JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

INFORMATION TECHNOLOGY LAB

B.Tech, Computer Science
Vth semester,2019-20

Encrypting Image with a text and a Key

And Sending Over Sockets

SUBMITTED BY:-

1. Chitrank Mishra (B2)(17103103)
2. Prakhar Agarwal (B2)(17103104)
3. Abhisek Das (B2)(17103125)
4. Shivam Pandey (B1)(17103013)
5. Rhythm Paliwal (B1)(17103001)

SUBMITTED To:-

Mr. Vikas Hassija

INTRODUCTION

Digital watermarking is the act of hiding a message related to a digital signal (i.e. an image, song, video) within the signal itself. It is a concept closely related to stenography, in that they both hide a message inside a digital signal. However, what separates them is their goal. Watermarking tries to hide a message related to the actual content of the digital signal, while in stenography the digital signal has no relation to the message, and it is merely used as a cover to hide its existence. Here, In the project we use digital watermarking as a feature of a chat application.

Application of the Project:

As we all know that privacy is a myth but leaving your data in open could be unsafe for you and for your close circle members. That's being secure and prevent your data from any leakage. That's what our application do.
Our Program secure the chat among the peer groups by the use of digital watermarking as a feature.

Following tasks are performed in the programs :

1. Socket Programming is used for the connection of server and client for exchange of the information among peers.
2. Image Encryption is also done along with watermarking which means data is two level secured.
3. Random key is generated at the time of image encryption , So no one can think of the stealing the key and encryt data.

Server Program:

There are several functions in the program which are as follows:

1. Upgrade_key() & generate_key():

```
string upgrade_key(string str) {
    int n = str.length();
    for (int i = 0; i < str.length(); i++)
        str[i] = (str[i] + n) % 256;
    return str;
}

string generate_key() {
    int key_len = 0;
    while (key_len == 0)
        key_len = rand() % 50;

    string key = "";
    for (int i = 0; i < key_len; i++) {
        key += (char)(rand() % 256);
    }
    return key;
}
```

->Used for updating and generation the random key at the time of image encryption.

2. AddText():

This function has two pasing variables, one is for string or text which we want to send and other is the image in which we are going to hide the text.

```
void AddText(string st,Mat image){

    char ch;

    ch=s[0];
    int len=st.length(),cnt=0;

    int bit_count = 0;

    bool last_null_char = false;
    bool encoded = false;

    for(int row=0; row < image.rows; row++) {
        for(int col=0; col < image.cols; col++) {
            for(int color=0; color < 3; color++) {

                Vec3b pixel = image.at<Vec3b>(Point(row,col));

                if(isBitSet(ch,7-bit_count))
                    pixel.val[color] |= 1;
                else
                    pixel.val[color] &= ~1;

                image.at<Vec3b>(Point(row,col)) = pixel;

                bit_count++;

                if(last_null_char && bit_count == 8) {
                    encoded  = true;
                    goto OUT;
                }

                if(bit_count == 8) {
                    bit_count = 0;
                    cnt++;
                    ch=st[cnt];
```

3. EncryptImage():

After hiding the text we will do further encrytion of the imae so that it will be more difficult for someone to deoce or decrypt the text we want to remain confidential.

```
string EncrytImage(){
    string key =generate_key();
    int key_thres = key.length();
    int key_p = 0;
    Mat image = imread("Images/Imagewithtext.jpeg");
    for (int r = 0; r < image.rows; r++) {
        for (int c = 0; c < image.cols; c++) {
            for (int p = 0; p < 3; p++) {

                image.at<Vec3b>(r, c)[p]= (image.at<Vec3b>(r, c)[p] + (int)key[key_p])%256;
                key_p++;
                if (key_p == key_thres) {
                    key = upgrade_key(key);
                    key_p = 0;
                    key_thres--;
                    if (key_thres <= 0)
                        key_thres = key.length();
                }
            }
        }
    }
    imwrite("Images/EncrytedImage.jpeg",image);

    return key;
}
```

4. Send_Image():

This function is for sending the image and the key through the socket to the client.

Client Program:

Some function are same as of server program i.e; update_key() and socket creation.

1. Extract_Text():

This function is for taking out the text that we hide in the image using digital watermarking.

```
void Extract_Text(){

    Mat image = imread("Received/Imagewithtext.jpeg");
    if(image.empty()) {
        cout << "Image Error\n";
        exit(-1);
    }

    char ch=0;
    int bit_count = 0;

    for(int row=0; row < image.rows; row++) {
        for(int col=0; col < image.cols; col++) {
            for(int color=0; color < 3; color++) {

                Vec3b pixel = image.at<Vec3b>(Point(row,col));

                if(isBitSet(pixel.val[color],0))
                    ch |= 1;

                bit_count++;

                if(bit_count == 8) {

                    if(ch == '\0')
                        goto OUT;

                    bit_count = 0;
                    cout << ch;
                    ch = 0;
                }
                else {
                    ch = ch << 1;
                }

            }
MS  OUTPUT  DEBUG CONSOLE  TERMINAL
s you should add the directory containing `opencv.pc'
 PKG_CONFIG_PATH environment variable
```

2. Decryt_Image():

This function come first in role than extract text because we have to decrpyt the image first using the key and after that we gonna ake out the hidden text.

```
void Decryt_Image(string key){
    int key_thres = key.length();
    int key_p = 0;
    Mat image = imread("Received/EncrytImage.jpeg");
    for (int r = 0; r < image.rows; r++) {
        for (int c = 0; c < image.cols; c++) {
            for (int p = 0; p < 3; p++) {

                image.at<Vec3b>(r, c)[p] = (image.at<Vec3b>(r, c)[p] - (int)key[key_p] +256) % 256;
                key_p++;
                if (key_p == key_thres) {
                    key = upgrade_key(key);
                    key_p = 0;
                    key_thres--;
                    if (key_thres <= 0)
                        key_thres = key.length();
                }
            }
        }
    }
    imwrite("Received/Imagewithtext.jpeg",image);

}
```

Tools and Technologies Used:

1. VS Code(Editor)
2. OpenCV : For accessing the image and maniulate it.
3. C++ : Programming Language
4. Socket Programming