International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 01 | Jan-2016 www.irjet.net p-ISSN: 2395-0072

Information hiding based on optimization technique for Encrypted Images

Durfi Ashraf[1], Sayiema Amin[2]

[1,2] *P.G. student, Department of Electronics & Communication Engineering,*

Adesh Institute of Engineering & Technology, Punjab, India

Abstract- ***With the development of latest technologies in communication and computer networks, the importance and value of exchanged data over the internet or other media types are increasing. One of the fundamental issues with sending information over the web is the 'security threat' it postures which means that the confidential or personal information can be hacked or stolen in various perspectives. Text fusion in images is an important technology for image processing. We have lots of information related to the reports and need lots of space to store and the proper position and name which relates that image with that data. Reversible data hiding is a method to embed extra message into some unsatisfactory spread media, for example, medical or military images with a reversible way so that the original content can be accurately restored. Generally data hiding is utilized for communication which is secret. In this paper, a technique is proposed for reversible data hiding in encrypted images. An algorithm is proposed which will first find out the area of interest and after that noisy pixel. BAT algorithm is used to find the coordinates of the noisy pixels and will embed text data on it and after that rest of the data will be fed into the border area of the images. The research work also provides the knowledge about the ability of the algorithm to secure and protect data against attacks and its speed and efficiency in doing so.***

Keywords — Graphical user Interface, Performance analysis, Image encryption, reversible data hiding, separable reversible data hiding, AOI, NAOI, BAT, image recovery, Computation time.

1. INTRODUCTION

Presently in these days, applications of computer have diffused into each sphere of life for control of numerous sophisticated applications. Lot of these applications is of safety critical, substantial, and complex. Subsequently, there must be reliable software. In other words, software of good quality and also high reliability is crucial. Aside from presence of numerous techniques for enhanced reliability, software testing is very common and important followed methodology. Thus, testing remains the most vital part of quality assurance in the practice of the development of software. Even though there are so many techniques are available for quality assurance such as design reviews, formal specifications, inspection and checking of model exists till today. Since the software now has now an imperative part in our lives both socially and economically. There is pressure for professionals of software to concentrate on quality issues. The software with poor quality that can bring loss of property or life is no more adequate to society. [1]

Software testing is very expensive and labor-intensive; it represents pretty nearly 50% of the cost of development of software system. If the process of testing could be automated, the cost of software developing ought to be decreased fundamentally. Of the problems included in software testing, one is of specific importance here; the issue of creating test data. Generation of test data in testing of software is the procedure of distinguishing program input data which fulfill criterion of selected testing. A test data generator is a device which helps a programmer in generation of test data for a program. There are three types of generators of test data: path wise test generators, data specification generators, and random test generators. [2]

Software testing is an imperative activity in lifecycle of software development. Organizations of software spend significant part of their budget in activities which are related to testing. Testing incorporates executing a program on a set of test cases and contrasting the actual results with the results that are expected. Testing ought to concentrate on prevention of fault. Test cases are normally derived for some artifacts of software like specifications, design or may be implementation of the system. For testing a system, the implementation ought to be understood first which should be possible by making a suitable model of the system. A typical source for tests is the program code. Each time the execution of program is done, the program is tested by the user. So we need to execute the program with the particular intent of removing and fixing errors. With a specific end goal to locate the highest conceivable number of errors, there must be conduction of tests systematically and test cases ought to be designed by utilizing restrained techniques. UML (Unified Modeling Language) is a broadly accepted arrangement of notations for modeling the system which is object oriented. It has numerous diagrams for portraying the dynamic behavior of objects in the system. [3]

The requirement for better quality means more pressure for testing of software and for test engineers dealing with it. Software Testing is the procedure of executing a program with the goal of discovering errors. Each code of software has been assessed and confirmed through activities of Software Quality and Assurance yet these activities definitely not sufficient. Every time the delivery of the software to the client is done, it has been tested thoroughly by the client before gets it to the production. In this way QA team needs to test the software before it gets to the client. Flow of information of testing is said to be as a technique of testing which indicates the technique to choose input test cases and analyze results of test. Majority of the engineers of software would concur that testing is a crucial component of the process of quality software, and is a standout among the most costly and challenging activities carried out during development and maintenance of software. The process of software testing has been depicted as a sequence of phases, steps and procedures that bring about Quality Software Development Delivery to the production. Hence Testing can be depicted as a procedure utilized for establishing that the software has achieved a predetermined degree of quality regarding selected attributes. [1]

Testing is an imperative phase development and maintenance of the software. It is a challenging task for investigation of unified modeling language (UML) models following the information about a system is dispersed across various views of model. The models of UML are proposed to help to minimize the complexity of the problem, with the increment in complexities and sizes of product. Still, the models of UML themselves turn out to be complex and large including a numerous interactions crosswise over several objects. It is unwieldy for producing test models such as control flow graph from source code. This is particularly valid for large programs. The UML sequence diagrams are utilized for modeling distinct behavior of an object through sequence graph. Such transitions and states are discriminating to choose the invocations of particular operation that would be made taking into account the conditions emerging during the execution of a scenario. For testing at unit level, we can get tests from state machine diagrams of UML, which embody the behavioral depiction of every component. [4]

The information about system is disseminated over a various views of the model of the system, captured through substantial number of diagrams. The models of UML are expected to help in reducing the complexity of the problem, with increase in complexities and sizes of the product. Still, the models of UML themselves become complex and large include a large number of interactions crosswise over many objects. Many products of software are based on state. In such type of systems, the behavior of system is depicted by its state. In this manner, unless a system is made to accept all its conceivable states and tested, it would not be possible to reveal bugs based on state. Adequate testing of the system of such type of software needs satisfactory coverage of transitions and states of the system. It is a noteworthy issue to fulfill the requirement specification for the production of high-quality software systematically. A legitimately generated test suite may not just find the errors in the system of software and additionally help in minimizing the high cost connected with software testing. [4]

2. AUTOMATED TESTING

Automated software testing is a procedure in which tools of software execute pre-scripted tests on an application of software before it is discharged into production. The aim of automated testing is to simplify as much of the effort of testing as conceivable with minimum set of scripts. If unit testing devours a huge percentage of resources of the team of quality assurance (QA), such as then this process may be a decent candidate for automation. The tools of automated testing have the ability of executing tests, reporting results and contrasting results with prior runs of test. Tests completed out with these tools can be repeatedly run at any time of day.

The strategy of process being utilized to execute automation is known as test automation framework. Various frameworks have been implemented over many years by organizations of testing and commercial vendors. Automation testing with commercial off-the-shelf (COTS) or software with open source can be complicated, as they mostly need customization. In most organizations, automation is implemented only when it has been resolved when it has been depicted that the program of manual testing is not fulfilling expectations and it is unrealistic to acquire most human testers. [5]
The steps of automation process are [6]:
1. Test Tool selection
2. Define scope of automation
3. Planning. Design and Development
4. Test Execution
5. Maintenance

3. UML DIAGRAM

Unified Modeling Language (UML) is a standardized used generally for modeling of language in the field of software engineering. The Unified Modeling Language incorporates a set of techniques of graphic notation to produce visual models of object-oriented software-intensive systems. The Unified Modeling Language was developed by Grady Booch, Ivar Jacobson and Jim Rumbaing at Rational Software in 1990s. It was adopted by the Object Management Group (OMG) in 1997, and has been overseen by this association ever since. In 2000, the Unified Modeling Language was acknowledged by the International Standard Organization for Standardization (ISO) as standard for Industry for modeling intensive systems of software. The Unified Modeling Language is becoming fastly a required skill for anyone virtually incorporated in projects of software. UI designers, project managers and testers, requirement analysts, architects, software developers and database professionals are progressively being asked to make and use specifications which are written in UML. It clearly gives explanation for the concepts such as enterprise architecture, business modeling and objects oriented analysis. UML is now a de-facto standard for modeling of software. There are mainly two reasons the requirement of modeling l

anguage. Firstly, the model offers a blueprint for developers so they exactly know what they need to develop and for pr oject managers so that they calculate precisely the cost of gi ven project. Secondly and more significantly, UML is the bri dge between non-technical users and technical developers. [7]

The UML became popular because of the following reasons [7]:

1. It is very flexible. It permits many distinctive types of mo deling such as Defining database architecture, Sequencing o f events and Business Process modeling event workflow.
2. Supports Methodologies which are Object-Oriented: UML supports object-oriented methodologies. UML can be utiliz ed to design both object oriented and relational models.
3. Platform-Independent: It permits architecture of softwar e to model any type of application on any operating system in any language of programming or any network.
4. The integration of different stages of SDLC through tools of UML has brought the analyst, modeler, designer, and the developers of software application closer to one another.

4. RELATED WORK

Lots of research work has been done in the area of separabl e reversible data hiding. Many efficient methods have been proposed for reversible data hiding for encrypted images. S ome important work in the area of reversible data hiding is as follows:

In reference [5], a novel approach for separable reversible d ata hiding in encrypted images is proposed. In the first stag e, owner of the matter encrypts the original image which is uncompressed by utilizing the encryption key. At that point, data hider may compress the least significant bits of the en crypted image utilizing and key of data hiding to make an in adequate space to oblige some extra information. With an e ncoded image containing additional information, if a receiv er has the key of data hiding, he can obtain the extra inform ation however he doesn't know the content of an image. If t he receiver has both the key of encryption and key of data h iding, he can obtain the extra information and also recupera te the original content with no lapse by misusing the spatial relationship in regular image when the amount of extra inf ormation is not too very expansive.

In Paper [6], authors propose an authenticated and secure d iscrete reversible data hiding in cipher images manages aut hentication and security. In the first stage, owner of the con tent encodes the image which is original and uncompressed utilizing an encryption key. At that point, data hider may co mpress the least significant bits of the encrypted image utili zing and key of data hiding to make an inadequate space to oblige some extra information. With an encoded image cont aining additional information, if a receiver has the key of da ta hiding, he can obtain the extra information however he d oesn't know the content of an image. If the receiver has bot h the key of encryption and key of data hiding, he can obtai n the extra information and also recuperate the original con tent with no lapse by misusing the spatial relationship in re gular image when the amount of extra information is not to o very vast. It is additionally a downside that if the receiver has any one key as known, and afterward he can take any o ne data from the encoded information. SHA-1 algorithm is b eing utilized keeping in mind to achieve authentication.

Reference [7] explains about the technique for reversible da ta hiding in encrypted images using DCT. Reversible data hi ding is a strategy to insert extra message into some spread media with a reversible way so that the original content can be consummately restored after extraction of the message which is hidden. The extraction of data can be accomplished by analyzing the block smoothness. This paper adopts a pla n for measuring the blocks smoothness, and uses the schem e of closest match to further diminishing the rate of error of removed bits. The results of experiment reveal that the pro posed technique offers better execution over side match sys tem.

In reference [8], more consideration is paid to reversible da ta hiding in images that are encrypted, since it keeps up the amazing property that the original cover can be losslessly r ecuperated after inserted data is obtained while securing th e content of an image confidentially. All past techniques em bed information by reversibly memory space from the encr ypted images, which may be liable to few errors on extracti on of data or restoration of image. A novel strategy is propo sed in this paper by saving space of memory before encrypt ion with a conventional RDH (Reversible Data Hiding) techn ique, and accordingly it is simple for the information hider t o reversibly embed information in the image.

In paper [9], method of data hiding by adjusting histogram of medical images and dissimilarities in view of division of b lock is proposed. Reversible data hiding by utilizing method of modification of histogram considers the distinctions of t he pixels which enhance the capacity of data hiding. To imp rove the capacity of data hiding, the method of block divisio n is preferred.

5. MOTIVATION

Reversible data hiding is a procedure through which origin al image can be recuperated back losslessly after embedded message is extracted. This technique is used in law forensic s, military and medical images where no alteration of the or iginal image is permitted. The main aim of this study is to hi de the data in encrypted images. In this work, AOI (area of i nterest) is finding out and fuse the related document in the NAOI (non area of interest) of the image. The data is fused a t the borders of the images and then build the particular an d pre-defined border space. The algorithm is proposed whic h will first find out the area of interest and then noisy pixels of an image. BAT algorithm is used to find the coordinates of the noisy pixels.

The research is based on following objectives:

1. To Study various reversible data hiding techniques with evaluation parameters.
2. To study and implement BAT algorithm for generating hash key.

3. Implementing hash key for data fusion into the host image.
4. Decrypting host image and data from embedded image.
5. Evaluating various parameters for calculating the percentage of improvement.

6. PROPOSED SCHEME

Generally there are two types of techniques of data hiding-separable reversible data hiding and non-separable reversi ble data hiding. The aim of this research is to hide the data r eversibly using BAT optimization for encrypted images. In i t, area of interest is find out for particular image and then fu se the related document in the NAOI of the image. The techn ique used is to fuse data at the borders of the image and bui ld the particular and pre-defined border space. An algorith m is proposed which will first find out area of interest and t hen use BAT algorithm to find out the coordinates of the noi sy pixels and embed text data on these pixels and rest data will be fed into the border area of images.

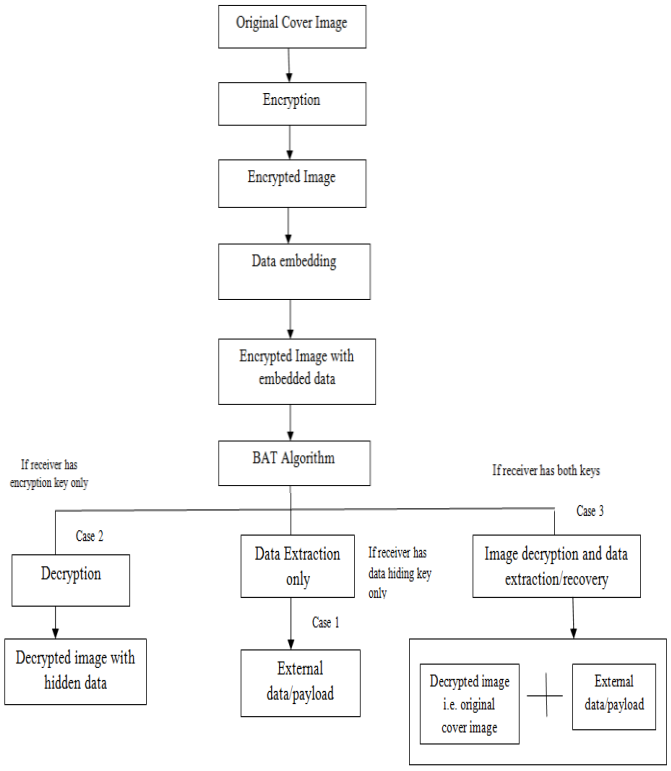


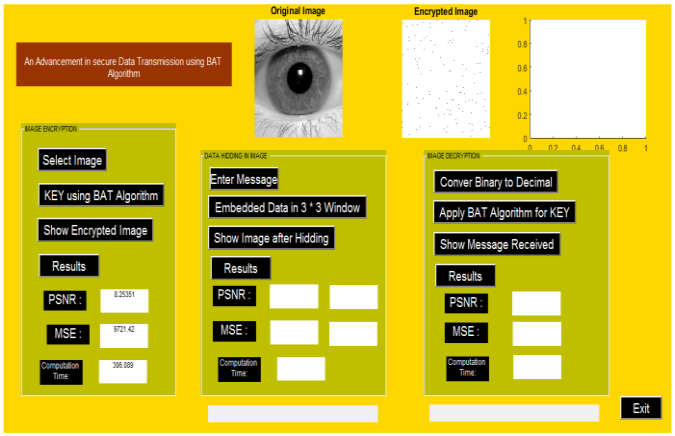
Fig. 3 Flowchart of proposed work

7. RESULTS AND DISCUSSIONS

This section presents the simulated results of encryption, data hiding, decryption and computation time.

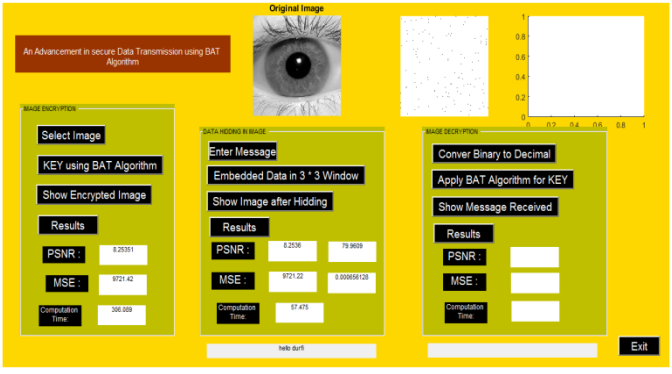
- **Encryption and Computation Time**

```
*****************************************
*                                       *
*       S - B O X   C R E A T I O N     *
*                                       *
*   (this might take a few seconds ;-))  *
*                                       *
*****************************************

    s_box : 63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76
            ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0
            b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15
            04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75
            09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84
            53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf
            d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8
            51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2
            cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73
            60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db
            e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79
            e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08
            ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a
            70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e
            e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df
            8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16


inv_s_box : 52 09 6a d5 30 36 a5 38 bf 40 a3 9e 81 f3 d7 fb
            7c e3 39 82 9b 2f ff 87 34 8e 43 44 c4 de e9 cb
            54 7b 94 32 a6 c2 23 3d ee 4c 95 0b 42 fa c3 4e
            08 2e a1 66 28 d9 24 b2 76 5b a2 49 6d 8b d1 25
            72 f8 f6 64 86 68 98 16 d4 a4 5c cc 5d 65 b6 92
            6c 70 48 50 fd ed b9 da 5e 15 46 57 a7 8d 9d 84
            90 d8 ab 00 8c bc d3 0a f7 e4 58 05 b8 b3 45 06
            d0 2c 1e 8f ca 3f 0f 02 c1 af bd 03 01 13 8a 6b
            3a 91 11 41 4f 67 dc ea 97 f2 cf ce f0 b4 e6 73
            96 ac 74 22 e7 ad 35 85 e2 f9 37 e8 1c 75 df 6e
            47 f1 1a 71 1d 29 c5 89 6f b7 62 0e aa 18 be 1b
            fc 56 3e 4b c6 d2 79 20 9a db c0 fe 78 cd 5a f4
            1f dd a8 33 88 07 c7 31 b1 12 10 59 27 80 ec 5f
            60 51 7f a9 19 b5 4a 0d 2d e5 7a 9f 93 c9 9c ef
            a0 e0 3b 4d ae 2a f5 b0 c8 eb bb 3c 83 53 99 61
            17 2b 04 7e ba 77 d6 26 e1 69 14 63 55 21 0c 7d


*********************************************
*                                           *
*         R C O N   C R E A T I O N         *
*                                           *
*********************************************

rcon : 01 00 00 00
       02 00 00 00
       04 00 00 00
       08 00 00 00
       10 00 00 00
       20 00 00 00
       40 00 00 00
       80 00 00 00
       1b 00 00 00
       36 00 00 00


*********************************************
*                                           *
*         K E Y   E X P A N S I O N         *
*                                           *
*********************************************

w(1:4, :) :         2d 0d 61 22
                    3e 2f 01 5d
                    13 21 17 28
                    0a 26 03 07

After rot_word :    26 03 07 0a

After sub_bytes :   f7 7b c5 67

rcon(05, :) :       01 00 00 00

After rcon xor :    f6 7b c5 67

w(05, :) :          db 76 a4 45

w(06, :) :          e5 59 a5 18

w(07, :) :          f6 78 b2 30

w(08, :) :          fc 5e b1 37

After rot_word :    5e b1 37 fc
```




```
*********************************************
*                                           *
*    P O L Y _ M A T   C R E A T I O N      *
*                                           *
*********************************************

    poly_mat : 02 03 01 01
               01 02 03 01
               01 01 02 03
               03 01 01 02

inv_poly_mat : 0e 0b 0d 09
               09 0e 0b 0d
               0d 09 0e 0b
               0b 0d 09 0e
```

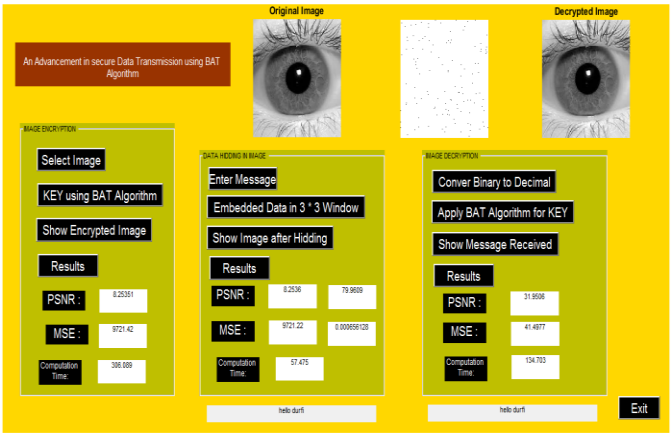
- **Data Hiding and Computation Time**



- **Decryption and Computation Time**

```
s_box : 63 7c 77 7b f2 6b 6f c5 30 01 67 2b fe d7 ab 76
        ca 82 c9 7d fa 59 47 f0 ad d4 a2 af 9c a4 72 c0
        b7 fd 93 26 36 3f f7 cc 34 a5 e5 f1 71 d8 31 15
        04 c7 23 c3 18 96 05 9a 07 12 80 e2 eb 27 b2 75
        09 83 2c 1a 1b 6e 5a a0 52 3b d6 b3 29 e3 2f 84
        53 d1 00 ed 20 fc b1 5b 6a cb be 39 4a 4c 58 cf
        d0 ef aa fb 43 4d 33 85 45 f9 02 7f 50 3c 9f a8
        51 a3 40 8f 92 9d 38 f5 bc b6 da 21 10 ff f3 d2
        cd 0c 13 ec 5f 97 44 17 c4 a7 7e 3d 64 5d 19 73
        60 81 4f dc 22 2a 90 88 46 ee b8 14 de 5e 0b db
        e0 32 3a 0a 49 06 24 5c c2 d3 ac 62 91 95 e4 79
        e7 c8 37 6d 8d d5 4e a9 6c 56 f4 ea 65 7a ae 08
        ba 78 25 2e 1c a6 b4 c6 e8 dd 74 1f 4b bd 8b 8a
        70 3e b5 66 48 03 f6 0e 61 35 57 b9 86 c1 1d 9e
        e1 f8 98 11 69 d9 8e 94 9b 1e 87 e9 ce 55 28 df
        8c a1 89 0d bf e6 42 68 41 99 2d 0f b0 54 bb 16

inv_s_box : 52 09 6a d5 30 36 a5 38 bf 40 a3 9e 81 f3 d7 fb
            7c e3 39 82 9b 2f ff 87 34 8e 43 44 c4 de e9 cb
            54 7b 94 32 a6 c2 23 3d ee 4c 95 0b 42 fa c3 4e
            08 2e a1 66 28 d9 24 b2 76 5b a2 49 6d 8b d1 25
            72 f8 f6 64 86 68 98 16 d4 a4 5c cc 5d 65 b6 92
            6c 70 48 50 fd ed b9 da 5e 15 46 57 a7 8d 9d 84
            90 d8 ab 00 8c bc d3 0a f7 e4 58 05 b8 b3 45 06
            d0 2c 1e 8f ca 3f 0f 02 c1 af bd 03 01 13 8a 6b
            3a 91 11 41 4f 67 dc ea 97 f2 cf ce f0 b4 e6 73
            96 ac 74 22 e7 ad 35 85 e2 f9 37 e8 1c 75 df 6e
            47 f1 1a 71 1d 29 c5 89 6f b7 62 0e aa 18 be 1b
            fc 56 3e 4b c6 d2 79 20 9a db c0 fe 78 cd 5a f4
            1f dd a8 33 88 07 c7 31 b1 12 10 59 27 80 ec 5f
            60 51 7f a9 19 b5 4a 0d 2d e5 7a 9f 93 c9 9c ef
            a0 e0 3b 4d ae 2a f5 b0 c8 eb bb 3c 83 53 99 61
            17 2b 04 7e ba 77 d6 26 e1 69 14 63 55 21 0c 7d

*********************************************
*                                           *
*        R C O N   C R E A T I O N          *
*                                           *
*********************************************

rcon : 01 00 00 00
       02 00 00 00
       04 00 00 00
       08 00 00 00
       10 00 00 00
       20 00 00 00
       40 00 00 00
       80 00 00 00
       1b 00 00 00
       36 00 00 00

*********************************************
*                                           *
*        K E Y   E X P A N S I O N          *
*                                           *
*********************************************

w(1:4, :) :        2d 0d 61 22
                   3e 2f 01 5d
                   13 21 17 28
                   0a 26 03 07

After rot_word :   26 03 07 0a

After sub_bytes :  f7 7b c5 67

rcon(05, :) :      01 00 00 00

After rcon xor :   f6 7b c5 67

w(05, :) :         db 76 a4 45

w(06, :) :         e5 59 a5 18

w(07, :) :         f6 78 b2 30

w(08, :) :         fc 5e b1 37

After rot_word :   5e b1 37 fc
```

```
*****************************************
*                                       *
*    P O L Y _ M A T   C R E A T I O N  *
*                                       *
*****************************************

    poly_mat : 02 03 01 01
               01 02 03 01
               01 01 02 03
               03 01 01 02

inv_poly_mat : 0e 0b 0d 09
               09 0e 0b 0d
               0d 09 0e 0b
               0b 0d 09 0e

recieved_msg =

hello durfi
```



8. CONCLUSION

The main aim of this research is to implement BAT algorithm for encrypted images to hide the data. Reversible data hiding is a procedure in which original image can be regained back without any loss and also after getting the embedded message. It hides the secret information in the digital image in such a way that just the approved individual could interpret the secret data and restore the original image. We have used BAT algorithm for hiding data. Hence if the image is transmitted, it is secured. Our proposed algorithm has shown better results than the algorithms already present for encryption of images.

REFERENCES

[1] V.Santhosh, J.Harish, R.V.K Sumanth, N.Vijay "Separable Reversible Data Hiding In Encrypted Image", Department of CSE, GRIET..

[2] Rini.J, "Study on Separable Reversible Data Hiding in Encrypted Images", *Intenational Journal of Advancements in Research & Technology* (IJOART), Vol.2, No.12, December 2013.

[3] S.Poongodi, Dr.B.Kalavathi, M.Shanmugapriya "Secure Transformation of Data in Encrypted Image using Reversible Data hiding Technique", *International Journal of Engineering Science and Innovative Technology (IJESIT),* Vol.2, No.4, July 2013.

[4] Vinit K.Agham, Tareek M Pattewar "Separable Reversible Data Hiding Techniqe Based on RGB-LSB Method", *International Journal of Research in Advent Technology*, Vol.1, No.3, October 2013.

[5] Xinpeng Zhang, "Separable Reversible Data Hiding in Encypted Image", *IEEE* , Vol.7, No.2, 2012.

[6] C.Anuradha, S.Lavanya, "Secure and Authenticated Reversible Data Hiding in Encrypted Image", *International Journal of Advanced Research in Computer Science and Software Engineering,*, Vol 3, Issue 4,April 2013.

[7] Vimal, Mahendra Kumar Patil, "Reversible Data Hiding In Encrypted Images using DCT*", International Journal of Engineering And Science*, Vol.3 Issue 3, June 2013, pp. 45-49.

[8] Lalit Dhande, Priya Khune, Vinod Deore, Dnyaneshwar Gawade, "Hide Inside-Separable Reversible Data Hiding in Encrypted Image", *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* Vol.3, Issue 9, February 2014.

[9] A.Lavanya, V.Natarajan, "Data Hiding Using Histogram Modification of Difference in Medical Images Based on Block Division", International Conference on Recent Trends in Information Technology (ICRTIT), *IEEE*, 2012. [10] T. Margaret, "Reversible Data Hiding In Encrypted Images by XOR Ciphering Technique*", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol.3, Issue 2, February 2014.

[11] Shruti M. Rakhunde, "Reversible Data Hiding using Visual Cryptography: A Review", *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE),* Vol.2, Issue 1, January 2014.

[12] Masoud Nosrati, Ronak Karimi, Mehdi Hariri, "Reversible Data Hiding: Principles, Techniques and Recent Studies", *World Applied Programming Journal*, Vol. 2, Issue 5, May 2012.

[13] Dr. V Khanaa, Dr, Krishna Mohanta, "Secure and Authenticated Reversible Data Hiding In Encrypted Images" *International Journal of Engineering and Computer Science,* Vol.2, Issue 3, March 2013.

[14] Harish G, Smitha ShekarB, Prjawal R, Sunil S Shetty, "Reversible Data Hiding In Encrypted Images by Reserving Room before Encryption", *International Journal of Engineering Research*,Vol.3, Issue 7, pp 435-437, July 2014.

[15] Prof. Praveen Bhanodia, Prof. Mrudula, Prashant Kale, " Separable Reversible Data Hiding Using Matrix Addition Compression Approach for Color Images", *International Journal of Emerging Technology and Advanced Engineering*, Vol.3, Issue 2, Februray 2013.