CHAPTER 8
PROPOSED ALGORITHM AS A SYSTEM

8.1 Introduction

As a part of the research work on “Digital Data Protection Using Steganography and Hardware Identification”, six algorithms were developed for providing effective data protection & software protection. The results of these algorithms were analyzed and their performance on application platforms like mobile, desktops/laptops and tablets were tested. In order to show the use of the proposed algorithms in real world applications a software protection system was developed.

Six algorithms proposed as part of this thesis were,

1. Stegano DB for secure DB storage of data using steganography techniques.
2. Pixel pattern based new Steganography algorithm which is difficult to detect.
3. Desktop/Laptop application protection using Steganography.
5. Mobile application protection using Steganography.

With the help of the above algorithms an application is developed to show how the algorithm can make use for a real-world scenario. The Explanation of working of the system is explained in the coming section.

8.2 Working Model of SteganoDB

A small application was developed in .net to implement the SteganoDB package. The main interface of the software which allows to select an image and load data into it in JSON format. There are separate options for inserting, upserting and deleting data. Clicking on “Load an Image DB” button on the screen will load an
image and read the existing records then show it on the right pane. If there are no records existing, it will display an error saying that no records could be found.

![Image of database functions](image)

**Fig.8.1 Screen shot of Main screen**

For inserting the data “Insert Data” button can be used. Then the user will get the options to insert data in user specified format. For updating the data, you can choose the option Upsert. For any deletion of inserted record can use the option Delete records. The existing records can be deleted using this option. Hence data maintenance is easy using these options. Same image can be used for storing different data and the maintenance of these records can be done using the same image is an advantage. Using existing steganography techniques this is not possible. Data cannot be adding multiple times within the existing steganography techniques. Also the maintenance of data is not easy. Data can have retrieved and see at any time and check if needed. The data insertion is shown in the next screen shot.
The records as per the input will be inserted into the image in JSON format using Steganography. For inserting records into an image Pixel Pattern based Steganography System can be used. Similarly, the Upsert and Delete buttons can be used to edit or delete the records selected. Main screen of application after inserting couple of records in the previous step.

After storing the image will be saved inside the system. Only users who have the StegoDB package will be able to insert and edit the data on the images.
8.3 Working Model of Pixel Pattern based Steganography

As part of the thesis a small application was developed in .net to implement the Pixel Pattern based Steganography. The first step is to encode the text to be hidden into the image which needs to be used. The application allows for both encoding and decoding the text to be hidden and show the o/p to the user. The application screens below, show each of the steps of data embedding and data extraction processes. For embedding the data select the image (cover image) to embed the data and then enter the message to be hidden inside the cover image after that, press the button Encode to create stego image. It will encode the text message first then analyze the cover image and find out the suitable pixels to embed the data. Then store the metadata. While retrieving the data the reverse process is taking place.

![Fig.8.4 Screen shots of embedding the data into cover image](image.png)
For extracting the text, first select the stego image and click on the button ‘Decode Stego image’. Then it will find out the metadata and retrace the data from image. The resultant data will be displayed in the screen. The example screen shot is given below.

![Steganography - Decode image](image)

Fig.8.5 Screen shots of decoded message is displaying

For testing purpose, given a small message. Lengthy data or text file input can be given as data into the image.

### 8.4 Working Model of Mobile Data Protection

In order to implement the mobile protection method proposed a windows mobile based application with additional steganography [19] based protection included. The development environment used was Visual Studio 2013, and tests were run on two mobiles. Mobile-1 was a Lumia 535 running Windows Phone version 8.1 and Mobile-2 was a Lumia 822 running Windows phone 8.1. Both phones have 1GB...
RAM. The software can however be run on even mobile phones having 512 MB RAM. Since most modern mobile phones now a days come with minimum of 1GB of RAM the software will run fine on any mobile phone. For steganography the application needs to have image manipulation capabilities.

The sample application we created for the testing purpose is a very simple application to track expenses based on location where the expense occurred. When the application is initially run, it checks for the stego image and matches the values between the image and mobile properties. If it matches, the application is started. For stego image creation first of all registering module (Authentication) is needed. It will collect the user information and fetch the unique key. Using that information, the SteganoDB software creates the data structure and it will be embedding into the selected image using Pixel Pattern based Steganography. This way the stego image will create and will be attached with the application. After protecting the application, in each run it will check for the stego image and ensure the authenticity of the user. By using this protecting module, any type of application/ file/ message can protect. The screen shots of gathering information for the key formation is given below.

In case the application is pirated and installed on a different mobile, the application will check for the registration information and will refuse to start. It will give an error message and asks the user to register the application as in the Fig. 8.8. In the same way file/message can be protected. The steps for protecting file/message is almost the same way. For communicating file/message or for storing of it can be made use of this protection frame work.
Fig. 8.6 – Screen shot of user information gathering

Fig. 8.7 – Screen shot of successful running of protected application

Fig. 8.8 Screen shot of failure of protected application which is tried for piracy
8.5 Working Model of Digital Data Protection

A sample application was developed in Visual Basic with the protection module. In the protection software there are three modules. First one is the authentication module. Second is the stego image creation and bounding of stego image with application which is to be protected. Third is the application protection.

Step 1 – Authentication Module

This system will first run the authentication module to retrieve the user information. First module runs in the user side and fetch the information needed for creating unique formation. This information gets by the licensor via mail. The unique hardware key and this information will be sent to the license issuing authority (server). To the user it will simply appear as a random generated key. The user information retrieval screen is shown below.

![Fig. 8.9 Screen shots of Registration wizard](image1)

![Fig. 8.10 Text file created after execution of authentication module](image2)
It will collect the user information and encode the information using AES encryption algorithm. This encrypted text is stored as a text file and it sends via mail to the Licensed server.

**Step 2 – Key File Creation**

This module is executed from the server side on receiving the registration key from the user. Steganography algorithm used in this module will then hide the key into an image and this resultant image file is send to the user. Second module is completely done by the licensor. Using the registration information, the licensor creates the stego image by adding the unique and identification information inside the image, using SteganoDB and Pixel Pattern based Steganography. This stego image will send to the user and it will kept with the application. The protection module will use very little system resources and hence the user will not experience any degradation in performance.

![Screen shots of Key file creation](image)

*Fig. 8.11 Screen shots of Key file creation*

![Stego image created after execution](image)

*Fig. 8.12 Stego image created after execution*
When pressing the Open Key File button, it will find out the text file received from the user and fetch the necessary information for creating the key file. All information will be converted into SteganoDB structure using SteganoDB system implementation and write it into the selected image using Pixel Pattern based Steganography. The Key file have all the Identification information. The Key file creation is done with the help of SteganoDB System and Pixel Pattern based Steganography Algorithm application. The working of those system is explained in the sections 8.2 and 8.3 by using it key file is created. All image formats like .bmp, .jpg etc. can be used as the cover image. The key is embedded inside the original image and produces the stego-image as shown in the Fig.8.12. Human eyes perception capability will not be able to identify the embedded key within the image. Hence the picture can safely hide the key inside it, with maximum confidentiality.

After creating the stego image the licensor send this stego image to the user, and this will be bound with the application which is to protect. This will act as the unique key which protect the application from piracy.

**Step 3 - Software Execution**

The user on receiving the key file can use it for registering the software. Subsequently each time when the software is started, it will get the unique property used and compare it with the unique key provided with the SteganoDB. If both of them matches, the execution be allowed. Else if the comparison fails the execution will stop with an error message. This protection can be attached with any application which is opted for protection.

Any type of file/database can be protect using this protection module. For protecting file/database the same procedure is following.
8.6 Summary

This protection system is adaptable and it can be extensible. This can be applicable to any type of system. According to application it can be slightly modified and can be used. For a variety of applications this system can be use. Using this a lot of attacks can be avoided, this is detailed in Chapter 7.