8. Extraction example

inside the cell represents flipped patterns. Applying decision tree the bit sequence 11010111 will be extracted.

5. PERFORMANCE ANALYSIS

For an experimentation, a set of 1431 binary images used. Various sizes and the variety of messages and the key combinations used to test the system. The sample set of images with the message "success" with the key "abcabca" in binary form is embedded in different images of different sizes. The sample images with some standard images are presented below with a total of 64 encrypted bits embedded in them where the first 8 bits are encrypted message length and later 56 bits represent an encrypted message. Figure 9 represents images before hiding the data. Figure 10 shows images after embedding 64-bit encrypted data using 2 patterns and 4 patterns in each series respectively. Figure 11 represents the eye part of the baboon image in original and zoomed size before and after embedding data.

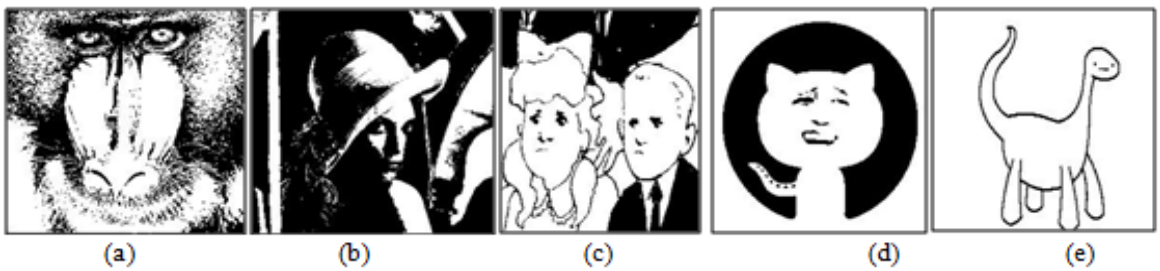

Figure 9. Original images before information hiding

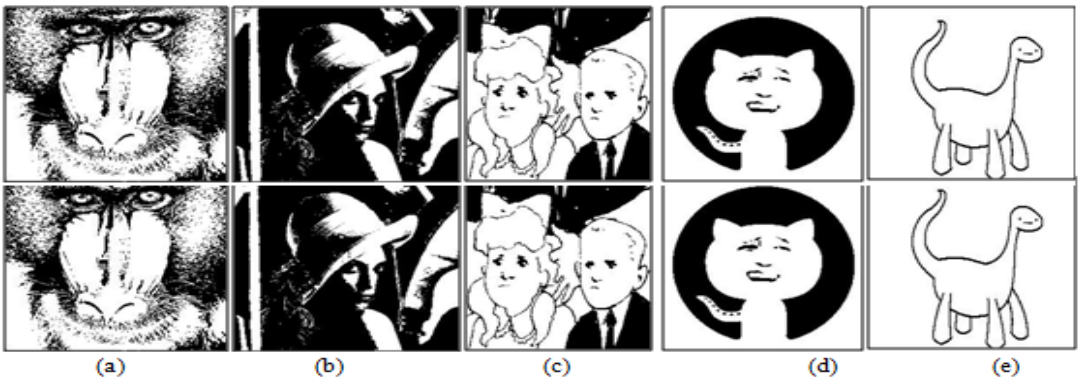

Figure 10. Images after hiding 64-bit data with series S1 and S2 having 2(top) and 4(bottom) patterns respectively a)Baboon b) Lena c) Cartoon1 d) Cat e) Dinosaur f) Hero

Figure 11. Eye part of baboon image using 4 patterns in series (a) before hiding data (b) zoomed part before hiding (c) after hiding data (b) zoomed part after hiding data

With the experimentation, it is noticed that the visual differences between the original cover images and the cover images with the hidden data are hardly detected with the naked eye. The Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are two common quality measurements to measure the difference between the cover image and the image with hidden data.

As shown in Tables 7, 8 and Figure 12, this technique performs well. For the above cases, the number of pixels flipped for series considering two patterns and four patterns are analyzed. As per observations, 50-80 % of total number of bits to be hidden is possible without changing the pixel colour. The performance of this technique is dependent on the arrangement of pixels in the cover image and key bits. If more patterns are considered in series, then there will be availability of more patterns for embedding, and distortion will reduce.

This method is compared with the method discussed in [6], where a lookup table of flip-score for 69 patterns is presented. The flip scores of the pattern decide whether the block is embeddable or not. The flip-scores greater than or equal to 0.125 were selected from literature [6] for data hiding from the range of calculated scores 0, 0.01, 0.125,0.25,0.375,0.625 considering all 512 patterns for 3X3 block. Similarly, patterns were considered for hiding data in [31]. The other schemes based on special pixel selection [13] and edge based grid pattern [32] are also compared, and it is identified that this method performs better, providing more data hiding capacity. Table 9 and Figure 13 shows the comparative analysis, where our method using 2 and 4 patterns in series S1 and S2 is compared with other methods using block based technique, edge grid based and special selection of pixel based on texture of the image. The data hiding capacity

TABLE 7. MSE and PSNR for series having 2 and 4 patterns for Figures 9 and 10

Image	Image size	Flipped pixel count (no. in bits) 2 patterns in each series	Flipped pixel count (no. in bits) 4 patterns in each series	MSE (2) in 10^{-4}	PSNR (2)	MSE (4) in 10^{-4}	PSNR (4)
Baboon	200X200	29	28	7.25	31.40	7.00	31.55
Lena	256X256	37	32	5.65	32.48	4.88	33.11
Cartoon1	257X196	30	34	5.96	32.25	6.75	31.71
Cat	225X225	27	26	5.33	32.73	5.14	32.89
Dinosaur	285X177	31	30	6.15	32.11	5.95	32.26
Hero	225X225	12	23	2.37	36.25	4.54	33.43

TABLE 8. The ratio of flipped pixel to message length for Figures 9 and 10

Image	Image size	Image DPI	Msg. size (no of bits)	Flipped pixel count (no. in bits) 2 patterns in each series	Flipped pixel count (no. in bits) 4 patterns in each series	Ratio (F/T) (2)	Ratio (F/T) (4)
Baboon	200X200	96	64	29	28	0.45	0.44
Lena	256X256	96	64	37	32	0.58	0.50
Cartoon1	257X196	96	64	30	34	0.47	0.53
Cat	225X225	96	64	27	26	0.42	0.41
Dinosaur	285X177	96	64	31	30	0.48	0.47
Hero	225X225	96	64	12	23	0.19	0.36

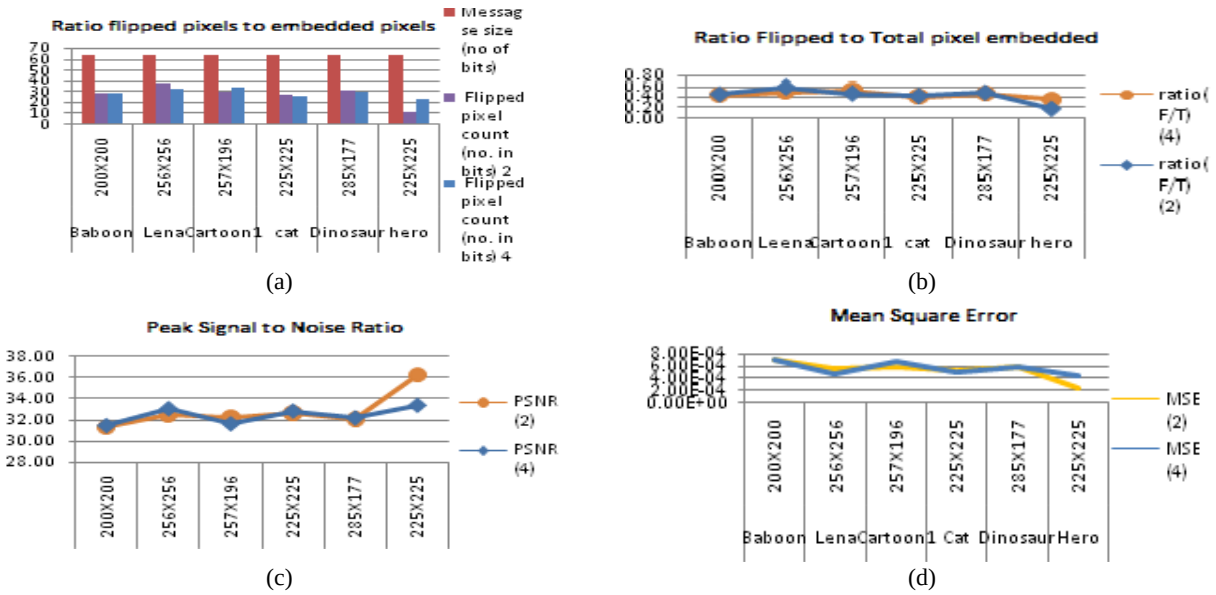

Figure 12. Performance analysis plots with 2 and 4 patterns in series S1 & S2 respectively (a) Message size Vs. flipped pixels (b) ratio of flipped pixels to no. of bits in message (c) MSE (d) PSNR

TABLE 9. Comparative Analysis

Method	Image	Image size	Flipped pixel count	Ratio (F/T)	MSE in 10^{-4}	PSNR
(patterns 2) Ours			**29**	**0.45**	**7.25**	**31.40**
(patterns 4) Ours			**28**	**0.44**	**7.00**	**31.55**
[4]			34	0.53	8.50	30.71
[9]	Baboon	200X200	33	0.52	8.25	30.84
[31]			31	0.48	7.75	31.11
[32]			35	0.55	8.75	30.58
(patterns 2) Ours			**37**	**0.58**	**5.65**	**32.48**
(patterns 4) Ours			**32**	**0.50**	**4.88**	**33.11**
[4]			37	0.58	5.65	32.48
[9]	Lena	256X256	27	0.42	4.12	33.85
[31]			30	0.47	4.58	33.39
[32]			31	0.48	4.73	33.25
(patterns 2) Ours			**30**	**0.47**	**5.96**	**32.25**
(patterns 4) Ours			**34**	**0.53**	**6.75**	**31.71**

[4]			31	0.48	6.15	32.11
[9]	Cartoon1	257X196	33	0.52	6.55	31.84
[31]			30	0.47	5.96	32.25
[32]			37	0.58	7.35	31.34
(patterns 2) Ours			**27**	**0.42**	**5.33**	**32.73**
(patterns 4) Ours			**26**	**0.41**	**5.14**	**32.89**
[4]			33	0.52	6.52	31.86
[9]	Cat	225X225	31	0.48	6.12	32.13
[31]			30	0.47	5.93	32.27
[32]			34	0.53	6.72	31.73

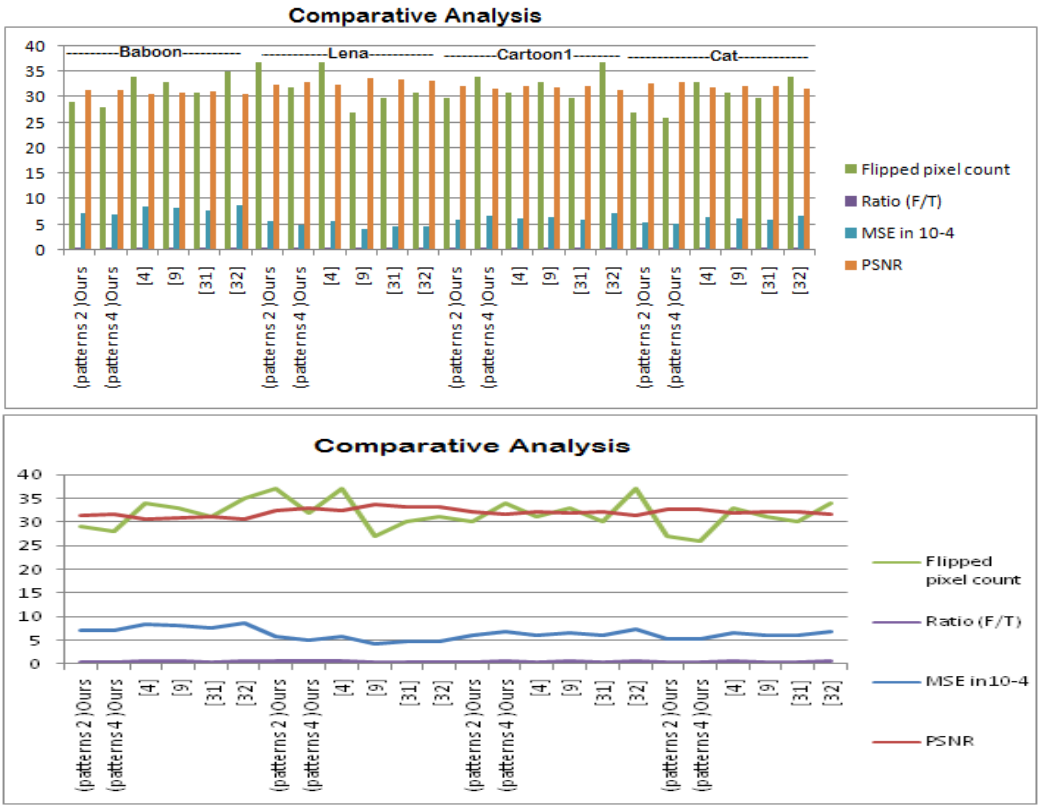

Figure 13. Comparative analysis plots with 2 and 4 patterns in series S1 & S2 respectively (a) Message size Vs. flipped pixels (b) ratio of flipped pixels to no. of bits in message (c) MSE (d) PSNR

varies depending on patterns frequency, arrangement of patterns considered in series S1 and S2. The distortion caused due to flipping pixels for data hiding depends on the sequence of series pattern occurrence in the image, secret key bit, and encrypted information bit. More the match, the less the distortion. So this technique performs well for providing good data hiding capacity for images providing more number of non-uniform blocks. For the test images, the flipped pixel count is 20-50 % of the number of bits to be hidden; hence visual distortion is also comparable to other techniques.

6. CONCLUSION

In this paper, a novel information-hiding method in the binary image is proposed. The decision tree is used to decide whether the pattern is suitable for information hiding or not, as well as which bit, 0 or 1, to be embedded. The decision tree uses three parameters for making any decision. First, it requires the series of 3X3 block patterns represented in 9 bits of the input image, the second information bit to be embedded, and third, the secret key bit. Once the decision tree is built for the input image at

the sender side, it can be applied for any information to be hidden. To enhance the security of information, it is encrypted before embedding. This method utilizes cover image patterns as much as possible and hence minimizes visual distortion. A similar decision tree is built and applied at the receiver side, which requires two parameters, the series of 3X3 block patterns represented in 9 bits of an input image with hidden information, i.e., watermarked or stego image, and the secret key bit. Decryption is applied to extract information to get the original information. This method is applicable in applications that use a binary medium for secret communication in sensitive domains like military, defense, and medical imaging. In experiments, two and four patterns are considered in series. The results show that 50-80% of pixels are utilized in their original form to hide sample data and images. From this observation, it's concluded that 50-80% of data bits are hidden without flipping pixels, which means distortion is less. It is also concluded that the need to flip pixels to hide the data bit depends on the pattern of the input image as frequencies of the 9-bit pattern are based on image 3X3 pixel block pattern occurrences second, the secret key bit sequence, and information bit sequences. Best the match less will be distortion. The hiding capacity depends on a number of patterns considered in series and their occurrences. This will prove to be an efficient information-hiding method for binary images in which the original image is not required for extracting information. Next, it will not be possible to identify the block that is carrying which bit without knowing the secret key. Next, the encryption applied to hidden data further enhances the security of hidden information.In future perspective , this method is applicable for Steganography and watermarking applications.To further reduce the distortion factor of the proposed work, pixel to be flipped for data hiding can be selected by checking the surrounding pixels in a block. If this is incorporated, than accordingly, the receiver side will also need to identify the strategy to locate the block carrying the data bit.

7. REFERENCES

1. Tseng, Y.-C. and Pan, H.-K., "Data hiding in 2-color images", ***IEEE Transactions on Computers***, Vol. 51, No. 7, (2002), 873-880. doi: 10.1109/TC.2002.1017706.
2. Filler, T., Judas, J. and Fridrich, J., "Minimizing additive distortion in steganography using syndrome-trellis codes", ***IEEE Transactions on Information Forensics and Security***, Vol. 6, No. 3, (2011), 920-935. doi: 10.1109/TIFS.2011.2134094.
3. Mei, Q.G., Wong, E.K. and Memon, N.D., "Data hiding in binary text documents", in Security and watermarking of multimedia contents III, SPIE. Vol. 4314, (2001), 369-375.
4. Chhajed, G.J., Inamdar, V. and Attar, V., "Steganography in black and white picture images", in 2008 Congress on Image and Signal Processing, IEEE. Vol. 2, (2008), 141-144.
5. Deshmukh, K.V. and Chhajed, G.J., "A steganographic method for data hiding in binary image using edge based grids", ***International Journal of Computer Applications in Technology***, Vol. 5, No. 4, (2014), 1369-1374.
6. Lu, H., Shi, X., Shi, Y.Q., Kot, A.C. and Chen, L., "Watermark embedding in dc components of dct for binary images", in 2002 IEEE Workshop on Multimedia Signal Processing., IEEE. (2002), 300-303.
7. Wu, M. and Liu, B., "Data hiding in binary image for authentication and annotation", ***IEEE Transactions on Multimedia***, Vol. 6, No. 4, (2004), 528-538. doi: 10.1109/TMM.2004.830814.
8. Yang, H. and Kot, A.C., "Pattern-based data hiding for binary image authentication by connectivity-preserving", ***IEEE Transactions on Multimedia***, Vol. 9, No. 3, (2007), 475-486. doi: 10.1109/TMM.2006.887990.
9. Yang, H., Kot, A.C. and Rahardja, S., "Orthogonal data embedding for binary images in morphological transform domain-a high-capacity approach", ***IEEE Transactions on Multimedia***, Vol. 10, No. 3, (2008), 339-351. doi: 10.1109/TMM.2008.917404.
10. Tseng, Y.-C., Chen, Y.-Y. and Pan, H.-K., "A secure data hiding scheme for binary images", ***IEEE Transactions on Communications***, Vol. 50, No. 8, (2002), 1227-1231. doi: 10.1109/ICWAPR.2007.4420668.
11. Yang, H. and Kot, A.C., "Binary image authentication with tampering localization by embedding cryptographic signature and block identifier", ***IEEE Signal Processing Letters***, Vol. 13, No. 12, (2006), 741-744. doi: 10.1109/LSP.2006.879829.
12. Xuan, G., Shi, Y.Q., Chai, P., Tong, X., Teng, J. and Li, J., "Reversible binary image data hiding by run-length histogram modification", in 2008 19th International Conference on Pattern Recognition, IEEE. (2008), 1-4.
13. Feng, B., Lu, W. and Sun, W., "Secure binary image steganography based on minimizing the distortion on the texture", ***IEEE Transactions on Information Forensics and Security***, Vol. 10, No. 2, (2014), 243-255. doi: 10.1109/TIFS.2014.2368364
14. Cao, H. and Kot, A.C., "On establishing edge adaptive grid for bilevel image data hiding", ***IEEE Transactions on Information Forensics and Security***, Vol. 8, No. 9, (2013), 1508-1518. doi: 10.1109/TIFS.2013.2274041.
15. Guo, M. and Zhang, H., "High capacity data hiding for binary image authentication", in 2010 20th International Conference on Pattern Recognition, IEEE. (2010), 1441-1444.
16. Pevný, T., Filler, T. and Bas, P., "Using high-dimensional image models to perform highly undetectable steganography", in Information Hiding: 12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12, Springer. (2010), 161-177.
17. Wu, N.-I. and Hwang, M.-S., "Development of a data hiding scheme based on combination theory for lowering the visual noise in binary images", ***Displays***, Vol. 49, (2017), 116-123. doi. https://doi.org/10.1016/j.displa.2017.07.009
18. Mol, M.H. and Reji, P., "A secure binary data hiding and comparison technique", ***International Journal of Scientific Engineering and Research***, Vol. 7, No. 7, (2016), 810.
19. Feng, B., Lu, W. and Sun, W., "High capacity data hiding scheme for binary images based on minimizing flipping distortion", in Digital-Forensics and Watermarking: 12th International Workshop, IWDW 2013, Auckland, New Zealand, October 1-4, 2013. Revised Selected Papers 12, Springer. (2014), 514-528.
20. Bhattacharyya, D., Haveliya, A. and Kim, T.-h., "Secure data hiding in binary text document for authentication", ***Applied Mathematics***, Vol. 8, No. 1L, (2014), 371-378. doi: 10.12785/amis/081L47.
21. Ding, W. and Wang, Y., "Data hiding in binary image with high payload", ***Arabian Journal for Science and Engineering***, Vol.

43, No., (2018), 7737-7745. doi. https://doi.org/10.1007/s13369-018-3130-5

22. Liu, T.-H. and Chang, L.-W., "An adaptive data hiding technique for binary images", in Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., IEEE. Vol. 4, (2004), 831-833.
23. Manikandan, V. and Masilamani, V., "Reversible data hiding scheme during encryption using machine learning", ***Procedia Computer Science***, Vol. 133, (2018), 348-356. doi. https://doi.org/10.1016/j.procs.2018.07.043
24. Nguyen, T.-S., Chang, C.-C. and Hsueh, H.-S., "High capacity data hiding for binary image based on block classification", ***Multimedia Tools and Applications***, Vol. 75, (2016), 8513-8526. doi. https://doi.org/10.1007/s11042-015-2768-1
25. Lee, Y., Kim, H. and Park, Y., "A new data hiding scheme for binary image authentication with small image distortion", ***Information Sciences***, Vol. 179, No. 22, (2009), 3866-3884. doi. https://doi.org/10.1016/j.ins.2009.07.014
26. Yin, X., Lu, W., Zhang, J., Chen, J. and Liu, W., "Reversible data hiding in binary images by flipping pattern pair with opposite center pixel", ***Journal of Visual Communication and Image Representation***, Vol. 70, (2020), 102816. doi. https://doi.org/10.1016/j.jvcir.2020.102816
27. Li, F., Zhang, L. and Wei, W., "Reversible data hiding in encrypted binary image with shared pixel prediction and halving compression", ***EURASIP Journal on Image and Video Processing***, Vol. 2020, (2020), 1-21. doi. https://doi.org/10.1186/s13640-020-00522-6
28. Chhajed, G.J. and Garg, B.R., Applying decision tree for hiding data in binary images for secure and secret information flow, in Cybersecurity measures for e-government frameworks. 2022, IGI Global.175-186.
29. Chhajed, G. and Garg, B., "Data hiding in binary images for secret and secure communication using decision tree", in 4th EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing: BDCC 2021, Springer. (2022), 69-84.
30. Chhajed, G. and Garg, B., "Information security by hiding data in binary images based on block-diagonal partition pattern", in 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS), IEEE. (2022), 1-6.
31. J Chhajed, G. and Garg, B.R., "Hiding data in binary images using block-diagonal partition pattern", ***International Journal of Computing and Digital Systems***, (2023). doi. http://dx.doi.org/10.12785/ijcds/130176
32. Asadi Saeed Abad, F. and Hamidi, H., "An architecture for security and protection of big data", ***International Journal of Engineering, Transactions A: Basics***, Vol. 30, No. 10, (2017), 1479-1486.

COPYRIGHTS

©2023 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.

Persian Abstract

چکیده

در عصر دیجیتال امروز، امنیت و ارتباطات ایمن از ضروریات است. برنامه ها اغلب مقادیر زیادی از داده های خصوصی را به عنوان تصاویر باینری منتقل می کنند. این تحقیق یک طرح منحصر به فرد را پیشنهاد می کند که از درخت تصمیم گیری مبتنی بر هیستوگرام الگوی پوشش برای پنهان کردن اطلاعات و استخراج از تصاویر باینری استفاده می کند. هدف این تحقیق ارائه یک رویکرد پنهان کردن داده با ظرفیت بزرگ برای پنهان کردن داده ها، اعوجاج جزئی احتمالی، امنیت و مشکل در کشف داده های پنهان است. این روش از الگوهای بلوک پیکسلی 3×3 با فرکانس بالا برای مبهم کردن داده ها استفاده می کند. دو سری از الگوها بر اساس فرکانس الگوی بلوک مرتب شده شناسایی می شوند. برای ساختن یک درخت تصمیم برای جاسازی، این الگوهای سری، بیت های کلیدی و بیت های اطلاعاتی با هم به عنوان پارامتر کار می کنند. اطلاعات با استفاده از یک کلید مخفی برای اطمینان از امنیت پیام قبل از پنهان شدن رمزگذاری می شوند. درخت تصمیم مناسب بودن بلوک و تعبیه بیت را با یا بدون چرخش در سمت فرستنده تعیین می کند. یک هیستوگرام از الگوهای بلوک پیکسل X33 برای تصویر دریافتی حاوی داده های پنهان تولید می شود و دو سری مشابه با روش جاسازی در سمت گیرنده شناسایی می شوند. درخت تصمیم ارزیابی می کند که آیا یک بلوک تصویر حاوی بیت اطلاعات است و تصمیم می گیرد که آیا بیت "0" یا "1" باشد. این درخت تصمیم، بیت های داده پنهان را با تجزیه و تحلیل الگوهای سری و بیت های کلید استخراج می کند. کلید مخفی بیت های پنهان بازیابی شده را رمزگشایی می کند و داده های اصلی را آشکار می کند. بر اساس تحقیقات، 50 تا 80 درصد از بیت های پنهان بدون برگرداندن پیکسل ها منتقل می شوند و به طور خودکار اعوجاج بصری را کاهش می دهند. این طرح نسبت به روش های مشابه بهتر عمل می کند و در استگانوگرافی و واترمارکینگ قابل استفاده است.