Security Project

Steganography Chat

Team Members

Nada Nasser 34-4993

Salma Ossama 34-4259

Fatma Hossam 34-2059

In this project, we have implemented the steganography chat which is a chat application based on the steganography idea; but what is **Steganography?!**

Steganography is simply the art of hiding data within data. For example, you could hide some text within the pixels of an image or, within the pixels of a given frame of a video. You could even do the same with audio or any other kind of file.

In our project, we hide the text within an image.

We have implemented a decentralized P2P chat application which applies steganographic techniques to relay secret messages back and forth between multiple users using Python, it has a lobby chat room, in which the user can broadcast a message to all the other available users; additionally it supports the private chat in which the user can send a message to only a specific user.

The features of the project are:

Steganography: it was implemented when the client sends a message to the server then the server hide the message(text) into a specific image using the “lsb” library from “stegano” and then sends the image that contains the hidden text to the other client where the other client uses the method “reveal” from “stegano” to extract the hidden text from the image.

Authentication: we implemented it by encrypting the password entered by the user using the “Fernet” library from ”cryptography” by generating a key from the password then using it to encrypt the password.

Capacity: we have calculated the capacity by dividing the number of bits that have been changed from the original image used to hide the message using the “Stegano lsb” library by the total number of bits of the image.

Fidelity: we measured the Peak Signal To Noise Ratio using pre-implemented library “PSNR” from sewar python.

Access Control: we implemented it by giving the user the option of broadcasting the message to all the users in the chat room or by sending the message to a specific user that he can choose from the list of users in the chat room.

One of the attacks that we have avoided is the man in the middle attack by using the steganography technique to exchange messages between clients because the attacker could not differentiate between a normal image and an image containing a hidden text and even if he detected that there is a hidden text in the image he would not be able to reveal the text from the image since he does not know which steganography technique was used to hide the text in the image.

We have used the following libraries:

- tkinter
- threading
- messagebox from tkinter
- * from socket
- time
- sys
- cryptography
- base64
- os
- default_backend from cryptography.hazmat.backends
- hashes from cryptography.hazmat.primitives
- PBKDF2HMAC from cryptography.hazmat.primitives.kdf.pbkdf2
- Fernet from cryptography.fernet
- lsb from stegano
- psnr from sewar.full_ref
- cv2

In the authentication feature, we have used the “Fernet” library to encrypt the password entered by the user before sending it to the server by generating a key that depends on the password entered and then encrypting the password using this key to ensure that the man in the middle attack will not be able to know the password of the user.

In the fidelity feature, we have used “psnr” function from the sewar library that calculate the peak signal to noise ration of two images which were in our case the original image and the one after hiding the text in it to ensure the fidelity.

We have applied the steganography technique, we ensured message authentication and integrity, we have used the “lsb” from “stegano” functions(hide/reveal); the hide function

was used in the server side to hide the message(text) that should be transferred to the other client within a specific image and send it to the client that will on his side use the method “reveal” to reveal the text hidden within the image sent by the server and be able to read the original text.

References:

https://github.com/TenGumis/chat-python. : This is the link to a pre-implemented chatting application that we have used to add to it the security features.

https://pypi.org/project/Stegano/?fbclid=IwAR1_XhO3H1uT_0QU3NpR_oLFdN4H-MNQbK3pqiFJsbWwocWkI2GgFAtLfck : This is the link of the “stegano” library that we have used it to hide the text within the image using their “lsb” function.

https://pypi.org/project/sewar/: This is the link of the “sewar” library that we have used it to calculate the Peak Signal to Noise Ration(PSNR) to assess the fidelity feature using their “psnr” function.

https://www.programcreek.com/python/example/98805/cryptography.fernet.Fernet This is the link of the “fernet” library that we have used it to encrypt the password entered by the client before sending it to the server to ensure the authentication feature.