

Stegano
Technical Report

By

Sean Reidy

C00227196

Institute of Technology Carlow

April 2021

Contents

Introduction	3
Tools and Technologies used	3
Android Studio	3
Eclipse IDE	3
Code Snippets	4
MainActivity Class	4
Embed Class	5
Extract Class	9
Encrypt Class	12
Decrypt Class.....	14
Scan Class	15
EmbedExtract class	18
Activity Files	24
MainActivity activity file	24
Embed activity file.....	26
Extract activity file.....	28
Encrypt activity file.....	30
Decrypt activity file	31
Scan activity file	33

Introduction

This document will include the technologies used and needed to run the application. Since there is a lot of files and code involved, I have only included the most important code snippets.

Tools and Technologies used

Android Studio

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. You can download android studio here: [Android Studio](#)

Eclipse IDE

Eclipse is an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications. You can download eclipse ide here: [Eclipse Download](#)

I chose to code my application using Java.

Version

For android studio I used version 4.1.2. I am aware that there is a newer version of Android Studio which came out a week ago, however, I decided that I would stick to 4.1.2 for my application just to be safe. In the past, android updates how resulted in me changing code to make it work again.

Code Snippets

MainActivity Class

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button embed = findViewById(R.id.embedBtn);
        Button extract = findViewById(R.id.extractBtn);
        Button encrypt = findViewById(R.id.button);
        Button decrypt = findViewById(R.id.button2);
        Button scan = findViewById(R.id.scan_btn);

        embed.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(), Embed.class));
            }
        });

        extract.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(getApplicationContext(), Extract.class));
            }
        });

        encrypt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), Encrypt.class));
            }
        });

        decrypt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(getApplicationContext(), Decrypt.class));
            }
        });

        scan.setOnClickListener(new View.OnClickListener() {
            @Override
```

```

        public void onClick(View v) {
            startActivity(new Intent(getApplicationContext(), Scan.class));
        }
    });
}
}
}

```

Embed Class

```

public class Embed extends AppCompatActivity implements StringEmbedCallback {
    private TextView is_embedded;
    private ImageView imageView;
    private EditText message;
    private EditText secretKey;
    private static final int SELECT_PICTURE = 100;
    private static final String TAG = "Embed Class";
    private Bitmap ogImage;
    private Bitmap newImage;
    private TextEmbed embedText;
    private ImageSteganography imageSteganography;
    private ProgressDialog save;
    private Uri filepath;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_embed);
        getSupportActionBar().setTitle("Embed");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        //Frontend connections below
        is_embedded = findViewById(R.id.is_embeddedView);
        imageView = findViewById(R.id.imageview);
        message = findViewById(R.id.message);
        secretKey = findViewById(R.id.secretKey);
        Button SaveBtn = findViewById(R.id.saveBtn);
        Button imageBtn = findViewById(R.id.choosebtn);
        Button embedButton = findViewById(R.id.embedBtn);

        checkAppPermissions();

        embedButton.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

public void onClick(View view) {

    if (TextUtils.isEmpty(message.getText())) {
        message.setError("Please Enter a secret message");
    } else if (TextUtils.isEmpty(secretKey.getText())) {
        secretKey.setError("Please Enter a secret key");
    } else {

        is_embedded.setText("");
        if (filepath != null) {
            if (message.getText() != null) {
                ///below is used to create the new stego image and to embed the text. Object
instant
                imageSteganography = new ImageSteganography(message.getText().toString(),
                    secretKey.getText().toString(),
                    ogImage);

                embedText = new TextEmbed(Embed.this, Embed.this);

                embedText.execute(imageSteganography);
            }
        }
    }
}
});

```

```

imageBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        selectImage();
    }
});

```

```

SaveBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final Bitmap imgToSave = new Image();
        Thread PerformEncoding = new Thread(new Runnable() {
            @Override
            public void run() {
                saveImage(imgToSave);
            }
        });
    }
});

```

```

    });
    save = new ProgressDialog(Embed.this);
    save.setMessage("Saving, Please Wait...");
    save.setTitle("Saving Image");
    save.setIndeterminate(false);
    save.setCancelable(false);
    save.show();
    PerformEncoding.start();
    }
});

}

private void selectImage() {
    //Intent intent = new Intent(Intent.ACTION_PICK,
    MediaStore.Images.Media.INTERNAL_CONTENT_URI);
    // startActivityForResult(Intent.createChooser(intent, "Select Image for embedding"),
    SELECT_PICTURE);
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "Select Picture"), SELECT_PICTURE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    /* if (resultCode == RESULT_OK && requestCode == SELECT_PICTURE){
        filepath = data.getData();
        imageView.setImageURI(filepath);
    }*/
    //Image set to imageView
    if (requestCode == SELECT_PICTURE && resultCode == RESULT_OK && data != null &&
    data.getData() != null) {

        filepath = data.getData();
        try {
            ogImage = MediaStore.Images.Media.getBitmap(getContentResolver(), filepath);

            imageView.setImageBitmap(ogImage);
        } catch (IOException e) {
            Log.d(TAG, "Error embedding :)" + e);
        }
    }
}

private void saveImage(Bitmap bitmapImage) {
    OutputStream fOut;

```

```

File file = new File(Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_DOWNLOADS), "EmbeddedMsg" + ".PNG"); // the File to save ,
try {
    fOut = new FileOutputStream(file);
    bitmapImage.compress(Bitmap.CompressFormat.PNG, 100, fOut); // save the Bitmap to a file
    fOut.flush(); // Not really required
    fOut.close(); // do not forget to close the stream
    is_embedded.post(new Runnable() {
        @Override
        public void run() {
            save.dismiss();
        }
    });
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

```

@Override
public void onStartStringEmbedding() {

}

```

```

@Override
public void onFinishStringEmbedding(ImageSteganography result) {

    //By the end of embedText

    if (result != null && result.isEncoded()) {
        newImage = result.getEncoded_image();
        is_embedded.setText("Embedded");
        imageView.setImageBitmap(newImage);
    }
}

```

```

private void checkAppPermissions() {
    int permissionWriteStorage = ContextCompat.checkSelfPermission(this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE);
    int ReadPermission = ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE);
    List<String> listPermissionsNeeded = new ArrayList<>();
    if (ReadPermission != PackageManager.PERMISSION_GRANTED) {
        listPermissionsNeeded.add(Manifest.permission.READ_EXTERNAL_STORAGE);
    }
}

```



```

    }
    if (permissionWriteStorage != PackageManager.PERMISSION_GRANTED) {
        listPermissionsNeeded.add(Manifest.permission.WRITE_EXTERNAL_STORAGE);
    }
    if (!listPermissionsNeeded.isEmpty()) {
        ActivityCompat.requestPermissions(this, listPermissionsNeeded.toArray(new String[0]), 1);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu1, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.Help:
            startActivity(new Intent(getApplicationContext(), EmbedHelp.class));
    }
    return super.onOptionsItemSelected(item);
}
}

```

Extract Class

```

public class Extract extends AppCompatActivity implements StringExtractCallback {

    private static final int SELECT_PICTURE = 100;
    private static final String TAG = "Extract Class";
    private EditText secretKey;
    private Uri filepath;
    private Bitmap ogImage;
    private TextView textView;
    private ImageView imageView;
    private EditText message;

    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_extract);
    getSupportActionBar().setTitle("Extract");
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    //Instantiation of UI components
    textView = findViewById(R.id.isExtracted);

    imageView = findViewById(R.id.imageview);

    message = findViewById(R.id.message);
    secretKey = findViewById(R.id.secretKey);

    Button chooseImage = findViewById(R.id.choosebtn);
    Button extractBtn = findViewById(R.id.extractBtn);

    //Choose Image Button
    chooseImage.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            ImageSelect();
        }
    });

    //Extract Button
    extractBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (TextUtils.isEmpty(secretKey.getText())){
                secretKey.setError("Please Enter a secret key");
            }else
            if (filepath != null) {

                //Making the ImageSteganography object
                ImageSteganography imageSteganography = new
ImageSteganography(secretKey.getText().toString(),
                    ogImage);

                //Making the TextExtract object
                TextExtract textExtracting = new TextExtract(Extract.this, Extract.this);

                //Execute Task
                textExtracting.execute(imageSteganography);
            }
        }
    });
}

```

```

}

private void ImageSelect() {
    // Intent intent = new Intent(Intent.ACTION_PICK,
    MediaStore.Images.Media.INTERNAL_CONTENT_URI);
    //startActivityForResult(Intent.createChooser(intent, "Select Image for embedding"),
    SELECT_PICTURE);
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "Select Picture"), SELECT_PICTURE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    /* if (resultCode == RESULT_OK && requestCode == SELECT_PICTURE){
        filepath = data.getData();
        imageView.setImageURI(filepath);
    }*/ //crashes for some reason when i use the above code
    //Image set to imageView
    if (requestCode == SELECT_PICTURE && resultCode == RESULT_OK && data != null &&
    data.getData() != null) {

        filepath = data.getData();
        try {
            ogImage = MediaStore.Images.Media.getBitmap(getContentResolver(), filepath);

            imageView.setImageBitmap(ogImage);
        } catch (IOException e) {
            Log.d(TAG, "Error extracting");
        }
    }
}

@Override
public void onStartStringEmbedding() {
    //Whatever you want to do by the start of textDecoding
}

@Override
public void onFinishStringEmbedding(ImageSteganography result) {

    //By the end of textDecoding

    if (result != null) {
        if (!result.isExtracted())

```

```

        textView.setText("No secret message found");
    else {
        if (!result.isSecretKeyWrong()) {
            textView.setText("Extracted");
            message.setText("" + result.getMessage());
        } else {
            textView.setText("Wring secret key");
        }
    }
} else {
    textView.setText("Select An Image First");
}

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu1, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.Help:
            startActivity(new Intent(getApplicationContext(), ExtractHelp.class));
    }
    return super.onOptionsItemSelected(item);
}
}

```

Encrypt Class

```

public class Encrypt extends AppCompatActivity {
    TextView output;
    Button encryptBtn;
    String outputtxt;
    String AES= "AES";
    EditText inputString, inputPswd;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_encrypt);
    }
}

```

```

getSupportActionBar().setTitle("Encrypt");
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
inputString = (EditText) findViewById(R.id.EncryptedInput);
inputPswd = (EditText) findViewById(R.id.pswdInput);
encryptBtn = (Button) findViewById(R.id.decryptBtn);
output = (TextView) findViewById(R.id.outputView);
encryptBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        try {
            if (TextUtils.isEmpty(inputString.getText())){
                inputString.setError("Please Enter a secret message");
            }else if (TextUtils.isEmpty(inputPswd.getText())){
                inputPswd.setError("Please Enter a password to encrypt");
            }else
                outputtxt = encryptString(inputString.getText().toString(), inputPswd.getText().toString());
            output.setText(outputtxt);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});

}

private String encryptString(String data, String password) throws Exception {
    SecretKeySpec key = generateKey(password);
    Cipher c = Cipher.getInstance(AES);
    c.init(Cipher.ENCRYPT_MODE, key);
    byte[] encVal = c.doFinal(data.getBytes());
    String encryptedVal = Base64.encodeToString(encVal, Base64.DEFAULT);
    return encryptedVal;
}

private SecretKeySpec generateKey(String password) throws Exception {
    final MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    byte[] bytes = password.getBytes("UTF-8");
    messageDigest.update(bytes, 0, bytes.length);
    byte[] key = messageDigest.digest();
    SecretKeySpec secretKeySpec = new SecretKeySpec(key, "AES");
    return secretKeySpec;
}

```

Decrypt Class -----

```
public class Decrypt extends AppCompatActivity {

    TextView output;
    Button decryptBtn;
    String outputtxt;
    String AES = "AES";
    EditText encryptedString, inputPswd;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_decrypt);
        getSupportActionBar().setTitle("Decrypt");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        encryptedString = (EditText) findViewById(R.id.EncryptedInput);
        inputPswd = (EditText) findViewById(R.id.pswdInput);
        decryptBtn = (Button) findViewById(R.id.decryptBtn);
        output = (TextView) findViewById(R.id.outputView);
        decryptBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                try {
                    if (TextUtils.isEmpty(encryptedString.getText())){
                        encryptedString.setError("Please Enter an encrypted string");
                    }else if (TextUtils.isEmpty(inputPswd.getText())){
                        inputPswd.setError("Please Enter a password to decrypt");
                    }else
                        outputtxt = decrypt(encryptedString.getText().toString(), inputPswd.getText().toString());
                } catch (Exception e) {
                    e.printStackTrace();
                }
                output.setText(outputtxt);
            }
        });
    }

    private String decrypt(String outputString, String passwd) throws Exception {
        SecretKeySpec key = generateKey(passwd);
        Cipher c = Cipher.getInstance(AES);
        c.init(Cipher.DECRYPT_MODE, key);
        byte[] decodedVal = Base64.decode(outputString, Base64.DEFAULT);
        byte[] decVal = c.doFinal(decodedVal);
        String decryptVal = new String(decVal);
        return decryptVal;
    }
}
```

```

}

private SecretKeySpec generateKey(String password) throws Exception {
    final MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    byte[] bytes = password.getBytes("UTF-8");
    messageDigest.update(bytes, 0, bytes.length);
    byte[] key = messageDigest.digest();
    SecretKeySpec secretKeySpec = new SecretKeySpec(key, "AES");
    return secretKeySpec;
}

```

Scan Class

--

```

public class Scan extends AppCompatActivity {
    private static final int SELECT_PICTURE = 100;
    private Uri filepath;
    private String filepath2;
    private Bitmap original_image;
    private ImageView imageView;
    private ImageView imageView2;
    private int whichBtn = 1;

    private String messageMd5;
    private String message2Md5;
    private static final String TAG = "Scan Class";

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_scan);
        getSupportActionBar().setTitle("Scan");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        Button choose_image_button = findViewById(R.id.choosebtn);
        Button choose_image_button2 = findViewById(R.id.choose_image_button2);
        imageView2 = findViewById(R.id.imageview2);
        imageView = findViewById(R.id.imageview);
        choose_image_button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                whichBtn = 1;
                ImageChooser();
            }
        });
    }
}

```

```

});

choose_image_button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        whichBtn = 2;
        ImageChooser2();
    }
});

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu1, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.Help:
            startActivity(new Intent(getApplicationContext(), ScanHelp.class));
    }
    return super.onOptionsItemSelected(item);
}

private void ImageChooser() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(Intent.createChooser(intent, "Select Picture"), SELECT_PICTURE);
}

private void ImageChooser2() {
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
}

```



```

startActivityForResult(Intent.createChooser(intent, "Select Picture"), SELECT_PICTURE);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    //Image set to imageView
    if (requestCode == SELECT_PICTURE && resultCode == RESULT_OK && data != null &&
data.getData() != null) {

        filepath = data.getData();

        try {
            original_image = MediaStore.Images.Media.getBitmap(getContentResolver(), filepath);
            if(whichBtn == 1){
                imageView.setImageBitmap(original_image);
            }
            else if (whichBtn == 2){
                imageView2.setImageBitmap(original_image);
            }

        } catch (IOException e) {
            Log.d(TAG, "Error with task : " + e);
        }

        filepath2 = data.getData().toString();

        try {
            if(whichBtn == 1){
                messageMd5 = getMD5EncryptedString(filepath2);
                Toast.makeText(this, messageMd5, Toast.LENGTH_LONG).show();
            }
            else if(whichBtn == 2){
                message2Md5 = getMD5EncryptedString(filepath2);
                Toast.makeText(this, message2Md5, Toast.LENGTH_LONG).show();
            }

        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(this,"Failed to retrieve MD5 value", Toast.LENGTH_LONG).show();
        }
    }
}

```

```

if(whichBtn ==1){
    TextView message = findViewById(R.id.message12);
    message.setText(messageMd5);
}
else if(whichBtn == 2)
{
    TextView message2 = findViewById(R.id.message2);
    message2.setText(message2Md5);
}
}

public String getMD5EncryptedString(String encTarget){
    MessageDigest mdEnc = null;
    try {
        mdEnc = MessageDigest.getInstance("MD5");
    } catch (NoSuchAlgorithmException e) {
        Toast.makeText(this,"Exception while encrypting to md5", Toast.LENGTH_SHORT).show();
        e.printStackTrace();
    } // Encryption algorithm
    mdEnc.update(encTarget.getBytes(), 0, encTarget.length());
    String md5 = new BigInteger(1, mdEnc.digest()).toString(16);
    while ( md5.length() < 32 ) {
        md5 = "0"+md5;
    }
    return md5;
}
}

```

EmbedExtract class

```

class EmbedExtract {
    private static final String END_MESSAGE_CONSTANT = "#!@";
    private static final String START_MESSAGE_CONSTANT = "@!#";
    private static final String TAG = EmbedExtract.class.getName();
    private static final int[] binary = {16, 8, 0};
    private static final byte[] andByte = {(byte) 0xC0, 0x30, 0x0C, 0x03};
    private static final int[] toShift = {6, 4, 2, 0};

    private static byte[] embedMsg(int[] integer_pixel_array, int image_columns, int image_rows,
        MessageEmbedStatus embedMsgStatus, ProgressHandler progressHandler) {

        //below is for the 3 RGB channels
        int channels = 3;

        int shiftIndex = 4;

```

```

//creating 2D byte_array
byte[] result = new byte[image_rows * image_columns * channels];

int resultIndex = 0;

for (int row = 0; row < image_rows; row++) {

    for (int col = 0; col < image_columns; col++) {

        //2D matrix into a 1D matrix
        int element = row * image_columns + col;

        byte tmp;

        for (int channelIndex = 0; channelIndex < channels; channelIndex++) {

            if (!embedMsgStatus.isMsgEmbedded()) {

                // Shifting integer value by 2 in left and replacing the two least significant digits with
the message_byte_array values..
                //If you wanted to use more then the last 2 significant bits, add here...
                tmp = (byte) (((integer_pixel_array[element] >> binary[channelIndex]) & 0xFF) & 0xFC)
| ((embedMsgStatus.getByteArrayMessage())[embedMsgStatus.getCurrentMessageIndex()] >>
toShift[(shiftIndex++)
                % toShift.length]) & 0x3)); // 6

                if (shiftIndex % toShift.length == 0) {

                    embedMsgStatus.incrementMessageIndex();

                    if (progressHandler != null)
                        progressHandler.increment(1);

                }

                if (embedMsgStatus.getCurrentMessageIndex() ==
embedMsgStatus.getByteArrayMessage().length) {

                    embedMsgStatus.setMsgEmbedded();

                    if (progressHandler != null)
                        progressHandler.finished();

                }
            } else {
                //Simply copy the integer to result array
                tmp = (byte) (((integer_pixel_array[element] >> binary[channelIndex]) & 0xFF));
            }
        }
    }
}

```

```

        result[resultIndex++] = tmp;

    }

}

return result;

}

public static List<Bitmap> embedMsg(List<Bitmap> splitted_images,
    String encrypted_message, ProgressHandler progressHandler) {

    // result method

    List<Bitmap> result = new ArrayList<>(splitted_images.size());

    //Adding start and end message constants to the encrypted message
    encrypted_message = encrypted_message + END_MESSAGE_CONSTANT;
    encrypted_message = START_MESSAGE_CONSTANT + encrypted_message;

    //getting byte array from string
    byte[] byte_encrypted_message = encrypted_message.getBytes(Charset.forName("ISO-8859-
1"));

    //Message embedding Status
    MessageEmbedStatus message = new MessageEmbedStatus(byte_encrypted_message,
encrypted_message);

    //Progress Handler
    if (progressHandler != null) {
        progressHandler.setTotal(encrypted_message.getBytes(Charset.forName("ISO-8859-
1")).length);
    }

    //Just a log to get the byte message length
    Log.i(TAG, "Message length " + byte_encrypted_message.length);

    for (Bitmap bitmap : splitted_images) {

        if (!message.isMsgEmbedded()) {

```

```

//getting bitmap height and width
int width = bitmap.getWidth();
int height = bitmap.getHeight();

//Making 1D integer pixel array
int[] oneD = new int[width * height];
bitmap.getPixels(oneD, 0, width, 0, 0, width, height);

//getting bitmap density
int density = bitmap.getDensity();

//embedded final image
byte[] embeddedImage = embedMsg(oneD, width, height, message, progressHandler);

//converting byte_image_array to integer_array
int[] oneDMod = Utility.byteArrayToIntArray(embeddedImage);

//creating bitmap from encrypted_image_array
Bitmap encoded_Bitmap = Bitmap.createBitmap(width, height,
    Bitmap.Config.ARGB_8888);
encoded_Bitmap.setDensity(density);

int masterIndex = 0;

//setting pixel values of above bitmap
for (int j = 0; j < height; j++)
    for (int i = 0; i < width; i++) {

        encoded_Bitmap.setPixel(i, j, Color.argb(0xFF,
            oneDMod[masterIndex] >> 16 & 0xFF,
            oneDMod[masterIndex] >> 8 & 0xFF,
            oneDMod[masterIndex++] & 0xFF));

    }

    result.add(encoded_Bitmap);

} else {
    //Just add the image chunk to the result
    result.add(bitmap.copy(bitmap.getConfig(), false));
}
}

return result;
}

```

```

private static void extractMsg(byte[] byte_pixel_array, int image_columns,

```

```

        int image_rows, msgExtractStatus extractingStatus) {

//encrypted message
Vector<Byte> byte_encrypted_message = new Vector<>();

int shiftIndex = 4;

byte tmp = 0x00;

for (byte aByte_pixel_array : byte_pixel_array) {

//get last two bits from byte_pixel_array
tmp = (byte) (tmp | ((aByte_pixel_array << toShift[shiftIndex
    % toShift.length]) & andByte[shiftIndex++ % toShift.length]));

if (shiftIndex % toShift.length == 0) {
//adding temp byte value
byte_encrypted_message.addElement(tmp);

//converting byte value to string
byte[] nonso = {byte_encrypted_message.elementAt(byte_encrypted_message.size() - 1)};
String str = new String(nonso, Charset.forName("ISO-8859-1"));

if (extractingStatus.getMessage().endsWith(END_MESSAGE_CONSTANT)) {

    Log.i("TEST", "Decoding ended");

//fixing ISO-8859-1 decoding
byte[] temp = new byte[byte_encrypted_message.size()];

for (int index = 0; index < temp.length; index++)
    temp[index] = byte_encrypted_message.get(index);

String stra = new String(temp, Charset.forName("ISO-8859-1"));

extractingStatus.setMessage(stra.substring(0, stra.length() - 1));
//end fixing

extractingStatus.setEnded();

break;
} else {
//just add the extracted message to the original message

```

```

extractingStatus.setMessage(extractingStatus.getMessage() + str);

//If there was no message there and only start and end message constant was there
if (extractingStatus.getMessage().length() == START_MESSAGE_CONSTANT.length()
    && !START_MESSAGE_CONSTANT.equals(extractingStatus.getMessage())) {

    extractingStatus.setMessage("");
    extractingStatus.setEnded();

    break;
}
}

tmp = 0x00;
}

}

if (!Utility.isStringEmpty(extractingStatus.getMessage()))
    //removing start and end constants form message

    try {

extractingStatus.setMessage(extractingStatus.getMessage().substring(START_MESSAGE_CONSTANT.
length(), extractingStatus.getMessage()
    .length()
    - END_MESSAGE_CONSTANT.length()));
    } catch (Exception e) {
        e.printStackTrace();
    }

}

```

```

public static String extractMsg(List<Bitmap> encodedImages) {

    //Creating object
    msgExtractStatus extractStatus = new msgExtractStatus();

    for (Bitmap bit : encodedImages) {
        int[] pixels = new int[bit.getWidth() * bit.getHeight()];

        bit.getPixels(pixels, 0, bit.getWidth(), 0, 0, bit.getWidth(),
            bit.getHeight());

        byte[] b;

```

```

        b = Utility.convertArray(pixels);

        extractMsg(b, bit.getWidth(), bit.getHeight(), extractStatus);

        if (extractStatus.isEnded())
            break;
    }

    return extractStatus.getMessage();
}

//this calculates the number of pixels which are needed to embed the secret message.
public static int numberOfPixelForMessage(String message) {
    int result = -1;
    if (message != null) {
        message += END_MESSAGE_CONSTANT;
        message = START_MESSAGE_CONSTANT + message;
        result = message.getBytes(Charset.forName("ISO-8859-1")).length * 4 / 3;
    }

    return result;
}

```

Activity Files

MainActivity activity file

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/relativeLayout4"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="16dp">

<Button
    android:id="@+id/scan_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="44dp"
    android:layout_marginEnd="32dp"
    android:background="#7f00e0"

```



```
android:text="Scan"
android:textColor="#FFFFFF"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/extractBtn" />
```

```
<Button
    android:id="@+id/embedBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="108dp"
    android:background="#7f00e0"
    android:text="Embed"
    android:textColor="#FFFFFF"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView3" />
```

```
<Button
    android:id="@+id/extractBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="64dp"
    android:layout_marginEnd="32dp"
    android:background="#7f00e0"
    android:text="Extract"
    android:textColor="#FFFFFF"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/embedBtn" />
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="52dp"
    android:layout_marginTop="64dp"
    android:background="#7f00e0"
    android:text="Encrypt"
    android:textColor="#FFFFFF"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/embedBtn" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="52dp"
    android:layout_marginTop="44dp"
```

```
    android:background="#7f00e0"
    android:text="Decrypt"
    android:textColor="#FFFFFF"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button" />
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="359dp"
    android:layout_height="82dp"
    android:layout_marginTop="32dp"
    android:gravity="center"
    android:text="Welcome to Stegano !!!"
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Embed activity file

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp">
```

```
<ImageView
    android:id="@+id/imageview"
    android:layout_width="375dp"
    android:layout_height="165dp"
    android:layout_marginTop="88dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.363"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/choosebtn"
    android:layout_width="370dp"
    android:layout_height="61dp"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
```

```
android:background="#7f00e0"
android:text="Choose an image"
android:textColor="#FFFFFF"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.777"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.006" />
```

```
<EditText
    android:id="@+id/secretKey"
    android:layout_width="335dp"
    android:layout_height="56dp"
    android:ems="10"
    android:hint="Enter secret key here"
    android:inputType="textPassword"
    android:textSize="14sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.545"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageview"
    app:layout_constraintVertical_bias="0.287" />
```

```
<EditText
    android:id="@+id/message"
    android:layout_width="338dp"
    android:layout_height="62dp"
    android:ems="10"
    android:hint="Enter Secret message here"
    android:inputType="textPersonName"
    android:textSize="14sp"
    app:layout_constraintBottom_toTopOf="@+id/secretKey"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.512"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageview"
    app:layout_constraintVertical_bias="0.228" />
```

```
<Button
    android:id="@+id/embedBtn"
    android:layout_width="94dp"
    android:layout_height="50dp"
    android:background="#7f00e0"
    android:text="Embed"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
```

```
    android:textSize="14dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="@+id/secretKey"
    app:layout_constraintTop_toBottomOf="@+id/secretKey" />
```

```
<Button
    android:id="@+id/saveBtn"
    android:layout_width="90dp"
    android:layout_height="50dp"
    android:layout_marginEnd="12dp"
    android:background="#7f00e0"
    android:text="Save Image"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="@+id/secretKey"
    app:layout_constraintTop_toBottomOf="@+id/secretKey" />
```

```
<TextView
    android:id="@+id/is_embeddedView"
    android:layout_width="85dp"
    android:layout_height="25dp"
    android:textStyle="bold"
    android:textColor="#7f00e0"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/secretKey"
    app:layout_constraintVertical_bias="0.19" />
```

```
</android.support.constraint.ConstraintLayout>
```

Extract activity file

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relativeLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="16dp">

    <ImageView
```

```
android:id="@+id/imageview"  
android:layout_width="370dp"  
android:layout_height="176dp"  
android:layout_marginStart="4dp"  
android:layout_marginTop="12dp"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/choosebtn" />
```

<Button

```
android:id="@+id/choosebtn"  
android:layout_width="351dp"  
android:layout_height="46dp"  
android:layout_centerHorizontal="true"  
android:layout_centerVertical="true"  
android:layout_marginStart="8dp"  
android:layout_marginTop="4dp"  
android:background="#7f00e0"  
android:text="Choose Image"  
android:textColor="#FFFFFF"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<EditText

```
android:id="@+id/secretKey"  
android:layout_width="359dp"  
android:layout_height="55dp"  
android:layout_marginStart="8dp"  
android:layout_marginTop="12dp"  
android:ems="10"  
android:hint="Enter secret key here"  
android:inputType="textPassword"  
android:textSize="14sp"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/imageview" />
```

<EditText

```
android:id="@+id/message"  
android:layout_width="375dp"  
android:layout_height="53dp"  
android:layout_marginStart="4dp"  
android:layout_marginTop="12dp"  
android:ems="10"  
android:inputType="textPersonName"  
android:textSize="14sp"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/extractBtn" />
```

<Button

```
android:id="@+id/extractBtn"
android:layout_width="347dp"
android:layout_height="47dp"
android:layout_marginStart="8dp"
android:layout_marginTop="4dp"
android:background="#7f00e0"
android:text="Extract"
android:textAlignment="center"
android:textColor="#FFFFFF"
android:textSize="14dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/secretKey" />
```

```
<TextView
    android:id="@+id/isExtracted"
    android:textColor="#7f00e0"
    android:layout_width="70dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="152dp"
    android:layout_marginTop="16dp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/message" />
```

```
</android.support.constraint.ConstraintLayout>
```

Encrypt activity file

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/decryptBtn"
        android:layout_width="155dp"
        android:layout_height="60dp"
        android:layout_marginStart="128dp"
        android:layout_marginTop="56dp"
        android:text="Encrypt"
        android:background="#7f00e0"
        android:textColor="#FFFFFF"
```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/pswdInput" />
```

```
<EditText  
    android:id="@+id/outputView"  
    android:layout_width="399dp"  
    android:layout_height="71dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="28dp"  
    android:hint="Encrypted Message"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/decryptBtn" />
```

```
<EditText  
    android:id="@+id/EncryptedInput"  
    android:layout_width="348dp"  
    android:layout_height="49dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="16dp"  
    android:ems="10"  
    android:hint="Please enter your message here: "  
    android:inputType="textPersonName"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText  
    android:id="@+id/pswdInput"  
    android:layout_width="344dp"  
    android:layout_height="55dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="28dp"  
    android:ems="10"  
    android:hint="Please enter a secret key:"  
    android:inputType="textPersonName"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/EncryptedInput" />  
</android.support.constraint.ConstraintLayout>
```

Decrypt activity file

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"
```

```
android:layout_height="match_parent">
```

```
<Button  
    android:id="@+id/decryptBtn"  
    android:layout_width="155dp"  
    android:layout_height="60dp"  
    android:layout_marginStart="128dp"  
    android:layout_marginTop="56dp"  
    android:background="#7f00e0"  
    android:textColor="#FFFFFF"  
    android:text="Decrypt"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/pswdInput" />
```

```
<EditText  
    android:id="@+id/outputView"  
    android:layout_width="399dp"  
    android:layout_height="71dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="28dp"  
    android:hint="Decrypted Message"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/decryptBtn" />
```

```
<EditText  
    android:id="@+id/EncryptedInput"  
    android:layout_width="348dp"  
    android:layout_height="42dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="16dp"  
    android:ems="10"  
    android:hint="Please enter your encrypted string here: "  
    android:inputType="textPersonName"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText  
    android:id="@+id/pswdInput"  
    android:layout_width="346dp"  
    android:layout_height="52dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="28dp"  
    android:ems="10"  
    android:hint="Please enter a secret key:"  
    android:inputType="textPersonName"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/EncryptedInput" />
```



```
</android.support.constraint.ConstraintLayout>
```

Scan activity file

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/relativeLayout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_margin="16dp">

<TextView
    android:id="@+id/message2"
    android:layout_width="304dp"
    android:layout_height="54dp"
    android:layout_marginTop="304dp"
    android:ems="10"
    android:hint="MD5 Hash Value"
    android:inputType="textPersonName"

    android:textSize="14sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.346"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageview" />

<ImageView
    android:id="@+id/imageview2"
    android:layout_width="326dp"
    android:layout_height="47dp"
    android:layout_marginTop="368dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/choose_image_button2"
    android:layout_width="366dp"
    android:layout_height="27dp"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:background="#7f00e0"
    android:text="Choose an image"
    android:textColor="#FFFFFF"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintHorizontal_bias="0.777"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.471" />
```

```
<ImageView
    android:id="@+id/imageview"
    android:layout_width="327dp"
    android:layout_height="48dp"
    android:layout_marginTop="92dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.49"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:id="@+id/choosebtn"
    android:layout_width="371dp"
    android:layout_height="30dp"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:background="#7f00e0"
    android:text="Choose an image"
    android:textColor="#FFFFFF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintHorizontal_bias="0.777"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.006" />
```

```
<TextView
    android:id="@+id/message12"
    android:layout_width="304dp"
    android:layout_height="54dp"
    android:layout_marginTop="52dp"
    android:ems="10"
    android:hint="MD5 Hash Value"
    android:inputType="textPersonName"

    android:textSize="14sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/imageview" />
```

</android.support.constraint.ConstraintLayout>