IMAGE STEGANOGRAPHY

**APPLICATION DEVELOPMENT – iV**

**TESTING AND DOCUMENTATION**

*Submitted in partial fulfilment for the award of the degree*

*of*

**Master of Technology**

***in***

**Information Technology**

*by*

**Bhuvana Chandra Pathi**

**(17MIN0618)**



 **School of Information Technology and Engineering**

September 2019



**School of Information Technology and Engineering**

**DECLARATION BY THE CANDIDATE**

I hereby declare that the thesis entitled **“IMAGE STEGANOGRAPHY”** submitted by me to Vellore Institute of Technology, Vellore, in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Information Technology** is a record of bonafide APD work carried out by me. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

**Place**: Hyderabad

**Date**: 30-09-2020 **Signature of the Candidate**



# School of Information Technology and Engineering

**BONAFIDE CERTIFICATE**

This is to certify that the project work entitled “**IMAGE STEGANOGRAPHY”** by **Bhuvana Chandra (17MIN0618),** to Vellore Institute of Technology University, Vellore, in partial fulfilment of the requirement for the award of the degree of **Master of Technology** in **Information Technology**, is a project bonafide work carried out by him/her under my supervision. The project fulfils the requirement as per the regulations of this Institute and in my opinion, meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this Institute or any other Institute or University.

**Dr. S. Lokesh,**

**Internal Supervisor**

**Assistant Professor**

**VIT University**

**Internal Examiner(s) External Examiner(s)**

# DOCUMENT APPROVAL

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Printed Name | Title | Date | Signature |
| Submitted By | IMAGE | 30/09/2020 |  |
|  |
| Bhuvana Chandra | STEGANOGRAPHY |  |  |
| Evaluator | IMAGE |  |  |
|  |
|  | STEGANOGRAPHY | 30/09/2020 |  |

Contents

[1. **TEST PLAN** 7](#_Toc3565748)

[1.1 Introduction 7](#_Toc3565749)

[1.1.1 Plan Test 7](#_Toc3565750)

[1.1.2 Design Test 8](#_Toc3565751)

[1.1.3 Implement Test 8](#_Toc3565752)

[1.1.4 Execute Test 8](#_Toc3565753)

[1.1.5 Evaluate Test 9](#_Toc3565754)

[1.2 Testing Methods 9](#_Toc3565755)

[1.2.1 White Box Testing 9](#_Toc3565756)

[1.2.2 Black box Testing 9](#_Toc3565757)

[1.2.3 Unit testing 10](#_Toc3565758)

[1.2.4 Integration Testing 10](#_Toc3565759)

[1.2.5 Validation Testing 10](#_Toc3565760)

[1.2.6 User Acceptance Testing 11](#_Toc3565761)

[1.2.7 Data and Database Integrity 11](#_Toc3565762)

[1.2.8 System testing 11](#_Toc3565763)

[1.2.9 Performance Testing 11](#_Toc3565764)

[1.2.10 Load Testing 11](#_Toc3565765)

[1.2.11 Stress Testing 12](#_Toc3565766)

[1.2.12 Volume Testing 12](#_Toc3565767)

[1.3 Features to be tested 12](#_Toc3565768)

[2. **TEST DESIGN / TEST CASE SPECIFICATION** 13](#_Toc3565769)

[2.1 Start up screen Display 13](#_Toc3565770)

[2.2 Encryption 14](#_Toc3565771)

[2.3 Decryption 15](#_Toc3565772)

[3. **SIMULATION AND RESULTS** 16](#_Toc3565773)

[3.1 Simulation Setup: 16](#_Toc3565774)

[3.2 Perceptual Quality: 16](#_Toc3565775)

[3.4 Embedding Capacity 18](#_Toc3565776)

[3.5 Mean Square Error (MSE): 19](#_Toc3565777)

[3.6 Peak Signal to Noise Ratio (PSNR): 19](#_Toc3565778)

[3.7 Histograms of Transform domain techniques: 20](#_Toc3565779)

[4. **CONCLUSION AND FUTURE SCOPE** 23](#_Toc3565780)

[4.1 Conclusion 23](#_Toc3565781)

[4.2 Future Scope 23](#_Toc3565782)

# TEST PLAN

## 1.1 Introduction

Is the menu bar displayed in the appropriate contested some system related features included either in menus or tools? Do pull –Down menu operation and Tool-bars work properly? Are all menu function and pull-down sub function properly listed; Is it possible to invoke each menu function using a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.? In adequate testing or non-testing will leads to errors that may appear few months later.

This create two problems

1. Time delay between the cause and appearance of the problem.
2. The effect of the system errors on files and records within the system

The purpose of the system testing is to consider all the likely variations to which it will be suggested and push the systems to limits.

The testing process focuses on the logical intervals of the software ensuring that all statements have been tested and on functional interval is conducting tests to uncover errors and ensure that defined input will produce actual results that agree with the required results. Program level testing, modules level testing integrated and carried out.

|  |
| --- |
| 1.1.1 Plan Test |
| * Identify Requirements for Test |
| * Assess Risk |
| * Develop Test Strategy |
| * Identify Test Resources |
| * Create Schedule |
| * Generate Test Plan |

|  |
| --- |
| 1.1.2 Design Test |
| * Workload Analysis |
| * Develop Test Suite |
| * Identify and Describe Test Cases |
| * Identify and Structure Test Scripts |
| * Review and Access Test Coverage |
| 1.1.3 Implement Test |
| * Setup Test Environment |
| * Record or Program Test Scripts |
| * Develop Test Stubs and Drivers |
| * Identify Test-Specific functionality in the design and implementation model |
| * Establish External Data sets |
| 1.1.4 Execute Test |
| * Execute Test Scripts |
| * Evaluate Execution of Test |
| * Recover from Halted Test |
| * Verify the results |
| * Investigate Unexpected Results |
| * Log Defects |

|  |
| --- |
| 1.1.5 Evaluate Test |
| * Evaluate Test-Case Coverage |
| * Evaluate Code Coverage |
| * Analyze Defects |
| * Determine if Test Completion Criteria and Success Criteria have been achieved |
| * Create Test Evaluation Report |

## 1.2 Testing Methods

There are two major type of testing they are

1. White Box Testing.
2. Black Box Testing.

### 1.2.1 White Box Testing

White box sometimes called “Glass box testing” is a test case design uses the control structure of the procedural design to drive test case.

Using white box testing methods, the following tests were made on the system

1. All independent paths within a module have been exercised once. In our system, ensuring that case was selected and executed checked all case structures. The bugs that were prevailing in some part of the code where fixed
2. All logical decisions were checked for the truth and falsity of the values.

### 1.2.2 Black box Testing

Black box testing focuses on the functional requirements of the software. This is black box testing enables the software engineering to derive a set of input conditions that will fully exercise all functional requirements for a program. Black box testing is not an alternative to white box testing rather it is complementary approach that is likely to uncover a different class of errors that white box methods like.

* Interface errors
* Performance in data structure
* Performance errors
* Initializing and termination errors.

### 1.2.3 Unit testing

Unit testing is a software verification and validation method in which a programmer tests if individual units [of source code](http://en.wikipedia.org/wiki/Source_code) are fit for use. A unit is the smallest testable part of an application. In [procedural programming](http://en.wikipedia.org/wiki/Procedural_programming) a unit may be an individual function or procedure. Ideally, eac[h test case](http://en.wikipedia.org/wiki/Test_case) is independent from the others: substitutes like [method stubs](http://en.wikipedia.org/wiki/Method_stub), objects, fakes and [test harnesses](http://en.wikipedia.org/wiki/Test_harness) can be used to assist testing a module in isolation.

### 1.2.4 Integration Testing

This testing is sometimes called Integration and Testing. Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates and delivers as its output the integrated system ready for system testing.

### 1.2.5 Validation Testing

Validation Testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can reasonably expected by a customer. After validation test has been conducted, one of the following two possible conditions exists. The functions or performance characteristics confirm to specification and are accepted.

### 1.2.6 User Acceptance Testing

User acceptance of a system is a key factor of any system. The system under consideration is tested for the acceptance by constantly keeping in touch with the prospective system users at the same time of developing and marketing changes whenever required. This is done regarding the following points:

* Input Screen Design
* Output Screen Design

### 1.2.7 Data and Database Integrity

Testing Data integrity and database integrity test techniques verify that data is being stored by the system in a manner where the data is not compromised by updating, restoration, or retrieval processing. This type of testing is intended to uncover design flaws that may result in data corruption, unauthorized data access, lack of data integrity across multiple tables, and lack of adequate transaction performance. The database, data files, and the database or data file processes should be tested as a subsystem within the application.

### 1.2.8 System testing

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. They include the following functions: User login and logout, View Academic History, register for course, Manage Department Information, Manage Student Information, Manage Lecturer Information, Manage Course CatLog, Manage Offering Courses, Manage Financial Activities, Manage User Account

### 1.2.9 Performance Testing

Performance Testing covers a broad range of engineering or functional evaluations where a material, product, or system is not specified by detailed material or component specifications: Rather, emphasis is on the final measurable performance characteristics.

### 1.2.10 Load Testing

Load testing is the process of creating demand on a system or device and measuring its response.

### 1.2.11 Stress Testing

Stress testing is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Stress testing may have a more specific meaning in certain industries.

Verify system response during maximum user logins.

### 1.2.12 Volume Testing

Volume Testing belongs to the group of non-functional tests, which are often misunderstood and/or used interchangeably. Volume testing refers to testing a software application for a certain data volume.

Verify system response when Database at 90% capacity.

## 1.3 Features to be tested

We will perform testing on use case specification, functional requirement, and non-functional requirement of all use cases and functions. They include application load and unload, View file, View image, converting to bmp file, Manage Information, Add a file, Update a file, Viewing information, Encryption of file, Decryption of file

# TEST DESIGN / TEST CASE SPECIFICATION

## 2.1 Start-up screen Display

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test #*** | ***Description*** | ***Expected Outcome*** | ***As Expected*** |
| 1.1 | Open the application | The software will be loaded up as intended, there are no glitches or anomalies | Yes |
| 1.2 | Action > Encrypt image | The encryption screen will appear and the option to select the image will appear. | Yes |
| 1.3 | Action > Decrypt image | The decryption screen will appear and the option to select the image will appear. | Yes |
| 1.4 | Action > Browse | A dialogue box will appear to select image to choose. | Yes |
| 1.5 | Action > Close | The software will be closed. | Yes |

Table 2.1: start-up screen display

## 2.2 Encryption

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test #*** | ***Description*** | ***Expected Outcome*** | ***As Expected*** |
| 1.2.1 | Action > Encrypt image | The encryption screen will appear and the option to select the image will appear. | Yes |
| 1.2.2 | Select the image drive by clicking the browse | Image will be selected if there are no errors while selecting, in case of errors an appropriate pop up will display | Yes |
| 1.2.3 | Click on “Perform Action”  (if image loaded) | Operation will be perform. | Yes |
| 1.2.4 | Click on “Perform Action”  (if no image loaded) | Will show error. Encryption information incomplete. | Yes |
| 1.2.5 | View the image in different window | The image will be displayed in a different window | Yes |

Table 2.2 Encryption

## 2.3 Decryption

|  |  |  |  |
| --- | --- | --- | --- |
| ***Test #*** | ***Description*** | ***Expected Outcome*** | ***As Expected*** |
| 1.3.1 | Action > Decrypt image | The decryption screen will appear and the option to select the image will appear. | Yes |
| 1.3.2 | Select an image file | Image will be selected if no error while selecting. In case of any error a window will be appear. | Yes |
| 1.3.3 | User clicks the save button | Option to save file should open | Yes |
| 1.3.4 | Click on “Perform Action”  (if image loaded) | Operation will be perform. | Yes |
| 1.3.5 | Click on “Perform Action”  (if no image loaded) | Will show error. Decryption information incomplete. | Yes |

Table 2.3: Decryption

# 3. SIMULATION AND RESULTS

## 3.1 Simulation Setup:

The MATLAB Version 7.13.0.564 (R2011b) is used to implement and simulate for steganography techniques: LSB & Distortion of spatial domain steganography and Jsteg & Steg-hide of transform domain steganography. MATLAB is used because of large number of advanced inbuilt functions and image processing toolbox. We take results for various color cover image for different secret files.

The various simulation parameters are as given below:

|  |  |
| --- | --- |
| Cover image pixel size (N x N x3) | N=250, 500, 1024, 2048 |
| Secret text file size (kb) | 1,5 |
| Image type | Tiff, jpg |
| Simulation Tool | MATLAB 7.13.0.564 |
| Pseudo Random Number Generator | MATLAB rng with key 2 |
| Secret data for JPEG Steganography | Ajay Nain  Minor Project YMCA |

Table 3.1: Simulation parameter setup

The following results are taken for evaluation of these techniques:

## 3.2 Perceptual Quality:

We saw that visual quality of spatial domain steganography is better than transform domain techniques. The stego-image produced by all the 4 techniques on baby cover image with size of 500 x 500 x 3 are given below:



(a) Stego-image with LSB technique (b) Stego-image with distortion technique

Figure 3.2.2: Stego- image produced by Spatial domain techniques with secret file

We see that there is no so much change in perceptual quality of image to detect visual changes i.e. quality of embedded image is not degraded by these techniques



(a) Stego-image with Jsteg (b) Stego-image with Steg-hide

Figure 3.3.3: Stego- image produced by transfer domain techniques

Result shows that visual quality of stego-image produced by spatial domain techniques is less than that of produced by transform domain techniques. In spatial domain techniques as the size of secret file changes the more degradation is produced but in case of the degradation of image is not depend upon size of secret file so much.

## 3.4 Embedding Capacity

It is the size of the secret data that can be embed in cover image without deteriorating the integrity of the cover image. It can be represented in bytes or bit per pixel (bpp). It depends upon the characteristics of cover image and the embedding algorithm used for steganography.

The table shows the embedding capacity of the 4 techniques for different cover image:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Image size(N x N x 3)** | **LSB** | **Distortion** | **Jsteg** | **Steg-hide** |
| 250 | 23437 | 23437 | 61 | 8 |
| 500 | 93750 | 93750 | 136 | 30 |
| 1024 | 393216 | 393216 | 5798 | 797 |
| 2048 | 1572864 | 1572864 | 7843 | 829 |

Table 3.4.1: Embedding Capacity

Result shows that embedding capacity of spatial domain capacity is fix large quantity for a cover image size but capacity of transform domain techniques is very less and it is not fixed for a given size of cover image, it depends upon characteristics of cover image**.**

## 3.5 Mean Square Error (MSE):

The results of the all techniques for the given setup parameters are in the following table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Image size(N x N x 3) N=** | **LSB**  **Substitution** | **Distortion** | **Jsteg** | **Steg-hide** |
| 250 | 5.84E-07 | 1.11E-07 | 8.81E-05 | 8.87E-05 |
| 500 | 3.63E-08 | 6.94E-09 | 1.39E-05 | 1.45E-05 |
| 1024 | 3.95E-10 | 3.41E-10 | 2.20E-05 | 2.69E-05 |
| 2048 | 2.63E-11 | 2.13E-11 | 1.11E-06 | 1.23E-06 |

Table 3.5.1: MSE

We can analyze from the results that MSE for spatial domain techniques is very less than that of for transform domain technique. In case of transform domain techniques the lossy compression step of jpeg compression i.e. quantization is performed in the embedding process and hence very large MSE is produced and quality of cover image degraded more.

## 3.6 Peak Signal to Noise Ratio (PSNR):

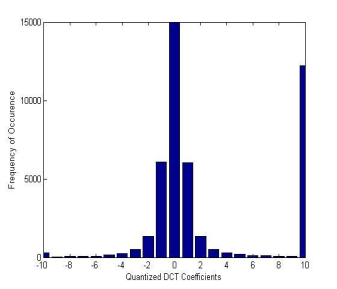
It is the measure of quality of the image by comparing the cover image with the stego-image. High PSNR indicates good perceptual quality of stego-image. The results of PSNR for all the techniques are in following table:

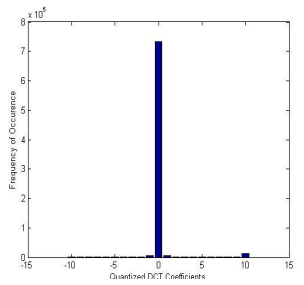
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Image Size(N x**  **N x 3)** | **LSB Substitution** | **Distortion** | **Jsteg** | **Steg-hide** |
| 250 | 110.4659 | 117.6791 | 88.6813 | 88.6532 |
| 500 | 122.5315 | 129.7203 | 96.6949 | 96.5209 |
| 1024 | 142.169 | 142.7986 | 94.7016 | 93.8368 |
| 2048 | 153.9334 | 154.8398 | 107.6790 | 107.2257 |

Table 3.6.1: PSNR in db

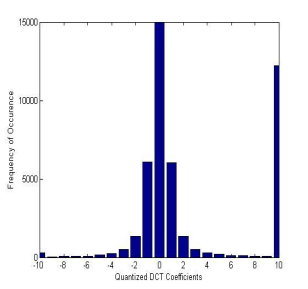
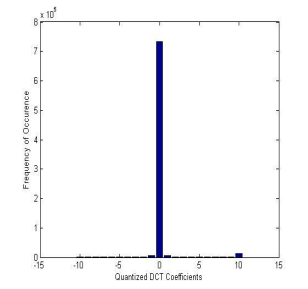
## 3.7 Histograms of Transform domain techniques:

Histogram is a measure of the number of occurrence of pixels with respect to particular pixel value. During embedding pixel value changes hence number of pixel having a particular pixel value changes. These changes can be used to detect steganography. We take histograms for baby color image with size of (500 x 500) for all the four the techniques separately.

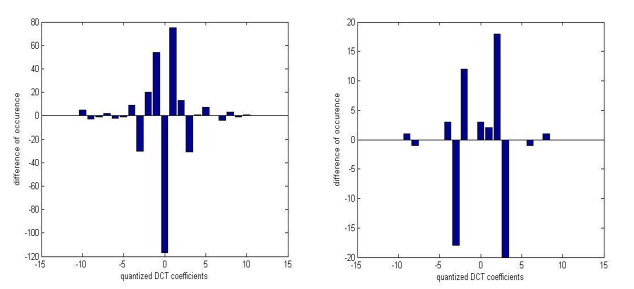
For transform domain techniques quantized coefficients are used for calculation of histogram instead of pixel value. The histogram results of jsteg and Steg-hide are below:



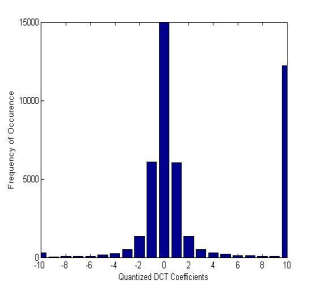
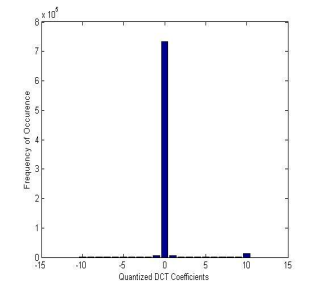
*Histogram for cover image*



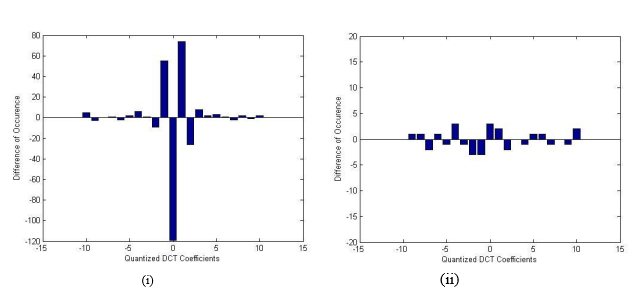
*Histogram of stego-image produced by jstag*



*Difference between original and Stego-image histogram(i) with JPEG compression (ii) without JPEG compression; stego-image is save in tif format*



*Histogram of stego-image produced by Steg-hide*



*Difference between original and Steg-hide produced stego-image histogram (i) with JPEG compression (ii) without JPEG compression; stego-image is save in tif format*

As we see if no compression is performed after embedding process the difference between the histogram is zero, hence this technique cannot be detect by steganalysis which use first order statics of stego-image.

# 4. CONCLUSION AND FUTURE SCOPE

## 4.1 Conclusion

Spatial domain techniques are easy ways to embed information, but they are highly vulnerable to even small cover modifications. Hence the size of stago-image cannot be reduced. An attacker can simply apply signal processing techniques in order to destroy the secret information entirely. In many cases even the small changes resulting out of lossy compression systems yield to total information loss. Transform domain methods hide messages in significant areas of the cover image which makes them more robust to attacks, such as compression, cropping, and some image processing. Hence lossy compression i.e. Jpeg compression can be performed and size of stago-image can be reduced. But the disadvantage of Jpeg-Steganography is that only few messages can be embedded in the cover-image. The embedding capacity of Jpeg steganography is very less than spatial domain techniques. The spatial domain techniques provide high PSNR, high perceptual quality and high embedding capacity but these not provide robustness. On the other hand transform domain provide robustness while providing very less embedding capacity, low PSNR and low perceptual quality**.**

## 4.2 Future Scope

We see there is a tradeoff between the three properties, perceptuality, embedding capacity and robustness. The new techniques should be develop to maintain the three properties at high level. The few areas which are still open in steganography are as below:

* Wavelet transform can be used to increase the embedding capacity while maintaining the robustness of Stego-image.
* Hamming coding or Matrix coding can be used to reduce the impact of steganography i.e. to increase the PSNR.
* Cryptography techniques like RSA, AES and hash functions can also be used with steganography to provide more security.