

Topic	Steganography	
Class Description	Students will be able to understand about steganography and they will learn how to hide data in image	
Class	C-228	
Class time	45 mins	
Goal	<ul style="list-style-type: none"> Understand about Steganography Hide data in image Decrypt data from image 	
Resources Required	<ul style="list-style-type: none"> Teacher Resources: <ul style="list-style-type: none"> Laptop with internet connectivity Earphones with mic Notebook and pen Visual Studio Code Student Resources: <ul style="list-style-type: none"> Laptop with internet connectivity Earphones with mic Notebook and pen Visual Studio Code 	
Class structure	Warm-Up Teacher - led Activity 1 Student - led Activity 1 Wrap-Up	10 mins 10 mins 20 mins 5 mins
WARM UP SESSION - 10mins		
Teacher Action		Student Action

<p>Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today?</p> <p>Okay, so you remember what we did in the last session</p> <p>Great!</p> <p>Any doubts from last session?</p> <p><i>The teacher clarifies doubts (if any)</i></p>	<p>ESR: Hi, thanks, yes, I am excited about it!</p>
Q&A Session	
Question	Answer
<p>What do you mean by encrypted data?</p> <p>A. Data in unreadable and secure form B. Data is in readable form C. Plain data D. None of the above</p>	<p>A</p>
<p>Why do we use netsh command?</p> <p>A. To get passwords of computer B. To get passwords of Locker C. To get passwords of Wi-Fi D. None of the above</p>	<p>C</p>
TEACHER-LED ACTIVITY - 10mins	
Teacher Initiates Screen Share	
<p style="text-align: center;"><u>ACTIVITY</u></p> <ul style="list-style-type: none"> • Learn about steganography 	

- **ASCII codes**
- **Save data inside image**

Teacher Action	Student Action
<p>Our last class we learned about cryptography, we saw how we can change our messages into unreadable form using encryption, and then how we can change them again using decryption.</p> <p>Have you ever played hide and seek?</p> <p>Great!</p> <p>That's my favourite game too!</p> <p>Apart from this have you ever hidden your secret message using an invisible link?</p> <p>You know what I remember doing as a kid was writing secret messages to friends using invisible ink. Sometimes if an invisible link is not available even lemon juice and iron do the same thing, so I used the tools my mom unwittingly provided: lemon juice and an iron. My secret message was written on a piece of paper and I told my friend to burn the lemon juice a bit with an iron or a match - some sort of heat source - to turn it brown. This revealed the secret message</p>	<p>ESR Yes!</p> <p>ESR Yes!</p>

<p>I wanted to share.</p> <p>Have you ever done that?</p> <p>If yes, then you can't believe we tried steganography!</p> <p>Now you will ask what is steganography?</p> <p>Any ideas?</p> <p>What Is Steganography?</p> <p>Steganography is the practice of hiding a secret message within (or on top of) something that is not secret. The message can be anything you want. Steganography involves embedding a secret piece of text inside of a picture or by hiding a secret message or script within a Word or Excel document.</p> <p>The main motive of steganography is to hide the intended information within any file, usually an image, an audio file, or a video, without actually altering the external appearance of the file.</p> <p>In today's session we are going to hide our secret message</p>	<p>ESR Yes!</p> <p>ESR NO!</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------

under image.

Now, you must be wondering how this will work,

Let's learn the technique for the same

- First, Again for this we need to keep track of ASCII codes.
- Second about image, Upload one image but it should be in png format, As we know Images are composed of digital data (pixels), which describe what's inside the picture, such as the color of each pixel. Every image is composed of pixels, and each pixel contains three values (red, green, blue).
- Third we must know about LSB (Least Significant Bit) LSB in which messages are hidden inside an image by replacing each pixel's least significant bit with the bits of the message to be hidden
- Fourth, use encoding and decoding steganographic secret messages into a cover image file by using techniques like least significant bit encoding. The resulting stego image looks very similar to your cover image file, with no visible changes. This completes

encoding. To retrieve the secret message, we will use a decoding process

Let's start coding

Import all required libraries

Boilerplate Code starts from here

- **from PIL import image** The Python Imaging Library Pillow or **PIL** adds image processing capabilities to the Python interpreter.
- Create a function to convert any type of data into binary, we'll use this to convert secret data and pixel values in the encoding and decoding phase.
- Create one function **genData()**, pass the argument data. This function will take the secret message and convert that message into **ASCII** codes using **ord()** function
- Create new array **newd = []**
- Using for loop check the index of all characters and convert into **ASCII** using **ord** and then using **append()** method add into **newd** array and then

return after conversion	
Click on Teacher Activity 1	Student clones the code from Student Activity 2
<pre> from PIL import Image def genData(data): # list of binary codes # of given data newd = [] for i in data: newd.append(format(ord(i), '08b')) return newd </pre>	
<p>Make the Main() Function</p> <p>For the main function, we ask the user which function they would like to perform — Encode or Decode.</p> <ul style="list-style-type: none"> • Take input from the user and save that in variable a • If user will press a==1, then do encode • elif user will press a==2 then do decode • If user press anything else, then raise exception • Call the main() function 	

```
def main():
    a = int(input("::: Welcome to Steganography :::\n"
                  "1. Encode\n2. Decode\n"))

    if (a == 1):
        encode()

    elif (a == 2):
        print("Decoded Word : " + decode())
    else:
        raise Exception("Enter correct input")

if __name__ == '__main__':
    # Calling main function
    main()
```

Now it's time to take the input image and secret message from user

Create function that takes the input image name and secret message from the user and calls **encode_enc ()** function to encode it

- Create **img** variable and save input image in this variable
- Using **open()** method read the image using **"r"**
- Create **data** variable and save hidden message in this variable
- Check if length of data is zero then raise exception and print data is empty
- Create **new_img_name** variable and that will save new image name
- Call the **main()** function


```
def encode():
    img = input("Enter image name(with extension) : ")
    image = Image.open(img, 'r')

    data = input("Enter data to be encoded : ")
    if (len(data) == 0):
        raise ValueError('Data is empty')

    newimg = image.copy()
    encode_enc(newimg, data)

    new_img_name = input("Enter the name of new image(with extension) : ")
    newimg.save(new_img_name, str(new_img_name.split(".")[1].upper()))

    main()
```

Teacher start writing code from here

Now here comes the main part how the data will be saved and how we can alter our pixels to save the data inside the image

- Each pixel contains three values which are Red, Green, Blue, these values range from **0 to 255**, in other words, they are 8-bit values.
- Let's take an example of how this technique works, suppose you want to hide the message "" into a **4x4** image which has the following pixel values:

**[(225, 12, 99), (155, 2, 50), (99, 51, 15), (15, 55, 22),
(155, 61, 87), (63, 30, 17), (1, 55, 19), (99, 81, 66),
(219, 77, 91), (69, 39, 50), (18, 20, 33), (25, 54, 190)]**

- Suppose you want to save secret message **"HI"**, Using the **ASCII Table**, we can convert the secret message into decimal values and then into binary:
 - **H: 0110100**
 - **I : 0110101**

- Now, we iterate over the pixel values one by one, after converting them to binary, we replace each least significant bit with that message bits sequentially (e.g 225 is 11100001, we replace the last bit, the bit in the right (1) with the first data bit (0) and so on). This will only modify the pixel values by +1 or -1 which is not noticeable at all.

Now Create a function to hide secret message into the image by altering the LSB

- For each character in the data, its ASCII value is taken and converted into 8-bit binary
- Three pixels are read at a time having a total of $3 \times 3 = 9$ RGB values. The first eight RGB values are used to store one character that is converted into an 8-bit binary.
- The corresponding RGB value and binary data are compared. ***If the binary digit is 1 then the RGB value is converted to odd and, otherwise, even.***
- The resulting pixel values after performing LSBS is as shown below:

[(224, 13, 99),(154, 3, 50),(98, 50, 15),(15, 54, 23),(154, 61, 87),(63, 30, 17),(1, 55, 19),(99, 81, 66),(219, 77, 91),(69, 39, 50),(18, 200, 33),(25, 54, 190)]

```
def modPix(pix, data):

    datalist = genData(data)
    lendata = len(datalist)
    imdata = iter(pix)

    for i in range(lendata):
        pix = [value for value in imdata.__next__():3] +
               imdata.__next__():3] +
               imdata.__next__():3]

        for j in range(0, 8):
            if (datalist[i][j] == '0' and pix[j] % 2 != 0):
                pix[j] -= 1

            elif (datalist[i][j] == '1' and pix[j] % 2 == 0):
                if (pix[j] != 0):
                    pix[j] -= 1
                else:
                    pix[j] += 1
                # pix[j] -= 1

        if (i == lendata - 1):
            if (pix[-1] % 2 == 0):
                if (pix[-1] != 0):
                    pix[-1] -= 1
                else:
                    pix[-1] += 1

            else:
                if (pix[-1] % 2 != 0):
                    pix[-1] -= 1

        pix = tuple(pix)
        yield pix[0:3]
        yield pix[3:6]
        yield pix[6:9]
```

Teacher download the boilerplate code from [Teacher Activity 2](#)

Student download the repository from [Student Activity 2](#)

Note :- Code to create a client is already provided in the boilerplate code. Explain the code to the Student.

Teacher Stops Screen Share

STUDENT-LED ACTIVITY - 20 mins

- Ask the student to press the ESC key to come back to the panel.
- Guide the student to start Screen Share.
- The teacher gets into Full Screen.

ACTIVITY	
<ul style="list-style-type: none"> ● Create the function for Encryption ● Create the function for Decryption ● 	
Teacher Action	Student Action
<p><i>Guide the student to get the boilerplate code from Student Activity 2</i></p>	<p><i>Student clones the code from Student Activity2</i></p>
<p>Now we know 8 bit for H but what about the last pixel left.</p> <p>The ninth value determines if more pixels should be read or not. If there is more data to be read, i.e. encoded or decoded, then the ninth pixel changes to even. Otherwise, if we want to stop reading pixels further, then make it odd.</p> <p>Create a function encode_enc()</p> <ul style="list-style-type: none"> ● Get the size of the image() ● Save the coordinates ● Using if condition check if more data is to be read ● If yes then make it even ● If no make it odd 	
<pre>def encode_enc(newimg, data): w = newimg.size[0] (x, y) = (0, 0) for pixel in modPix(newimg.getdata(), data): # Putting modified pixels in the new image newimg.putpixel((x, y), pixel) if (x == w - 1): x = 0 y += 1 else: x += 1</pre>	

Create a function to decode the hidden message from the stego image.

- Create variable `img` that will ask the stego image
- Using `open ()` and `"r"`
- Create one empty variable `data`
- The Python `iter()` function returns an iterator for the given object. The `iter()` function creates an object which can be iterated one element at a time.
- `getdata()` returns the contents of this image as a sequence object containing pixel values.
- It will take the one pixel value at time and then move to next three one by one
-
- Check the 8 bits , if it is divisible by 2 then return `binstr "0"`
- If not then make it `"1"`
- Convert ascii data into character again i.e readable form using `chr()`
- And then return the data

```
def decode():
    img = input("Enter image name(with extension) : ")
    image = Image.open(img, 'r')

    data = ''
    imgdata = iter(image.getdata())

    while (True):
        pixels = [value for value in imgdata.__next__()[:3] +
                  imgdata.__next__()[:3] +
                  imgdata.__next__()[:3]]

        binstr = ''

        for i in pixels[:8]:
            if (i % 2 == 0):
                binstr += '0'
            else:
                binstr += '1'




        data += chr(int(binstr, 2))
        if (pixels[-1] % 2 != 0):
            return data
```

Teacher Guides Student to Stop Screen Share

WRAP UP SESSION - 5 Mins

Quiz time - Click on in-class quiz

Question	Answer
What is steganography? A. To change a message into unreadable form B. To change the message into readable form C. To Provide the correct message D. To hide a secret message into something which is not hidden	D
What is LSB? A. Low Significant Bit B. Least Significant Bit	B

C. Low Significant Byte D. Low Signal Bit	
Why do we need imports from PIL? A. PIL helps in enabling image processing capabilities to Python Interpreter B. PIL allows you to load images and send images. C. PIL allows you to open the images. D. PIL offers a collection of images	A
End the quiz panel	
<p style="text-align: center;"><u>FEEDBACK</u></p> <ul style="list-style-type: none"> • Appreciate the students for their efforts in the class. • Ask the student to make notes for the reflection journal along with the code they wrote in today's class. 	
<p style="text-align: center;">Teacher Action</p>	<p style="text-align: center;">Student Action</p>
You get Hats off for your excellent work! In the next class	<p><i>Make sure you have given at least 2 Hats Off during the class for:</i></p> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="background-color: #0072bc; color: white; padding: 5px; text-align: center;">Creatively Solved Activities</div> <div style="margin: 0 10px;">+10</div>  </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="background-color: #0072bc; color: white; padding: 5px; text-align: center;">Great Question</div> <div style="margin: 0 10px;">+10</div>  </div> <div style="display: flex; align-items: center;"> <div style="background-color: #0072bc; color: white; padding: 5px; text-align: center;">Strong Concentration</div> <div style="margin: 0 10px;">+10</div>  </div> </div>
<p>Project Discussion</p>	

Teacher Clicks

✕ End Class

ADDITIONAL ACTIVITIES

Additional Activities

Encourage the student to write reflection notes in their reflection journal using markdown.

Use these as guiding questions:

- What happened today?
 - Describe what happened.
 - The code I wrote.
- How did I feel after the class?
- What have I learned about programming and developing games?
- What aspects of the class helped me? What did I find difficult?

The student uses the markdown editor to write her/his reflections in the reflection journal.

ACTIVITY LINKS

Activity Name	Description	Link
Teacher Activity1	ASCII Table	https://www.asciitable.com/
Teacher Activity 2	Teacher Boilerplate Code	https://github.com/pro-whitehatjr/Pro-C228-TeacherBoilerPlate

© 2021 - WhiteHat Education Technology Private Limited.

Note: This document is the original copyright of WhiteHat Education Technology Private Limited.

Please don't share, download or copy this file without permission.

Teacher Activity 2	Reference Code	https://github.com/pro-whitehatjr/Pro-C228_ReferenceCode
Student Activity 1	ASCII Table	https://www.asciitable.com/
Student Activity 2	Boilerplate Code	https://github.com/pro-whitehatjr/Pro-C228-StudentBoilerPlate