CAPSTONE PROJECT

HIDING A TEXT INSIDE AN IMG USING STEGANOGRAPHY (WITH WEBSITE)

Presented By:

Name :Arshad Pasha

College Name : Seshadripuram Degree College ,Mysore

Department :BCA (Computer Application)

edunet foundation

OUTLINE

- **Problem Statement** (Should not include solution)
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment (Step by Step Procedure)**
- **Result**
- **Conclusion**
- **Future Scope(Optonal)**
- **References**

PROBLEM STATEMENT

- The project aims to enhance secure communication by hiding secret messages within images using steganography. Traditional encryption can draw attention, while steganography conceals the existence of the message. The goal is to create a user-friendly web app for encrypting and decrypting messages. The solution should be accessible and robust. This project demonstrates practical cybersecurity applications.

SYSTEM APPROACH

The system is developed as a web application using Python and Flask for the backend, and HTML, CSS, and JavaScript for the frontend. The approach focuses on modularity, security, and ease of use. The backend handles image processing and steganography, while the frontend provides an intuitive interface for users. The application is deployed on Render, a cloud platform, ensuring accessibility and scalability.

SYSTEM APPROACH

- **System requirements:**

  - Python 3.8+

  - Web browser (Chrome, Firefox, etc.)

  - Internet connection

- **Libraries required to build the model:**

  Flask,- Pillow , numpy ,
  opencv- python ,matplotlib ,
  gunicorn (for deployment) .

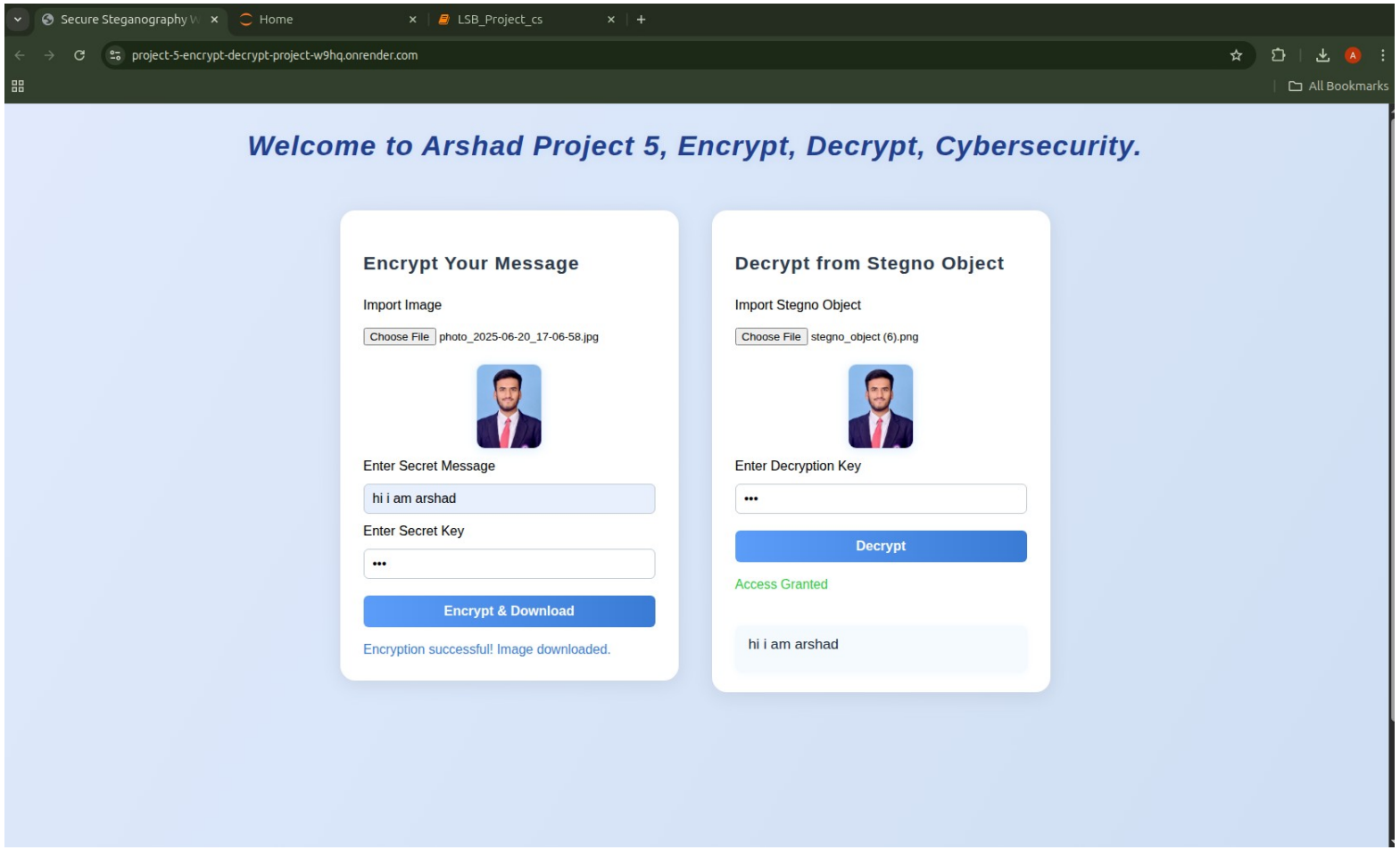
ALGORITHM & DEPLOYMENT

Step-by-step procedure:

- **1. User uploads a cover image and enters a secret message and key.**
- **2. The backend encrypts the message and hides it in the image using LSB (Least Significant Bit) steganography.**
- **3. The processed image (stegano object) is returned for download.**
- **4. For decryption, the user uploads the stegano image and enters the key.**
- **5. The backend extracts and decrypts the hidden message.**
- **6. The result is displayed to the user.**
- **7. The app is deployed on Render using Gunicorn as the WSGI server.**

RESULT

- - The web app allows users to securely hide and retrieve messages in images.
- - User interface is clean and easy to use



RESULT

Encrypt Your Message

Import Image

Choose File photo_2025-06-20_17-06-58.jpg

Enter Secret Message

hi i am arshad

Enter Secret Key

•••

Encrypt & Download

Encryption successful! Image downloaded.

Decrypt from Stegno Object

Import Stegno Object

Choose File stegno_object (6).png

Enter Decryption Key

•••

Decrypt

Access Granted

hi i am arshad

RESULT

- If Worng Key Input

```
#Encrypt using pixel modelfication
x_enc=x.copy()
# print if you needed  x_enc
#c,r,colorpanle
n=0  #num of rows
m=0 # num of clooums
z=0 #colour panel
l=len(text)
kl=0  # index of the key

for i in range(l):
    char_val=d[text[i]] ^ d[key[kl]]
    for bit_pos in range(8):
        bit=(char_val >> (7-bit_pos)) & 1   # imp .....very imp
        org_val = x_enc[n,m,z]
        x_enc[n,m,z]=(org_val & 0b11111110)| bit  # imp  0b11111110
        print(f"Embedding bit {bit} of '{text[i]}' at ({n},{m},{z}) original={org_val} new={x_enc[n, m, z]}")

        z=(z+1)%3
        if z==0:
            m=m+1
            if m==x_enc.shape[1]:
                m=0
                n=n+1
    kl=(kl+1)%len(key)
#ok
print(d[text[1]] ^ d[key[1]])
```

```
Embedding bit 1 of 'P' at (0,43,0) original=46 new=47
Embedding bit 1 of 'P' at (0,43,1) original=20 new=21
Embedding bit 0 of 'P' at (0,43,2) original=9 new=8
Embedding bit 0 of 'P' at (0,44,0) original=46 new=46
Embedding bit 0 of 'P' at (0,44,1) original=20 new=20
Embedding bit 1 of 'P' at (0,44,2) original=9 new=9
Embedding bit 0 of 'P' at (0,45,0) original=46 new=46
```

[8]:
```
#ASCII COnversion
d={chr(i):i for i in range(255)}
c={i:chr(i) for i in range(255)}
```

[10]:
```
# message and Encryption key Inputing takeing for User
text=input("Enter Your Text :")
key=input("Enter The Key :")
```

```
Enter Your Text : hi, i am Arshad Pasha
Enter The Key : 123
```

[11]:
```
#Loaning the img
image_path = r"new.webp"
x=cv2.imread(image_path)
x # as print ... ok arshad

xrgb = cv2.cvtColor(x, cv2.COLOR_BGR2RGB)
plt.title("This Image is BGR ..... which defult done by cv2")
plt.imshow(x)
plt.axis('off')
plt.show()

plt.title("This Image is RGB Orginal img ..... which defult done by cv2")

plt.imshow(xrgb)
plt.axis('off')
plt.show()
```

[12]:
```
#Encrypt using pixel modelfication
```

RESULT

```
Reading bit 0 from (0,53,1)
Reading bit 1 from (0,53,2)
Reading bit 0 from (0,54,0)
Reading bit 1 from (0,54,1)
Reading bit 0 from (0,54,2)
Reading bit 0 from (0,55,0)
Reading bit 1 from (0,55,1)
Reading bit 0 from (0,55,2)
Decrypted Byte : 82 XOR 51 = 97 -> 'a'
Decrpyred :
 hi, i am Arshad Pasha
```

]:

This is the image which has encrypted text.

```
[[[ 40  19  10]
  [ 41  19  10]
  [ 40  19  10]
  ...
  [ 36  16   8]
  [ 36  16   8]
  [ 36  16   8]]

 [[ 41  18  11]
  [ 41  18  11]
  [ 41  18  11]
  ...
```

GitHub Code Link : https://github.com/pashaarshad/Project-5-Encrypt--Decrypt--Project--Cybersecurity.git

Video Demo Of this Project Link : https://youtu.be/5dO9dxJoHmc?si=SbGLcU2Ztq0w-O4u

edunet foundation

CONCLUSION

- The Secure Steganography Web App provides an effective solution for secure communication by combining steganography and encryption. The project demonstrates the feasibility of hiding sensitive information in images and retrieving it securely. Challenges included handling various image formats and ensuring cross-platform compatibility. Future improvements could include support for more file types and advanced encryption algorithms.

FUTURE SCOPE(OPTIONAL)

- - Add support for audio and video steganography.
- - Implement user authentication and logging.
- - Integrate advanced cryptographic techniques.
- - Develop a mobile app version.
- And More ...

REFERENCES

- https://flask.palletsprojects.com/
- - https://pillow.readthedocs.io/
- - https://numpy.org/
- - https://opencv.org/
- - https://matplotlib.org/
- - Research papers and articles on steganography and cybersecurity.

THANK YOU

Special thanks to **Nanthini Mam** for her extraordinary teaching—deeply grateful.

edunet foundation