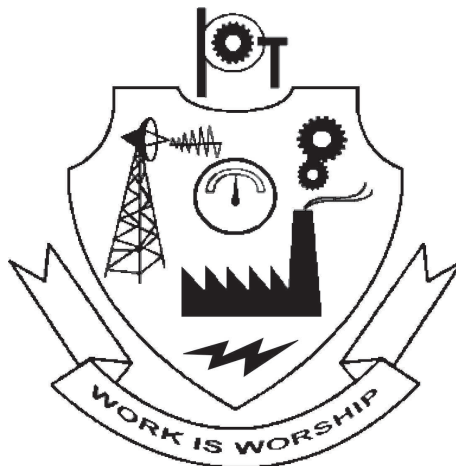


"GOVERNMENT POLYTECHNIC COLLEGE
PALAKKAD"

PROJECT REPORT 2020-2021



DEPARTMENT OF COMPUTER HARDWARE
ENGINEERING

2020-2021

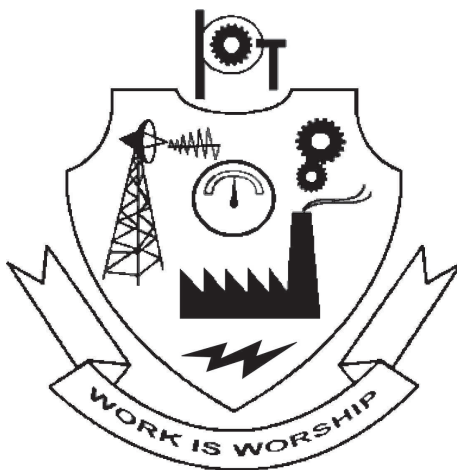
PROJECT REPORT
on

STEGANOGRAPHY

Submitted by

AJEESH M - 18150004
ANUDAS R - 18150016
MONISHA R - 18150033
MIDHUNA M - 18150030
AMAL IBRAHIM S - 18150010
YEDHU KRISHNA J C - 18150063

**GOVERNMENT POLYTECHNIC COLLEGE
PALAKKAD KERALA**



CERTIFICATE

This is to certify that the project report titled "**STEGANOGRAPHY**" has been presented by **ANUDAS R , AJEESH M , MONISHA R , MIDHUNA M , AMAL IBRHIM S , YEDHU KRISHNA J C** of final year Computer Hardware Engineering in partial fulfilment of the requirement for the award of Diploma in Computer Hardware Engineering under the Department of Technical Education, Kerala state during the year 2020-21.

Head of Department

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

First of all I thank GOD for showering his infinite blessings throughout my life. I extend my sincere thanks to Mr.SAJAN E.V, Head of department and Mrs.THAHIRA ,Lecturer in Computer Department for their co-operation and full support.

I would like to express my sincere and profound gratitude to all the staff members of our department for remaining the source of inspiration and encouragement to enable me to succeed in this endeavour.

PROJECT GROUP 9

AJEESH M - 18150004

ANUDAS R - 18150016

MONISHA R - 18150033

MIDHUNA M - 18150030

AMAL IBRAHIM S - 18150010

YEDHU KRISHNA J C - 18150063

Contents

1	INTRODUCTION	5
2	COMPONENTS USED	6
2.1	STEGHIDE	6
3	Sources Used	10
3.1	HTML	10
3.2	JAVASCRIPT	12
3.3	CSS	14
4	WORKING MECHANISM	16
5	BLOCK DIAGRAM	18
6	CODES	20
6.1	Html	20
6.2	scripts.js	23
6.3	Styles.css	27
7	CONCLUSION	31
8	REFERENCE	32

ABSTRACT

The project is a website for image steganography. Image steganography is the technique to hide a secret message inside an image. In the future cyber security is vital. In this method, the image is embedded with a secret message and sends to the receiver. No one can see the message inside this image file. Once the encoded image is received, it can be decoded using this same website. So this website contains two features, one is encoding and other one is decoding. After encoding the image and secret text message that we want to send, the encoded file will be in the image format which is very similar to the previous image. There is a download option to download the encoded image.

Chapter 1

INTRODUCTION

In the present life cyber security is a important thing which everybody want.Its because,not everything is safe in the cyber world.Hackers can steal anything which they want.Privacy problems are there.People using social medias donot know who all are seeing their private messages,files etc.These are all the trust issues we all have in the cyber world.Still though,we are all using it.We cannot avoid it any-more,because in the present life style for an adult person he needs his smart phone more than anything.So for all these people cyber security is a must.

For making our personal files and messages even more safe and secured steganography is a good way.By putting the files or images that we want to send in a normal image,others will not find it.Because that file cannot be seen if some others steal it or hack it.Only by decoding the image we will get that secret message.Here the pixels of the image are converted to string and embeded with the secret message which is already in a string format.

Chapter 2

COMPONENTS USED

2.1 STEGHIDE

Steghide is a steganography program that is able to hide data in various kinds of image- and audio-files. The color- respectively sample-frequencies are not changed thus making the embedding resistant against first-order statistical tests.

Features include the compression of the embedded data, encryption of the embedded data and automatic integrity checking using a checksum. The JPEG, BMP, WAV and AU file formats are supported for use as cover file. There are no restrictions on the format of the secret data. .

At first, the secret data is compressed and encrypted. Then a sequence of positions of pixels in the cover file is created based on a pseudo-random number generator initialized with the passphrase (the secret data will be embedded in the pixels at these positions). Of these positions those that do not need to be changed (because they already contain the correct value by chance) are sorted out. Then a graph-theoretic matching algorithm finds pairs of positions such that exchanging their values has the effect of embedding the corresponding part of the secret data. If the algorithm cannot find any more such pairs all exchanges are actually performed. The pixels at the remaining positions (the positions that are not part of such a pair) are also modified to contain the embedded data (but this is done by overwriting them, not by exchanging them with other pixels). The fact that (most of) the embedding is done by exchanging pixel values implies that the first-order statistics (i.e. the number

of times a color occurs in the picture) is not changed. For audio files the algorithm is the same, except that audio samples are used instead of pixels..

The default encryption algorithm is Rijndael with a key size of 128 bits (which is AES - the advanced encryption standard) in the cipher block chaining mode. If you do not trust this combination for whatever reason feel free to choose another algorithm/mode combination (information about all possible algorithms and modes is displayed by the encinfo command). The checksum is calculated using the CRC32 algorithm.

EMBEDDING

You should use the "embed" command if you want to embed secret data in a cover file. The following arguments can be used with the "embed" command: .

-cf, - -coverfile filename

Specify the cover file that will be used to embed data. The cover file must be in one of the following formats: AU, BMP, JPEG or WAV. The file-format will be detected automatically based on header information (the extension is not relevant). If this argument is omitted or filename is -, steghide will read the cover file from standard input.

-sf, - -stegofile filename

Specify the name for the stego file that will be created. If this argument is omitted when calling steghide with the "embed" command, then the modifications to embed the secret data will be made directly to the cover file without saving it under Specify the compression level. The compression level can be any number in 1...9 where 1 means best speed and 9 means best compression.

-Z, - -dontcompress

Do not compress the secret data before embedding it.

-K, - -nochecksum

Do not embed a CRC32 checksum. You can use this if the secret data already contains some type of checksum or if you do not want to embed those extra 32 bits needed for the checksum.

-N, - -dontembedname

Do not embed the file name of the secret file. If this option is used, the extractor needs to specify a filename to tell steghide where to write the embedded data.

EXTRACTING

If you have received a file that contains a message that has been embedded with steghide, use the "extract" command to extract it. The following arguments can be used with this command.

-sf, - -stegofile filename

Specify the stego file (the file that contains embedded data). If this argument is omitted or filename is -, steghide will read a stego file from standard input.

-xf, - -extractfile filename

Create a file with the name filename and write the data that is embedded in the stego file to it. This option overrides the filename that is embedded in the stego file. If this argument is omitted, the embedded data will be saved to the current directory under its original name.



```
firzen@kali:/tmp$ steghide extract -sf ~/Downloads/File_Patern_test.jpg
Enter passphrase:
could not extract any data with that passphrase!".
firzen@kali:/tmp$ steghide extract -sf ~/Downloads/File_Patern_test.jpg
Enter passphrase:
could not extract any data with that passphrase!" does already exist. overwrite ? (y/n)

could not extract any data with that passphrase!".
firzen@kali:/tmp$ ls
hsperfdata_firzen
ssh-sE35P4iEkkPi
systemd-private-c2c4f54d2ee64b818c6e1f3b02860937-color.service-eKlujd
systemd-private-c2c4f54d2ee64b818c6e1f3b02860937-haveged.service-p6orr4
systemd-private-c2c4f54d2ee64b818c6e1f3b02860937-ModemManager.service-EihTxu
systemd-private-c2c4f54d2ee64b818c6e1f3b02860937-systemd-logind.service-ZsDAVv
systemd-private-c2c4f54d2ee64b818c6e1f3b02860937-upower.service-q23qKk
Temp-20c4fff9-a5c3-478e-958b-7f009cc5fd58
Temp-29004d63-5e19-42ce-a615-e85426f4f2de
'test'$'\033'[1000Dcould not extract any data with that passphrase!'
tmux-1001
'VSCode Crashes'
```

Steghide

Chapter 3

Sources Used

3.1 HTML

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

Elements and Tags

HTML uses predefined tags and elements which tell the browser how to properly display the content. Remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

Html Page Structure

The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

```
<!DOCTYPE html>
<html>

  <head>
    <title> Title here </title>
  </head>

  <body>
    Web page content goes here.
  </body>

</html>
```

Features of Html

- It is easy to learn and easy to use.
- It is platform-independent.
- Images, videos, and audio can be added to a web page.
- Hypertext can be added to text.
- It is a markup language.

3.2 JAVASCRIPT

JavaScript is the world most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for Client-side developments as well as Server-side developments.

Internal JS

We can add JavaScript directly to our HTML file by writing the code inside the `<script>` tag. The `<script>` tag can either be placed inside the `<head>` or the `<body>` tag according to the requirement.

External JS

We can write JavaScript code in other file having an extension `.js` and then link this file inside the `<head>` tag of the HTML file in which we want to add this code.

Syntax

```
<script>  
// JavaScript Code  
</script>
```

Advantages Of Javascript

- Less server interaction
- Immediate feedback to the visitors
- Increased interactivity
- Richer interfaces

JavaScript Development Tools

One of major strengths of JavaScript is that it does not require expensive development tools. You can start with a simple text editor such as Notepad. Since it is an interpreted language inside the context of a web browser, you don't even need to buy a compiler.

To make our life simpler, various vendors have come up with very nice JavaScript editing tools. Some of them are listed here -

- *Microsoft FrontPage* - Microsoft has developed a popular HTML editor called FrontPage. FrontPage also provides web developers with a number of JavaScript tools to assist in the creation of interactive websites.
- *Macromedia Dreamweaver MX* - Macromedia Dreamweaver MX is a very popular HTML and JavaScript editor in the professional web development crowd. It provides several handy pre-built JavaScript components, integrates well with databases, and conforms to new standards such as XHTML and XML.
- *Macromedia HomeSite 5* - HomeSite 5 is a well-liked HTML and JavaScript editor from Macromedia that can be used to manage personal websites effectively.

Enabling JavaScript in Browsers

All the modern browsers come with built-in support for JavaScript. Frequently, you may need to enable or disable this support manually.

3.3 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

CSS is easy to learn and understood, but it provides powerful control over the presentation of an HTML document.

Syntax

A CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document.

A style rule set consists of a selector and declaration block.

Selector – h1

Declaration – color:blue;font size:12px;

CSS Selectors

1. *THE UNIVERSAL SELECTORS* : Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type
2. *THE ELEMENT SELECTOR* : The element selector selects elements based on the element name. You can select all p elements on a page like this (in this case, all p elements will be center-aligned, with a red text color)
3. *THE DESCENDANT SELECTOR* : Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to the em element only when it lies inside the ul tag.
4. *THE ID SELECTOR* : The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by the id of the element. The style rule below will be applied to the HTML element with id="para1":

Advantages of CSS

- *CSS saves time* - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- *Easy maintenance* - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- *Superior styles to HTML* - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- *Multiple Device Compatibility* - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- *Global web standards* - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Chapter 4

WORKING MECHANISM

ENCODING

The working of this project is very simple. We have to select the image which is used to hide secret message in it. To embed the secret message inside that image, we need to convert the selected image file into a base64 type. .

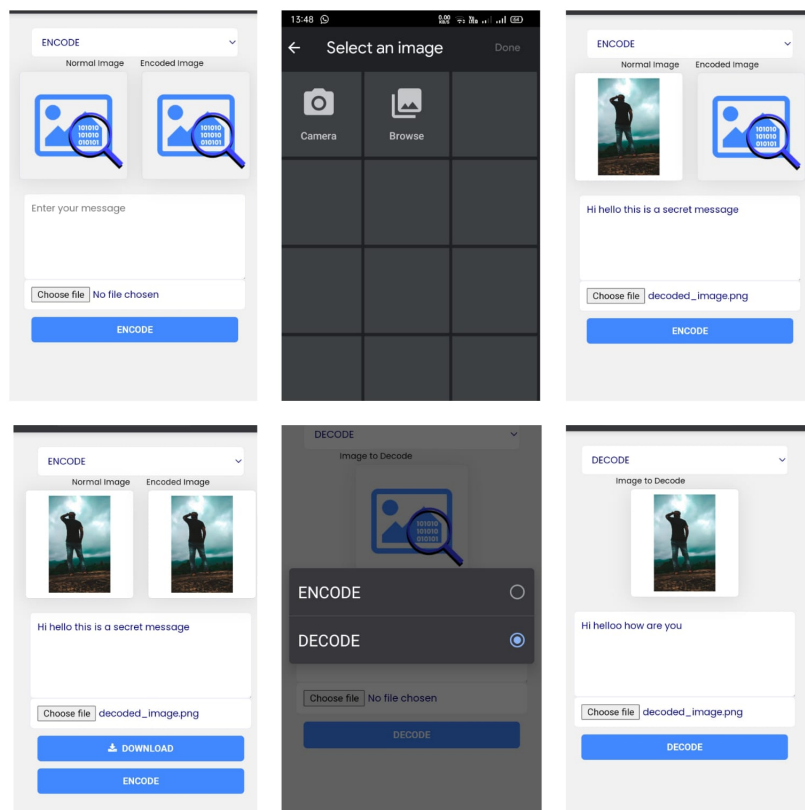
Base64 is an encoding algorithm that converts any characters, binary data, and even images or sound files into a readable string, which can be saved or transported over the network without data loss. The characters generated from Base64 encoding consist of Latin letters, digits, plus, and slash. Base64 is most commonly used as a MIME (Multipurpose Internet Mail Extensions) transfer encoding for email.

After converting it into the base64, the converted file and the message in string format will be handled by the steghide program which is used to encode the file. .

The steghide program will run and as the output it will give a string. By using javascript that string will be converted into an image file. That image file will be exactly the same like normal image file before encoding. That encoded image file can be downloaded using the download button. It will be saved inside the internal memory..

DECODING

For decoding, the encoded image file should be uploaded. After selecting the previously encoded file, it will be converted into the base64 format. It will be a string-like format. Then this string will be handed over to the stighide program for decoding. The decode function will run and the decoded message will be displayed in the output column of the website.

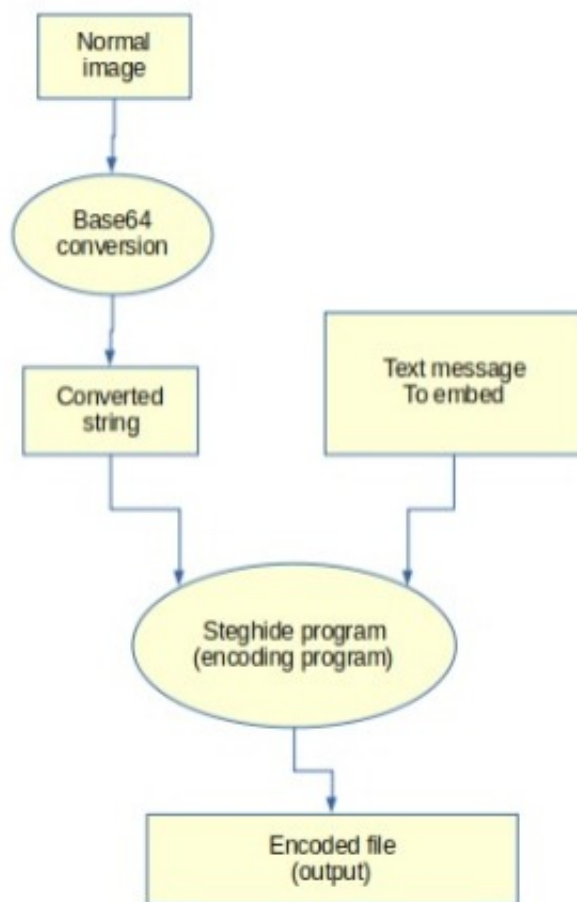


working model

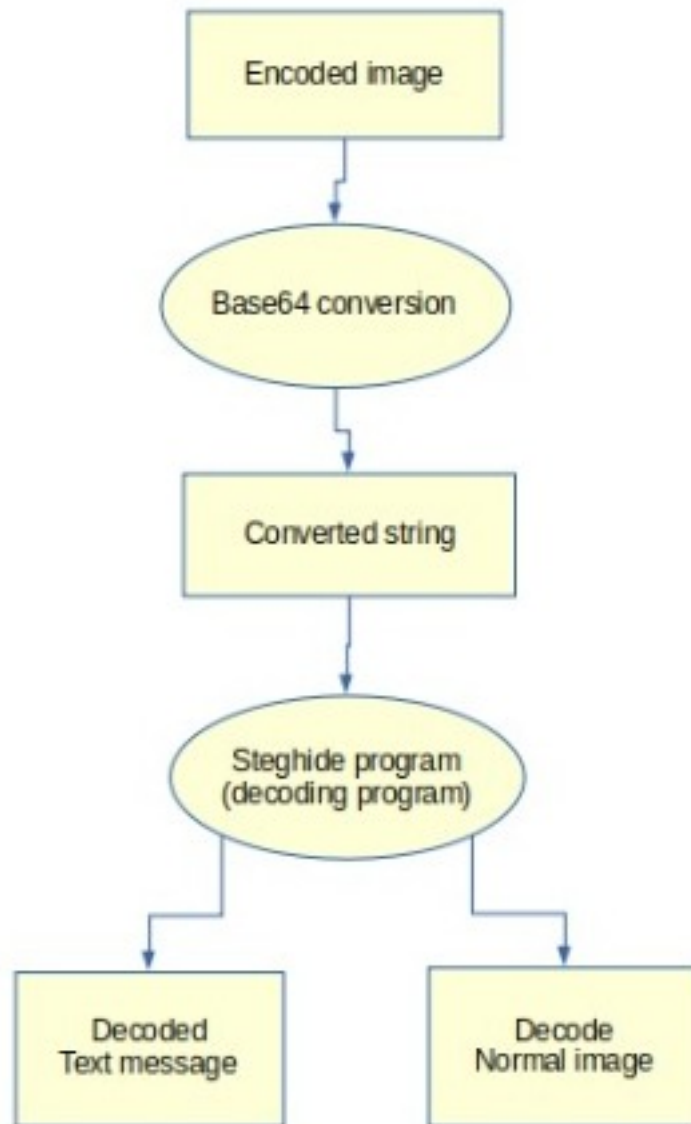
Chapter 5

BLOCK DIAGRAM

ENCODING



DECODING



Chapter 6

CODES

6.1 Html

```
<html>

  <head>

    <title>Image Steganography</title>

    <meta charset="utf-8">

    <meta name="theme-color" content="f0f0f0">

    <meta name="mobile-web-app-capable" content="yes">

    <meta name="viewport"

      content="width=device-width,initial-scale=1.0">

    <link href="css/styles.css" rel="stylesheet">

    <link href="images/image_search.png" rel = "icon" >

    <link rel="preconnect" href="https://fonts.gstatic.com">

    <link rel="stylesheet"

      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/
4.7.0/css/font-awesome.min.css">
```

```
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600display;=swap" rel="stylesheet">

</head>

<body>

<div class="loader" id="pan_load" style = "display : none;" >

<div class="load">

</div>

</div>

<div class="pan-main" id="pan_main" >

<form>

<select class="txt">

<option value="encode">ENCODE</option>

<option value="decode">DECODE</option>

</select>

<div class="pan-image">

<h id="head_normal" > NormalImage < /h >

<h id="head_coded" > EncodedImage < /h >

<img id="image_normal"src = "images/image_search.png"
alt = "preview" >

<img id="image_coded"src = "images/image_search.png"

alt="preview">

</div>
```

```
<textarea id="txt_message" class = "txt" cols = "4"
rows="5" placeholder="Enter your message"
required></textarea>

<input id="file_image" class = "txt" type = "file"
accept="image/*" required>

<button style="display:none;" type="button"
class="btn-solid" id="btn_down" >< i class = "fa
fa-download"></i>nbsp; DOWNLOAD</button>

<button class="btn-solid" id="btn_ub"
type="submit">ENCODE</button>

</form>

</div>

<script src="js/steganography.js"></script>

<script src="https://code.jquery.com/jquery-3.6.0.js"
integrity="sha256-H+K7U5CnXl1h5ywQfKtSj8PCmoN9aaq30gDh27Xc0jk="
crossorigin="anonymous"></script>

<script type="text/javascript"
src="js/scripts.js"></script>

<script async src="https://drv.tw/inc/wd.js"></script></body>

</html>
```

6.2 scripts.js

```
(document).ready(function()  
    var imageData;  
  
    ("form_select").change(function()  
  
        switch(("form_select option:selected").val())  
  
        case "decode":  
  
            ("imagecoded").hide("fast");  
  
            ("imagenormal").prop("src", "images/imagesearch.png");  
  
            ("headcoded").hide("fast");  
  
            ("headnormal").html("ImagetoDecode");  
  
            ("btnsub").html("DECODE");  
  
            ("btndown").hide("fast");  
  
            ("fileimage").val(null);  
  
            ("txtmessage").prop("required", false)  
  
            .prop("placeholder", "Decoded message will show here")  
  
            .prop("readonly", true)  
  
            .prop("value", "");  
  
            break;  
  
        case "encode":  
  
            ("imagenormal").prop("src", "images/imagesearch.png");  
  
            ("imagecoded").show("fast").prop("src", "images/imagesearch.png");  
  
            ("headcoded").show("fast");
```



```
("btnsub").html("ENCODE");

("headnormal").html("NormalImage");

("txtmessage").prop("required", true)

.prop("placeholder", "Enter your message")

.prop("readonly", false)

.prop("value", "");

("fileimage").val(null);

break;

);

("fileimage").change(function(e)
var reader=new FileReader();

reader.onloadend=function()

imageData=reader.result;

("imagenormal").attr("src", reader.result);

("btndown").hide("fast");

;

reader.readAsDataURL(e.target.files[0]);

);

("form").submit(function(e)

e.preventDefault();

switch(("form select option:selected").val())

case "encode":
```

```
("imagecoded").prop("src", steg.encode(("txtmessage").val(), imageData));

(btndown).show("fast");

//closeLoad();

break;

case "decode":

var data=steg.decode(imageData);

(txtmessage).val(data == ""?"nodata found

:-(":data.includes("è®¶i_i")?data.replace("è®¶i_i",""):data);

break;

);

(btndown).click(function()
try

var a=document.createElement("a");

a.setAttribute('href',("imagecoded").attr("src"));

a.setAttribute('download',"decodediimage.png");

(panmain).append(a);

a.style.display='none';

a.click();

(panmain).remove(a);

catch(e)

alert(e);

);
```

```
function showLoad()  
  
("panload").show("fast");  
  
function closeLoad()  
  
("panload").hide("fast");  
  
);
```

6.3 Styles.css

```
*  
  
margin:0;  
  
padding:0;  
  
body  
  
box-sizing:border-box;  
  
font-family:'Poppins',sans-serif;  
  
background:f1f1f1;  
  
.loader  
  
width:100  
  
height:100  
  
position:fixed;  
  
top:0;  
  
left:0;  
  
background:rgba(0,0,0,0.6);  
  
backdrop-filter:blur(5px);  
  
display:flex;  
  
z-index:10;  
  
align-items:center;  
  
justify-content:center;  
  
.load
```

```
width:100px;

height:100px;

border-bottom:16px solid f1f1f1;

border-right:16px solid 000080;

border-left:16px solid 000080;

border-top:16px solid f1f1f1;

border-radius:50

form

background:f1f1f1;

padding:10px;

user-select:none;

display:block;

margin:10px auto;

width:70

.txt

font-family:'Poppins',sans-serif;

color:000080;

padding:10px;

background:white;

width:90

margin:0 auto;
```

```
display:block;

outline:none;

border:0.5px solid rgba(0,0,180,0.1);

border-radius:5px;

.btn-solid

border:none;

outline:none;

border-radius:5px;

color:fff;

background:4188ff;

padding:10px;

width:90

margin:10px auto;

transition:0.2s;

display:block;

font-weight:bold;

.btn-solid:active

box-shadow:0 2px 10px rgba(0,0,0,0.5);

transform:scale(0.9);

.pan-image

display:flex;
```

```
justify-content:space-around;

position:relative;

padding:10px;

.pan-image img

width:50

flex:1 0.05 1;

margin:10px;

border-radius:5px;

border:0.5px solid rgba(0,0,180,0.1);

box-shadow:0 2px 30px rgba(0,0,0,0.1);

align-self:center;

.pan-image h

position:absolute;

top:0;

font-size:0.7rem;

.pan-image h:nth-child(1)

left:20

.pan-image h:nth-child(2)

right:20

@media screen and (max-width:700px)

form
```

Chapter 7

CONCLUSION

The aim of this project is to ensure safety and security while file transferring. The main processes happening here are encoding and decoding. Once the encoding process is finished the embedded secret message cannot be seen by second person unless it is decoded. So, through this process top secret datas can be send without any trace. The encoded image will not have any physical change. It will be like same as the normal picture. So in cyber security Image steganography have a major impact in ensuring safety.

Chapter 8

REFERENCE

www.elmah.io/tools/base64-image-encoder

www.steghide.sourceforge.net

www.google.com

www.w3schools.com

www.github.com