Security Project Report
Steganography Chat
Team 31

Jailan Raafat	34-2732
Farah Essam	34-3222
Laila Tarek	34-0721
Reem Hatem	34-0705

Summary:

In this project we implemented a Steganography Chat. Steganography is basically a more secure type of chats in which the data that is supposed to be sent is encoded first. The encoding process is done by hiding the text or data in an image and send this image instead and in the receiver side this image is decoded in order to get the text that was originally sent by the sender. Steganography Chat provides a more secure chat as the man in the middle won't be able to see except images. Our motivation was that chatting systems can be more secure.

As for the features:

Authentication was implemented using a password generator, this password generator generates a unique password for each user. The password is generated randomly from capital and small chars, numbers and symbols. The length of this password is fixed to 5, when the application is launched a pop-up message with the generated password. The user is required to enter this password in order to be able to enter the chat room.

Capacity feature was approached by coordinating between the text or data that should be sent and the size of the images. We set a certain threshold and we arranged some images in two arrays according to the size of the image, where the small sized images are in an array and the big sized images are in another array. If the size of the message sent from one user to another is greater than this threshold, we choose an image from the array with the big sized images.

The big sized images had a lot of details in order to hide any distortion in the image. Otherwise, if the size of the message sent from one user to another is less than the threshold, we choose an image from the array with the small sized images. The threshold was chosen by trial and error and it was a very tricky part in order to be set to the suitable value. The images chosen were all small in dimension and this helped in hiding any distortion that will happen.

Fidelity was handled with two ways; the first way was to pick small dimensioned images and the second way was to pick images with a lot of details and noisy to hide the text.

The Access Control or private chat was implemented as well. The application is launched and after that the users should enter their name, password, ports to be connected, ports to listen to and IP addresses. We enter the same port numbers for the sender and receiver, and the users enter the IP address of the user that they want to chat with and vice versa. After that both users will be able to chat with each other in a private chat.

The cryptographic algorithms used in this project:

the steganography technique used to encode the text messages sent from one user to another and the messages are hidden using LSB steganography technique (least significant bit) within an image that is later decoded on the receiver side. The cryptography was carried out through images as they are better at hiding distortion than audio or video.

Additionally, audio and video are costly in terms of performance and time consumed to encode the message sent before the receiver receives it. The second algorithm used in this project to ensure security in the chatting application is the authentication with OTP (one-time password) technique, where each time a new password is randomly generated for the user and the user needs to type it to login to the app. We used this type of authentication technique because the passwords generated are secure and hard to hack or know the password as it is randomly generated from a combination of capital and small letters, numbers, and symbols.

Security Project Report
Steganography Chat
Team 31

Jailan Raafat	34-2732
Farah Essam	34-3222
Laila Tarek	34-0721
Reem Hatem	34-0705

Access control was conducted by share the use of IP addresses between to users, where each of the users writes the IP address of the other in order to start a private conversation between them.

Many researches were conducted in order to find the most suitable combination of algorithms in order to provide security for the chatting application. Authentication was considered before using some static usernames and passwords from which the user must use one combination of a username and a password in order to login to the application. However, this method of authentication was not secure or dynamic so we implemented a function called password_gen that generates a one-time password that is randomly chosen from numbers, characters and symbols. Another research was conducted in order to decide which steganography technique should be implemented in order to ensure the best security. Image Steganography was found to be the most popular method used. Textual steganography can be achieved by altering the text formatting, or by altering certain characteristics of textual elements. Audio steganography, secret message is embedded into digitized audio signal which result slight altering of binary sequence of the corresponding audio file. In result we found that Image steganography is the best and most efficient method for implementation.

Image Steganography can be achieved by several techniques, LSB (Least Significant Bit algorithm), Masking and filtering, and Algorithms and Transformation. The LSB works best when the file is longer than the message file. When applying LSB techniques to each byte of a 24 bit image, three bits can be encoded into each pixel. If the LSB pixel value of cover image C(i,j) is equal to the message bit of the secret message to be embedded, the C(I,j) is not changed. Otherwise, we set the C(I,j) to the message bit of the secret message.

Security Project Report
Steganography Chat
Team 31

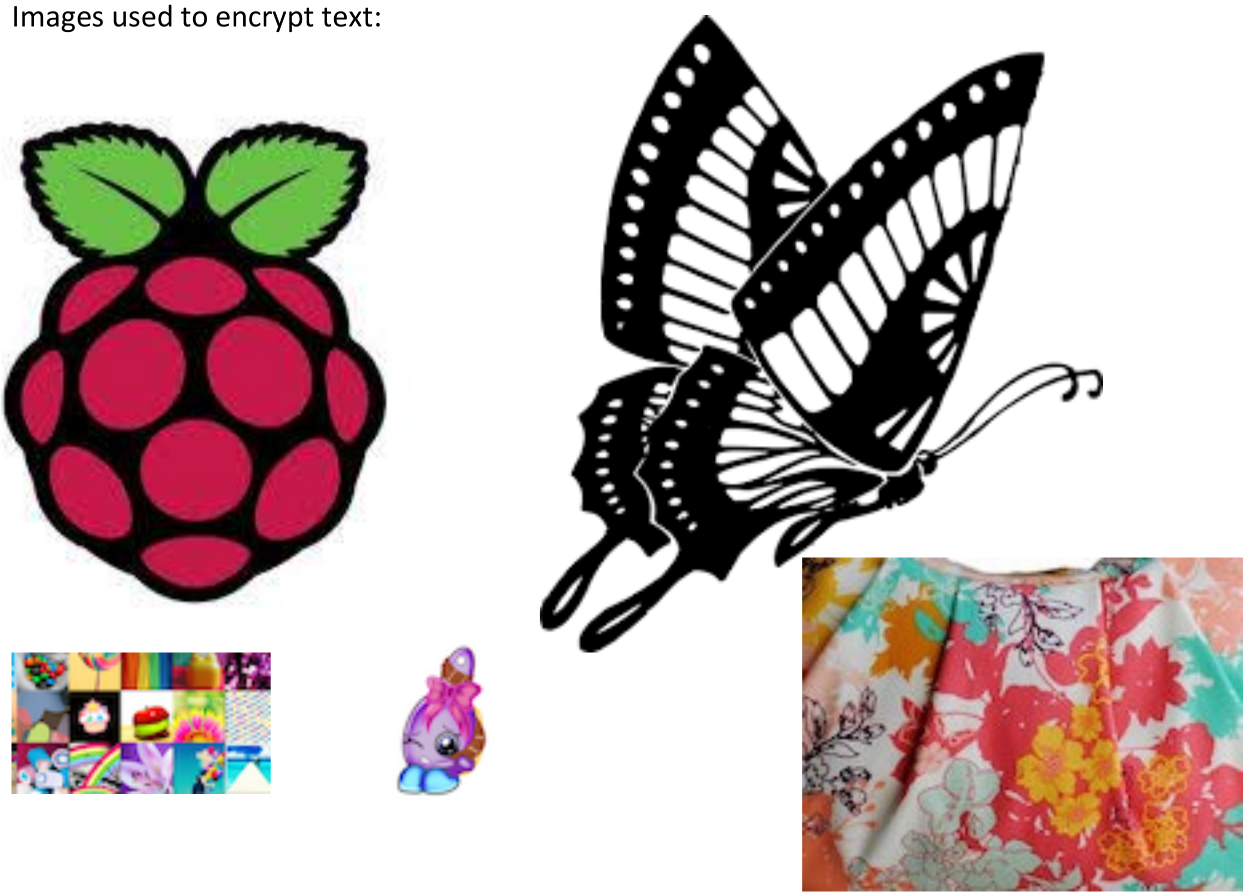
Jailan Raafat	34-2732
Farah Essam	34-3222
Laila Tarek	34-0721
Reem Hatem	34-0705

Attack Scenarios:

File only attack where the attackers can access the file and try to know if there is hidden data in it but they could not know the text encoded in it as the do not have the encoding or decoding algorithm. Another method to attack if the attacker can access both the file and the original file and by then the attacker can change the message or destroy it, this is handled as only the sender and the receiver have the encoding and decoding algorithm. Attackers can change the file format which will affect the algorithm as well, this could be fixed manually from the code depending on the file format. Attackers could also add small tweaks randomly in order to destroy the message this is handled by asking the sender to send the message again.

Screenshots:

Images used to encrypt text:



Security Project Report
Steganography Chat
Team 31

Jailan Raafat	34-2732
Farah Essam	34-3222
Laila Tarek	34-0721
Reem Hatem	34-0705

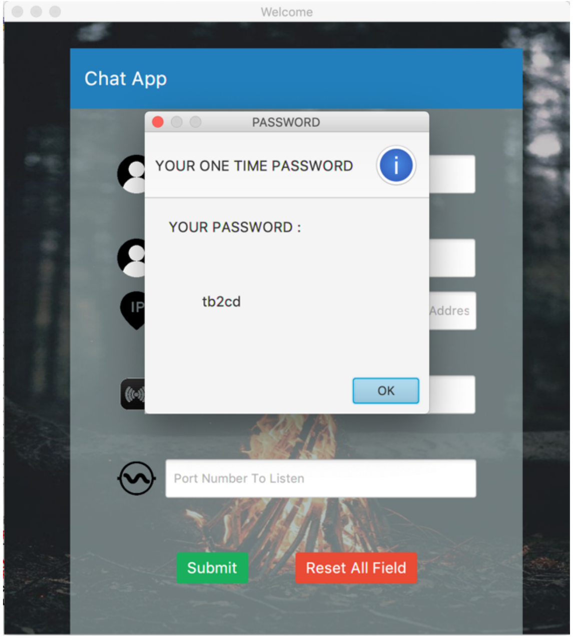
Image before and after hiding text:

Before

After

Password:

step 1



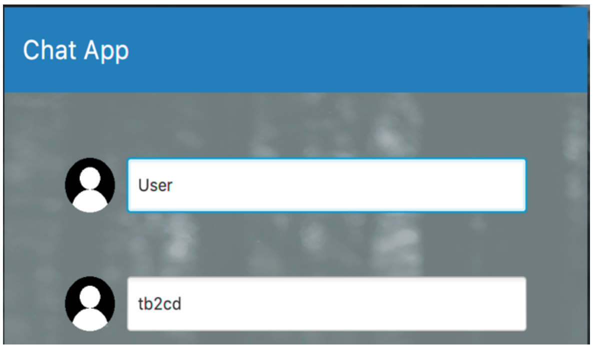
PASSWORD

YOUR ONE TIME PASSWORD

YOUR PASSWORD :

tb2cd

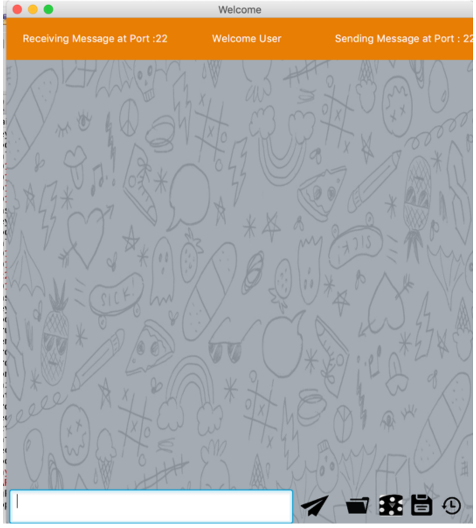
step 2



User

tb2cd

step 3



Receiving Message at Port :22

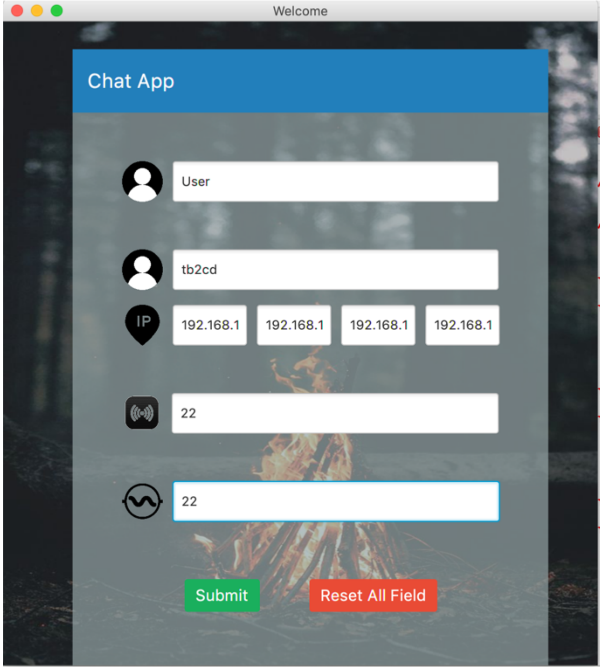
Welcome User

Sending Message at Port : 2

Security Project Report
Steganography Chat
Team 31

Jailan Raafat	34-2732
Farah Essam	34-3222
Laila Tarek	34-0721
Reem Hatem	34-0705

Entering all the data :



Welcome

Chat App

User

tb2cd

IP 192.168.1 192.168.1 192.168.1 192.168.1

22

22

References:

1. http://www.garykessler.net/library/steganography.html
2. https://github.com/Shadat-tonmoy/P2P-Chatting-And-File-Sharing-App
3. https://ccrma. edu/~eberdahl/Projects/Paranoia/ stanford.
4. https://code.google.com/p/crypto-js/
5. https://www.google.com/search?client=safari&rls=en&q=steganography&ie=UTF-8&oe=UTF-8