Digital Watermarking on Camera Captured Color Images

Engineering

KEYWORDS : Steganography, Fragile watermarking, hashing function (hash code).

Ms. Ankita P. Deshmukh Department of Electronics and Telecommunication Shram Sadhna Bombay Trust College Of Engineering and Technology ,Jalgaon , India

Dr. S.R.Suralkar Department of Electronics and Telecommunication Shram Sadhna Bombay Trust College Of Engineering and Technology ,Jalgaon , India

ABSTRACT

Digital watermarking provides secured communication. Due to vast expansion of Internet, digital data such as audio, images and videos has been used on a large scale for communication. To protect these digital media it is essential to use digital watermarking for authentication, copyrights and security purposes. This paper describes the detail concept of digital watermarking and the main contribution in the field of steganography. In image Steganography, a secret communication is achieved to hide a message into cover image (the original image where the message is been incorporated) and generate a stenographic-image (image with the incorporated message).

I. Introduction

A digital watermarking is a bit pattern ,inserted into a digital image files that identifies the file's copyright information (author, rights, etc). Digital Watermarking is done by embedding information in digital data, such that it cannot be detected without special software without the confirmation that the embedded data is present in all copies of the data that are made whether legally or otherwise, regardless of attempts to damage/ remove it. Digital watermarking has been used to authenticate images and overcome the problems related with the copyright protection. There are two types of watermarking systems; robust and fragile. Robust watermarks are used to keep in check the illegal copying and is made for the copyright protection. The fragile watermarks is used to detect every possible tampering in the watermarking of the digital media. The main purpose of watermarking is to hide a message in some image, to get new data image. Digital watermarking focuses mainly on the protection of rights and the authentication of digital media. Similar to stenographic methods, digital watermarking methods hide information in digital media. Steganography is the art of communication such that the presence of a message is not detected. The main purpose of steganography is to hide message in some image, to get new data image in such a way that an unauthorized user cannot detect the presence of message in new data image

II. Objectives

The main objective of Fragile watermarking[1] and hash coding concept used in the paper is to allow user for a secured communication using digital multi media. It allows user to hide a secret massage which is retrieve by the user at the receiver end. A Fragile watermarking technique is used which helps in detection of tempering. Fragile watermarking is very sensitive to tempering which does not allow the access of secret message even if the image is been slightly tempered. Hash function is used to check if the original file and the received file is identical or different.

III. Concept

Fig.1 shows a simple concept of embedding a message into image using steganography. Firstly the image which is to be embedded with the secret message, say the original image in convert into bit stream. Then the message which is to be encoded is converted into a bit stream. This message bit streams is then embedded in the original image to form a data image say the steganography image. This is identical to the original image. The difference in the original image and the stego-image can not be detected with the necked eyes. The detection of the message is the vice-versa process.

Now for the authentication, fragile watermarking is used. Fragile watermarking is destroyed when tempered. Considering this property, fragile watermarking can detect a slight tempering in the original image. This shows that some kind of hacking is tried on the image. And if the image is hacked the secret message cannot be retrieved.

Hashing algorithm is used to execute this concept. Hash function is applied on the block of image say the original image and if there is even a small change in the image the hash function will changed the code. This code is different from that of the code generated from the original cover image showing that the image is been tempered and the receiver can re-request for the image.

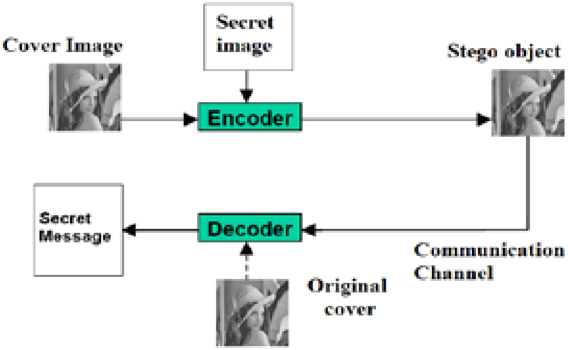


Figure 1: General block diagram of message encoding and decoding

IV. Algorithms:

In this paper, the fragile watermarking algorithm and hashing function is executed to effectively detect the tempering and threats on images.[1]

The Fragile watermarking algorithm:

The proposed fragile watermarking algorithm for hiding data message in cover image is given below:

C : Is a cover image
R : Is the no.of bits to be embedded in one block
H : No.of bytes in header
S : Size of cover image in bytes
D :Size of encoded message in bytes

inputOutputMarker : Contains the pointer location in encrypted file .

Following are the steps to hide data in image:

Step 1 :Find out cover image extension . If it is jpg image then header information will be available in first 3 bytes (0 – 2)

.If it is png file then header information will be available in first 42 bytes (0..41) and for other files ,say ,gif ,the header information will be available in first 32 bytes .

Step 2 : Read cover image and convert it to binary (byte [])

Step 3 : Create a new blank encrypted image or it could be an existing image that will be overridden .

Step 4 : Write out the following information in output file .

Header of the cover image C ,which is H bytes

Compute size of cover image S , and write it to output file S is 32 bit ,written as MSB3 , MSB2,MSB1,MSB0 .Each MSB is of 8bits each .

Write S – inputOutputMarker no.of bytes from cover image to encrypted image .

Write version of the algorithm used for creating encrypted file .Say 2.0.0 is written in 8-bit binary form.

Write 8 bit feature value .If extension is not null and compression is done then feature value is 7 else it is 6.

The encrypted file will be compressed with compression ration of 25 .Write 8bit compression ratio

Write 32-bit message size D from MSB to LSB .

Finally write message to be hidden as a byte [ ].

Retrieval of confidential data from encrypted image :

Step 1 : Read encrypted image and first compute its hash code using MD5 algorithm . If it is similar to original image ,then only data could be extracted .

Step 2 : Now read the encrypted image in binary form. Let masterByteArray contains the data of encrypted image .

Step 3 : Read header info and set inputOutputMarker accordingly.

Step 4 :Read image size from encrypted image .To compute its decimal value do the following operation :

Read first 8-bits of image size &(logical AND) 3 and shift it to left by 24 bits.

ii) Read next 8-bits of image size &(logical AND) 3 and shift it to left by 16 bits.

iii) Read next 8-bits of image size &(logical AND) 3 and shift it to left by 8 bits.

iv) Read next 8-bits of image size &(logical AND) 3 and shift it to left by 0 bits.

Step 5 : Read S-inputOutputMarker no.of bytes from encrypted file

Step 6 : Read 24bits of version info.

Step 7 : Read 8-bit feature value to see if the file is compressed.

Step 8: Read compression ratio .

Step 9 : Read size of encoded message .Create an empty array of D size.

Step 10. Read encoded message and store it in a byte[] and then convert it to character.

Hash code generation using MD5 algorithm :

In order to generate Hash codes, the encoded image is first converted into binary form that is (byte[]). The computation of the 32 character is done using MD5 algorithm. MD5 hash code is generated using J2ME SATSA-CRYPTO (JSR-177) APIS. The code uses MessageDigest class and its method update and digest to generate hash code. Then a user defined method is used to convert the binary message digest to its hexadecimal equivalent. This procedure provides us with a 32 character unique code which is hash code for respective images. This code is different for every image , thus if there is a slight change in image the code changes.

V. Results:

The results are based on three types of tempering and they are:

- Erasing/deleting a part of image.
- Image cropping.
- Color changing.

Table .1. Result for erasing/deleting image

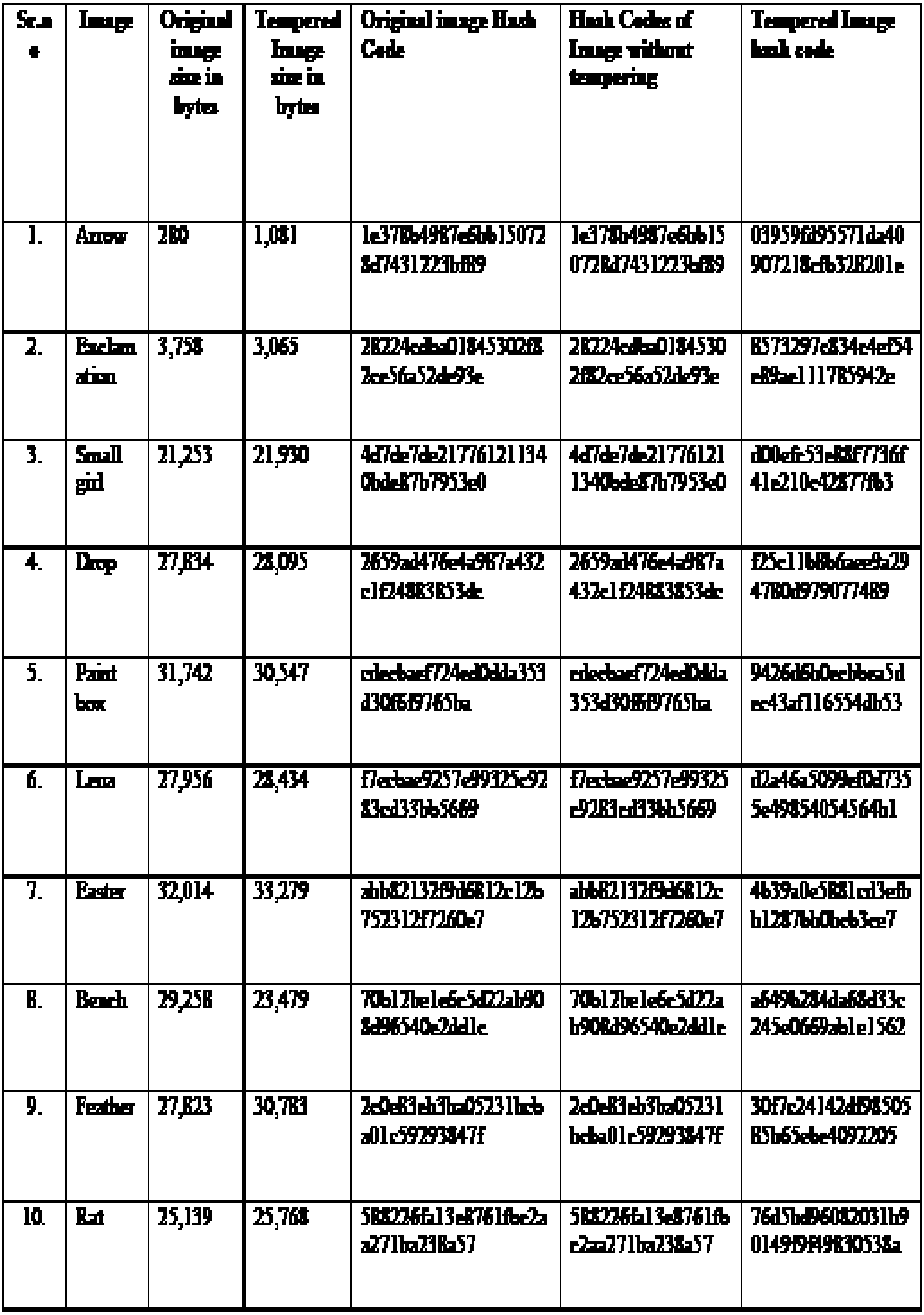
Sr.no	ImagIIIma Image	Original Image size in bytes	Tempered Size in bytes	Original image Hash code	Hash Codes of Image without tempering	Tempered Image hash code
1.	Arrow	280	1,039	1e378b4987e6bb150728d7431223b589	1e378b4987e6bb150728d7431223b589	7e286357a4b81161c337963b452492b2
2.	Exclamation	3,758	2,849	28bb4cdba01845302f82ce56d52de93c	28bb4cdba01845302f82ce56d52de93c	10e57ef53f361973def241b9e51cdcef
3.	Small girl	21,253	20,260	4d7de7de217761211340bde87b7953e0	4d7de7de217761211340bde87b7953e0	10e57ef53f361973def241b9e51cdcef
4.	Drop	27,834	27,720	2659ad476e4a987a432c1f24883853dc	2659ad476e4a987a432c1f24883853dc	297150e4239e285665ef666f8f117850
5.	Paint box	31,742	29,105	cdecbaef724ed0dda353d30f6f9765ba	cdecbaef724ed0dda353d30f6f9765ba	a440ef48bfb071a35aa9fcef655ec87b
6.	Lena	27,956	27,872	f7ecbae9257e99325c9283cd33bb5669	f7ecbae9257e99325c9283cd33bb5669	914aad273beb6fa87529216f1daae9f
7.	Easter	32,007	32,309	abb82132f986812c12b752312f7260e7	abb82132f986812c12b752312f7260e7	9be1809ed8b6a3b3db6223bed7c66c6c
8.	Bench	29,258	32,784	70b12be1e6c5d22ab908d96540e2dd1c	70b12be1e6c5d22ab908d96540e2dd1c	945bac35b348a72283a108e346e64827
9.	feather	27,823	32,239	2c0e83eb3ba05231bcba01c59293847f	2c0e83eb3ba05231bcba01c59293847f	247dcf0240c52e79d123cd02fa1bba80
10.	Rat	25,139	26,569	588226fa13e8761fbc2aa271ba238a57	588226fa13e8761fbc2aa271ba238a57	Ae5cdf40e2c3f377933753d0abe25b1d

Volume : 4 | Issue : 8 | Aug 2015 • ISSN No 2277 - 8179

Table .2. Results for Image Cropping

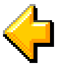
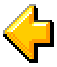
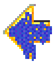
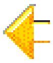
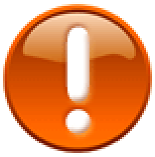
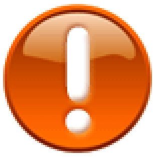
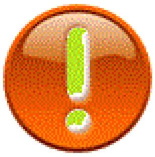
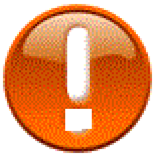
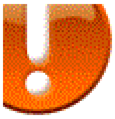
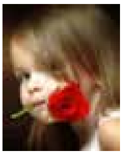
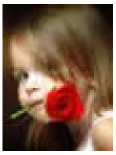
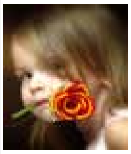
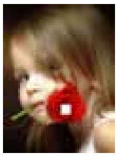
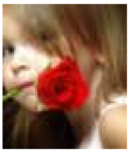
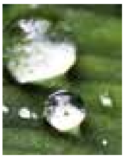
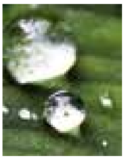
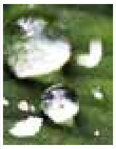
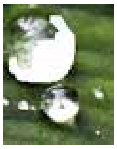
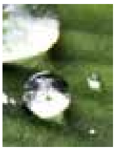
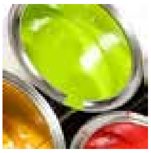
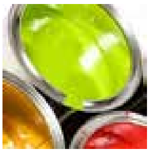
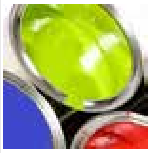
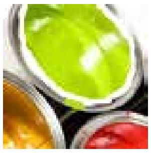
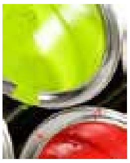
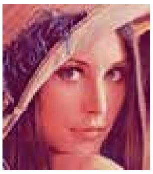
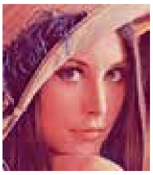
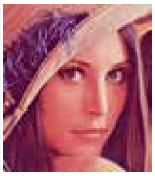
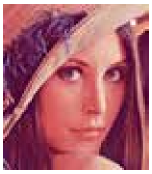
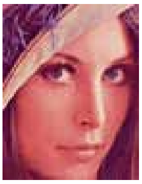
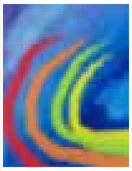
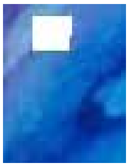
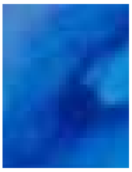
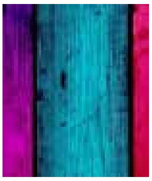
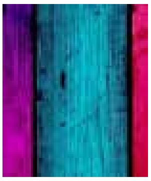
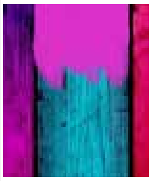
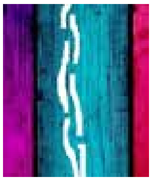
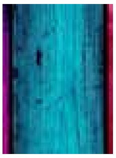
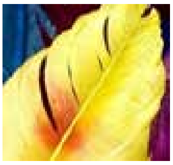
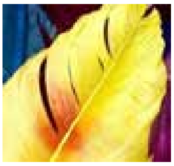
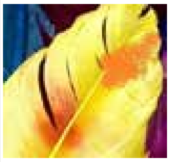
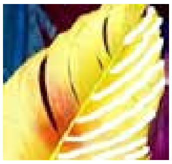
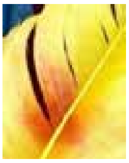
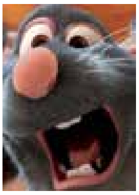
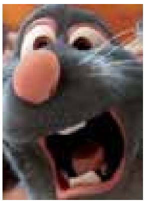
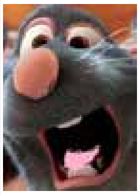
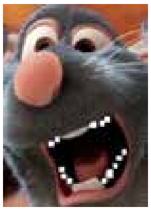
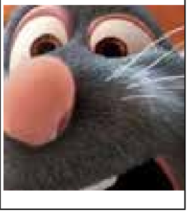
Sr.	Images	Original Size in bytes	Tempered Size in bytes	Original image Hash Code	Hash Codes of Image without tempering	Tempered Image hash code
1.	Arrow	280	1,039	1e378b4987e6bb150728d7431223b589	1e378b4987e6bb150728d7431223b589	e2af6480a50da00c775d600a66784283
2.	Exclamation	3,758	2,849	28bb4cdba01845302f82eef6d52de93c	28bb4cdba01845302f82eef6d52de93c	f11589feb4520549f72b7dd22b66e0f
3.	Small girl	21,253	20,260	4d7de7de217761211340bde87b7953e0	4d7de7de217761211340bde87b7953e0	825be97d6bd8670dd01bac2e2d908bae
4.	Drop	27,834	27,720	2659ad476e4a987a432c1f24883853de	2659ad476e4a987a432c1f24883853de	8d19ac9bd68a3ac9860ada2eaef6de06
5.	Paint box	31,742	29,105	cdecbaef724ed0dda353d30f6f976fba	cdecbaef724ed0dda353d30f6f976fba	5c03b63c00b400edceae4588f322869
6.	Lena	27,956	15,518	f7ecbae9257e99325c9283ed33bb5669	f7ecbae9257e99325c9283ed33bb5669	6b1ac566de01e6b87621227b9ea62204
7.	Easter	32,014	32,309	abb82132f9d6182c12b752312f7260e7	abb82132f9d6182c12b752312f7260e7	89f52c038443736acf48d703cbfcb34a
8.	Bench	29,258	32,784	70b12be1e6c5d22ab908d96540e2dd1c	70b12be1e6c5d22ab908d96540e2dd1c	fae3c2b6b67aeeb9b22cb2dc6bd851d3
9.	Feather	27,823	32,239	2c0e83eb3ba05231bcba01c592938475	2c0e83eb3ba05231bcba01c592938475	42e65b90411c6d37ae4d71ba67c33357
10.	Rat	25,139	26,569	588226fa13e8761fbc2aa271ba238a57	588226fa13e8761fbc2aa271ba238a57	47f161a82a3ba2874c070b2243717d5f

Table 3. Color change in images



Sr.no	Image	Original image size in bytes	Tempered Image size in bytes	Original image Hash Code	Hash Codes of Image without tempering	Tempered Image hash code
1.	Arrow	280	1,081	1e378b4987e6bb150728d7431223bf89	1e378b4987e6bb150728d7431223bf89	03959fd95571da409072l8cfb328201e
2.	Exclamation	3,758	3,065	28224cdba01845302f82ce56a52de93e	28224cdba01845302f82ce56a52de93e	8573297c834e4ef54e89ae1117859 42e
3.	Small girl	21,253	21,930	4d7de7de217761211340bde87b7953e0	4d7de7de217761211340bde87b7953e0	d00efe53e88f7736f41e210e42877fb3
4.	Drop	27,834	28,095	2659ad476e4a987a432c1f24883853dc	2659ad476e4a987a432c1f24883853dc	f25c11b8b6aee9a294780d979077489
5.	Paint box	31,742	30,547	cdecbaef724ed0dda353d30f6f9765ba	cdecbaef724ed0dda353d30f6f9765ba	9426d6b0ecbbea5dec43af116554db53
6.	Lena	27,956	28,434	f7ecbae9257e99325e9283cd33bb5669	f7ecbae9257e99325e9283cd33bb5669	d2a46a5099ef0d7355e49854054564b1
7.	Easter	32,014	33,279	abb82132f9d6812c12b752312f7260e7	abb82132f9d6812c12b752312f7260e7	4b39a0e5881cd3efbb1287bb0bcb3ce7
8.	Beach	29,258	23,479	70b12be1e6e5d22ab908d96540e2dd1c	70b12be1e6e5d22ab908d96540e2dd1c	a649b284da68d33c245e0669ab1e1562
9.	Feather	27,823	30,783	2c0e83eb3ba05231bcba01c59293847f	2c0e83eb3ba05231bcba01c59293847f	30f7c24142df9850585b65ebe4092205
10.	Rat	25,139	25,768	588226fa13e8761fbe2aa271ba238a57	588226fa13e8761fbe2aa271ba238a57	76d5bd96082031b90149f9f49830538a

Table 3. Studied Images

Sr.no	Image name	Image	Stego image	Color change	Erasing/deleting	Cropping
1.	Arrow					
2.	Exclamation					
3.	Small girl					
4.	Drop					
5.	Paint box					
6.	Lena					
7.	Easter					
8.	Bench					
9.	Feather					
10.	Rat					

VI. Conclusion:

The paper proposes Fragile watermarking algorithm for hiding data in the cover image and the hash code generation method by using MD5. For proposed methods, it shows that the tempering in the image can be detected with the help of hash code and the hidden message cannot be retrieved if the image is tempered. The change in hash code shows that the image is been tempered.

REFERENCE

1. Taha. Jassim and Raed Abd-Alhameed, Hussain Al-Ahmad," New Robust and Fragile Watermarking Scheme for Color Images Captured by Mobile Phone Cameras",2013 UKSim 15th International Conference on Computer Modelling and Simulation. | 2. Mehdi Hussain and Mureed Hussain," A Survey of Image Steganography Techniques" | International Journal of Advanced Science and Technology Vol. 54, May, 2013 | 3. Y. K. Jain and R. R. Ahirwal, "A Novel Image Steganography Method With Adaptive Number of Least Significant Bits Modification Based on Private Stego-Keys", International Journal of Computer Science and Security (IJCSS), vol. 4, (2010) March 1. | 4. H. Yang, X. Sun and G. Sun, "A High-Capacity Image Data Hiding Scheme Using Adaptive LSB Substitution", Journal: Radioengineering, vol. 18, no. 4, (2009), pp. 509-516. | 5. S. Channalli and A. Jadhav, "Steganography an Art of Hiding Data", International Journal on Computer Science and Engineering, IJCSE, vol. 1, no. 3, (2009). | 6. C.-H. Yang, C.-Y. Weng, S.-J. Wang, Member, IEEE and H.-M. Sun, "Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems", IEEE Transactions on Information Forensics and Security, vol. 3, no. (2008) September 3, pp. 488-497. | 7. K.-H. Jung, K.-J. Ha and K.-Y. Yoo, "Image data hiding method based on multi-pixel differencing and LSB substitution methods", Proc. 2008 International Conference on Convergence and Hybrid Information Technology (ICHIT '08), Daejeon (Korea), (2008) August 28-30, pp. 355-358. | 8. H. Zhang, G. Geng and C. Xiong, "Image Steganography Using Pixel-Value Differencing", Electronic Commerce and Security, ISECS '09. Second International Symposium on (2009) May. | 9. W. J. Chen, C. C. Chang and T. H. N. Le, "High Payload Steganography Mechanism Using Hybrid Edge Detector", Expert Systems with Applications (ESWA 2010), vol. 37, pp. 3292-3301, (2010) April 4. | 10. V. Madhu Viswanatham and J. Manikonda, "A Novel Technique for Embedding Data in Spatial Domain", International Journal on Computer Science and Engineering, IJCSE, vol. 2, (2010). | 11. M. A. Al-Husainy, "Image Steganography by Mapping Pixels to Letters", Journal of Computer Science, vol. 5, no. 1, (2009), pp. 33-38. | 12. H. Motameni, M. Norouzi, M. Jahandar and A. Hatami, "Labeling Method in Steganography", World Academy of Science, Engineering and Technology, France, (2007). | 13. B. Ahuja, M. Kaur and M. Rachna, "High Capacity Filter Based Steganography", International Journal of Recent Trends in Engineering, vol. 1, no. 1, (2009) May. | 14. M. Tanvir Parvez and A. Abdul-Aziz Gutub, "RGB Intensity Based Variable-Bits Image Steganography", IEEE Asia-Pacific Services Computing Conference, (2008), pp. 1322-1327. | 15. A. M. Hamid and M. L. M. Kiah, "Novel Approach for High Secure and High Rate Data Hidden in the Image Using Image Texture Analysis", International Journal of Engineering and Technology (IJET): 0975-4042, (2009). |