Project Report

Steganography Chat

TEAM 21

Andrew Akram 34-4673 T-07

Sally Habib 34-4106 T-07

Michael Adel 34-5181 T-07

Mariam Atef 34-12845 T-07

Mina William 34-0880 T-07

Summary of the Project

Please not that we've implemented only minimal GUI and features.
It's a simple java chat application. Users are able to send text and images. Users can also hide text in the image. The app manipulates the least significant bits of the pixels of the image to hide the message. The message string is first converted to a byte array. Then for each pixel, its two least significant bits are replaced by two bits of the message.
The client on the other side can then extract the least significant bits, concatenate them and obtain the string.
The chat application also had authentication implemented. However passwords were stored as plain-text in a database. We've used AES encryption to store the passwords into the database. When the user is signing up, he inputs the password as plain-text. Then the password is hashed and stored. Later when the user tries to login, the encrypted password is decrypted and compared against the input password. We've used a ready-made library that implements AES.

Chat Application Flow

The flow of the chat application starts with user registration using username and password. The initial screen shows the user registration and user login. Then a request is sent to the database to store the username and the password. The password is stored in the database hashed. Using AES encryption. After that, the user can login using his/her username and password. For user authentication, a request is send to the database containing the credentials for the user. The password is hashed using the same AES encryption. For matching usernames, If the password hashed to the same value as the stored password in the database, then the user is authenticated successfully. If the user is authenticated successfully, the chat application navigates to the chating screen. The user can get list of all the users which are connected to the server. Then for private chatting with any user, the user can click on the user button that he/she wishes to chat with him/her. In addition to that, the user can send a message to all the users that are connected to the server. In order to do that, the user send a broadcast message to the server, then the server forwards the message to all the user connected to the server. The chat application offers many features such as the message-read feature which help the user to identify which messages that he/she has read before and which messages that are recently received. The unseen received message appears first in the list of the messages. This feature saves time for the user to check only for the unviewed incoming message.

Motivation

Nowadays chatting applications becomes as an integral part of our daily digital life. As it process arange of very sensitive personal information on it everyday and without security it can be easy for many hackers to access. In these days chat applications and many communication systems are increasing privacy and security using steganography to help their users to connect and communicate with others without having to worry about hackers or the privacy of their messages or informations. As steganography is considered to be the art of hiding data so it does not even appear to exist. So it was very interesting to us to use this way in a chat application and discover the power of security .

Storing

In order to implement the app, we first obtained a working java chat application. Then we added to the user the feature of sending an image. The image is encoded to a string and sent through the messages socket. Then we've implemented two functions. The encryption function converts the string to a byte array, loops over the pixels of the image, and replaces each two least significant bits with the bits of the message. The decryption function loops over the pixels of the image, extracts the two least significant bits, concatenates all the bits together and converts them into a string.

Design Choices

In order to inject the message into the image, we've only used a simple loop that replaces the bits. We've also added a special sequence of characters to indicate the end of the secret message. We chose to replace the least significant bits because they have very little impact on the image. The message is hidden in the pixels while they remain almost unchanged to the eye.

In the project we are trying to hide a text message in some form of media. We chose the media type to be images. For any other regular user, or a man in the middle, he would only see an image. The message can only be obtained by inspecting the bits of the file.

Comparison:

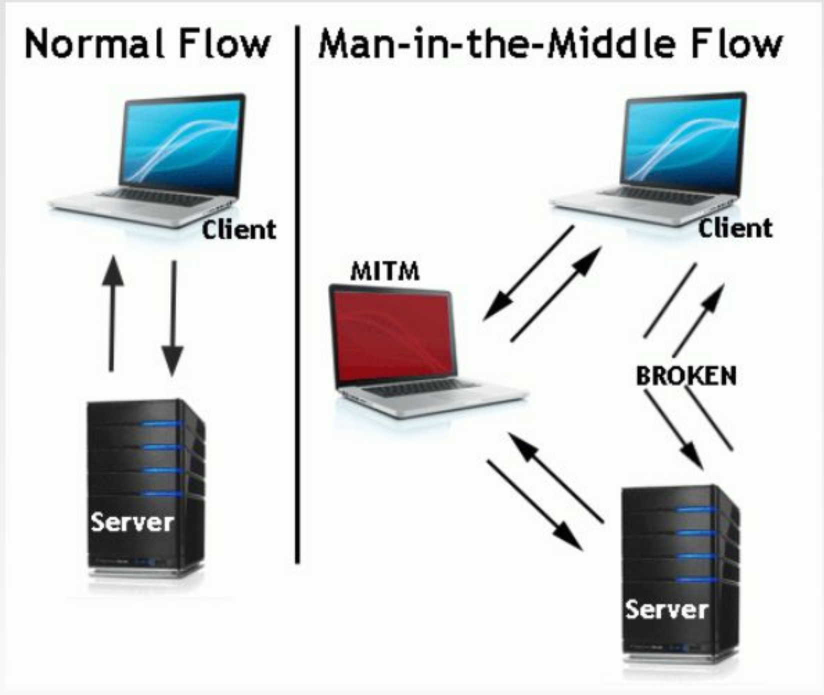
One method that came to mind was to replace whole pixels by bits of the message. We thought this method would have many drawbacks. First, it would add very clear noise to the image similar to salt and pepper noise. Also the locations of the pixels need to be well distributed, otherwise the characters would appear as lines in the image. Since the pixels need to be distributed, their locations would be dependent on the size of the image.

We can also compare the technique we used to hiding information in video files. If we assume that a picture is 500 x 500 pixels. This means that it contains 25000 pixels, which means that we can hide 1250 characters in that image. That should be sufficient to hide a small message. Video files are much larger. Files of such size are not needed to hide simple text messages.

We have researched encryption with AES. We haven’t used it in hiding text in images. It might add an additional layer of security.

Attacks:-

We have handled the man in the middle attack, that if a man tries to hack the server, he will only find an encoded image that is sent while in the real life the user is sending a text message. Before the encoding of the message we hide the text within the least significant bits of the image pixels. Thus, it is not suspicious because the original image is not distorted at a rate that the hacker will notice. Also upon sending the request to the database we used the url encoded content type not the json content type to make the parameters not easily identified if someone tried to hack the request. If the MITM tries to hack the database, he will find the passwords hashed thus he won’t be able to attack or know the passwords because he doesn’t have the key to decrypt the passwords.



Explanation of Each Computer Security Technique

1. User Authentication:- We have used this in the login/signup part. We had a separated server that has a initiates HTTP requests for signup and login.

2. Block ciphers:- The passwords are hashed using the AES algorithm “block-ciphers” so that the database is secured. The Advanced Encryption Standard, or AES, is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data. The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, the third mixes columns. The last transformation is a simple exclusive or (XOR) operation performed on each column using a different part of the encryption key -- longer keys need more rounds to complete.

Advantages:-

1. Speed of transformation:algorithms are linear in time and constant in space.
2. Low error propagation:an error in encrypting one symbol likely will not affect subsequent symbols.

Disadvantages:-

1. Low diffusion:all information of a plaintext symbol is contained in a single ciphertext symbol.
2. Susceptibility to insertions/ modifications:an active interceptor who breaks the algorithm might insert spurious text that looks authentic.

3. Steganography:- This is defined as a process of writing hidden messages by using some techniques that no one else knows the existence of the message. We have used this technique in the part of encrypting the text inside the images, so the the man in the middle attack won't know that the text is included in the image specially that we are encrypted only the least significant bits.

Advantages:-

1. One-Way Hashing :- Used to ensure that a third party has not tampered with a sent message. This is accomplished by creating a hash of the message using a fixed character length for every item in the message, when the original items are in fact of variable character length.

2. Attaching Text to an Image: - Explanatory notes are attached to an image. In the medical profession this could be used when one medical office sends an image to another medical office.

Disadvantages:-

1. Little known technology
2. Common technology
3. Technology still being developed for
4. certain formats
5. Most algorithms known to
6. government departments
7. Once detected message is known

Libraries used

We’ve used the “crypto.Cipher” library to implement AES encryption for storing passwords in the database. We’ve used java’s library “crypto.spec.IvParameterSpec” to compute the initialization vector. We’ve used “crypto.spec.SecretKeySpec” to compute the secret key used to encrypt the passwords.
We’ve also used “util.Base64” to encode and decode strings to and from base64.
We’ve also used some standard java libraries to do basic operations. For example, we’ve used “imageio.ImageIO” to read images, etc…

list of all references and links

https://giuliascalaberni.wordpress.com/2017/01/19/transfer-images-with-java-socket/

https://stackoverflow.com/questions/11654562/how-convert-byte-array-to-string

https://stackoverflow.com/questions/3732109/simple-http-server-in-java-using-only-java-se-api?fbclid=IwAR2POgTjLm1PXLYwrm1TQmIj4eI6uv57gzk4qgopjPGqouWCW1389JhG64Q

https://www.logicbig.com/tutorials/core-java-tutorial/http-server/http-server-basic.html

https://codinginfinite.com/java-tcp-client-server-chat-application-sockets/

https://medium.com/@ssaurel/create-a-simple-http-web-server-in-java-3fc12b29d5fd

http://www.java2s.com/Tutorial/Java/0320__Network/LightweightHTTPServer.htm

https://stackoverflow.com/questions/7178937/java-bufferedimage-to-png-format-base64-string

https://www.veracode.com/security/man-middle-attack

http://www.iosrjournals.org/iosr-jece/papers/NCNS/76-81.pdf

https://www.quora.com/Cryptography-What-are-the-advantages-and-disadvantages-of-block-ciphers-over-stream-ciphers