VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

K. R. Road, V. V. Pura, Bengaluru-560 004

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MOBILE APPLICATION DEVELOPMENT

MINI PROJECT (18CSMP68)

“ENCRYPTION DECRYPTION APP”

Submitted By

1BI19CS149	**SOURAV B S**
1BI19CS155	**SUGAM CHAND M**

for the academic year 2021-2022

Under the guidance of

Prof. Tejashwini P S

Assistant Professor

Dept. of CS & E, BIT

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018, Karnataka

BANGALORE INSTITUTE OF TECHNOLOGY

K.R. Road, V.V. Pura, Bengaluru-560 004

Department of Computer Science & Engineering

Certificate

This is to certify that the implementation of **Mobile Application Development MINI PROJECT (18CSMP68)** entitled “**ENCRYPTION DECRYPTION APP**” has been successfully completed by

1BI19CS149	**SOURAV B S**
1BI19CS155	**SUGAM CHAND M**

of VI semester B.E. for the partial fulfilment of the requirements for the Bachelor's degree in Computer Science & Engineering of the Visvesvaraya Technological University during the academic year 2021-2022.

Lab In charges:

Prof. Tejashwini P S	**Dr. J. GIRIJA**
Assistant Professor	Professor and Head
Dept. of CS & E, BIT	Dept. CS & E, BIT

Examiners: 1) 2)

ACKNOWLEDGEMENT

The knowledge & satisfaction that accompany the successful completion of any task would be incomplete without mention of the people who made it possible, whose guidance and encouragement crowned my effort with success. I would like to thank all and acknowledge the help I have received to carry out this Mini Project.

I would like to convey my thanks to the Head of Department **Dr. J. GIRIJA** for being kind enough to provide the necessary support to carry out the mini-project.

I am most humbled to mention the enthusiastic influence provided by the lab in-charge **Prof. Tejashwini P S,** Assistant Professor, Department of CS & E on the project for their ideas, time to time suggestions for being a constant guide and co-operation showed during the venture and making this project a great success.

I would also take this opportunity to thank my friends and family for their constant support and help. I'm very much pleasured to express my sincere gratitude for the friendly co-operation showed by all the **staff members** of the Computer Science Department, BIT.

1BI19CS149

SOURAV B S

1BI19CS155

SUGAM CHAND M

Table of Contents

Number Contents page no.

Chapter -1

INTRODUCTION

1.1 Android Studio

Android is one of the most popular mobile device platforms. The Android platform allows developers to write managed code using Java to manage and control the Android device. Android Studio is a popular IDE developed by Google for developing applications that are targeted at the Android platform. Note that Android Studio has replaced Eclipse as the IDE of choice for developing Android applications. This article presents a discussion on how to get started using the Android Studio for developing Android applications. Android Studio contains tools such as the Android Virtual Device Manager and the Android Device Monitor. It also contains Gradle, which helps you configure your Android application seamlessly. Some of the interesting features of Android Studio include the following:

- Support for a fast emulator
- Support for Gradle
- Support for plenty of code templates and GitHub integration
- Support for Google Cloud Platform
- Support for template-based wizards for creating Android designs and components
- Support for rich layout editor
- Support for deep code analysis
- Support for extensive set of tools and frameworks

The Android application environment provides three main types of layouts as Linear, Relative and Constraint layout.

Linear Layout- A layout where the positions of the children is aligned in a single direction. It has vertical and horizontal orientations in turn, the content of views can be set to display vertically or horizontally in the activity layout.

Relative Layout- A layout where the positions of the children can be described in relation to each other or to the parent makes a relationship between views or screen anywhere in the layout.

Constraint Layout- A layout on Android application is used to create more complex UI’s. it needs constraints to place a text view in the layout having attributes in the constraint widgets that is right constraint, left constraint and to make center constraint has zero.

1.2 Project Summary

We will be building an Android Application that can **Encrypt** and **Decrypt** a message using the Encoding and Decoding algorithm i.e., using AES algorithm respectively. The app’s homepage will the user two option:

1. **Encryption:** It is the process of transforming a readable message into an unreadable one. To do so we use encoding algorithms.
2. **Decryption:** It is the process of transforming data or information from an unreadable to readable form. To do so we use decoding algorithms.

1.3 Project Purpose

This app aims at providing secure transfer of data using secret key such that only the intended or authorized person will be able to decrypt and view the encrypted data.

Our application uses AES algorithm, which is a symmetric cryptographic technique.

The encrypted text can be decoded only by using the secret key generated by the algorithm. The text is encrypted and decrypted using the same secret key. The secret key must be shared to the receiver if he wants to decrypt the text.

1.4 Project Scope

The purpose of the encrypt decrypt application is to allow users be able send data securely, that is, sending secret messages securely. The users will be able to send data securely and be sure that it is sent to the correct user and the receiver has decoded it correctly. The encrypt decrypt application will be written in java, but due to the lack of experience in java, while developing the application, practicing techniques with java and working on it as much as possible will help hone some java skills and be more ready to develop the application. For the scope of the project, the project will be tested as the program is being developed. A encrypt/decrypt interface will be developed and tested, and GUI’s will be developed and tested, for the users’ benefits. When the encrypt decrypt application is near completion, more testing will be done in order to make it less buggy or more user friendly.

1.5 Problem Statement

The project aims to provide the correct data with security to the users. For some of the users the data might be lost during the transmission process in the network and for some, the data might be changed by the unauthorized person in the network and there are some other security problems in the network. Only Authorized persons i.e., who are using our application will be able to decrypt the data and view the data.

1.6 Objectives of the Project

The objective of this process is as follows:

1. To develop an instant encrypting solution to enable users to seamlessly with each other.
2. The project should be very easy to use enabling even a novice person to use it.

1.7 Organisation of the Project

The project was organised in a systematic way. First we analysed what are the basic features to be included in the project to make it acceptable. As it is a android based oriented project, we made the sketches prior, so as to have an idea like how our output must look like. After all these, the layouts of various activities was formulated as a paper work. All the required software were downloaded. Finally, the successful implementation of the project.

CHAPTER-2

SYSTEM SPECIFICATION

2.1 User Characteristics

- Start
- First, we view the app homepage where the “ENCRYPT TEXT” and “DECRYPT TEXT” options are available
- Then we select the required option that is either encryption or decryption
- If the “ENCRYPT TEXT” option is selected we are leaded to a page where we need to enter the data to be encrypted.
- Then we press the “ENCRYPT TEXT” button which generates a secret key which is sharable to others using the share to other apps option
- We send the secret key to the required user by clicking the “SEND SECRET KEY” via SMS by giving their phone number
- On the “DECRYPT TEXT” side we enter the encrypted text that is shared and the secret key that is shared via SMS by the sender
- When we click on the “DECRYPT TEXT” option we get the decrypted version of the data being sent which is being displayed on the screen.

2.2 Hardware Requirements

- Android Phone
- 128 MB minimum RAM Required
- Internet or LAN Connections
- Processor with Speed of 500 MHz

2.3 Software Requirements

- Operating System: Windows 10 or Linux (Fedora) or macOS
- Language used : Java
- Toolkit : Android Studio 4.1
- Size: 727 to 877 MB
- Resolution: 1280 x 800 minimum screen resolution.
- RAM: 8 GB RAM or more
- Disk space: 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator

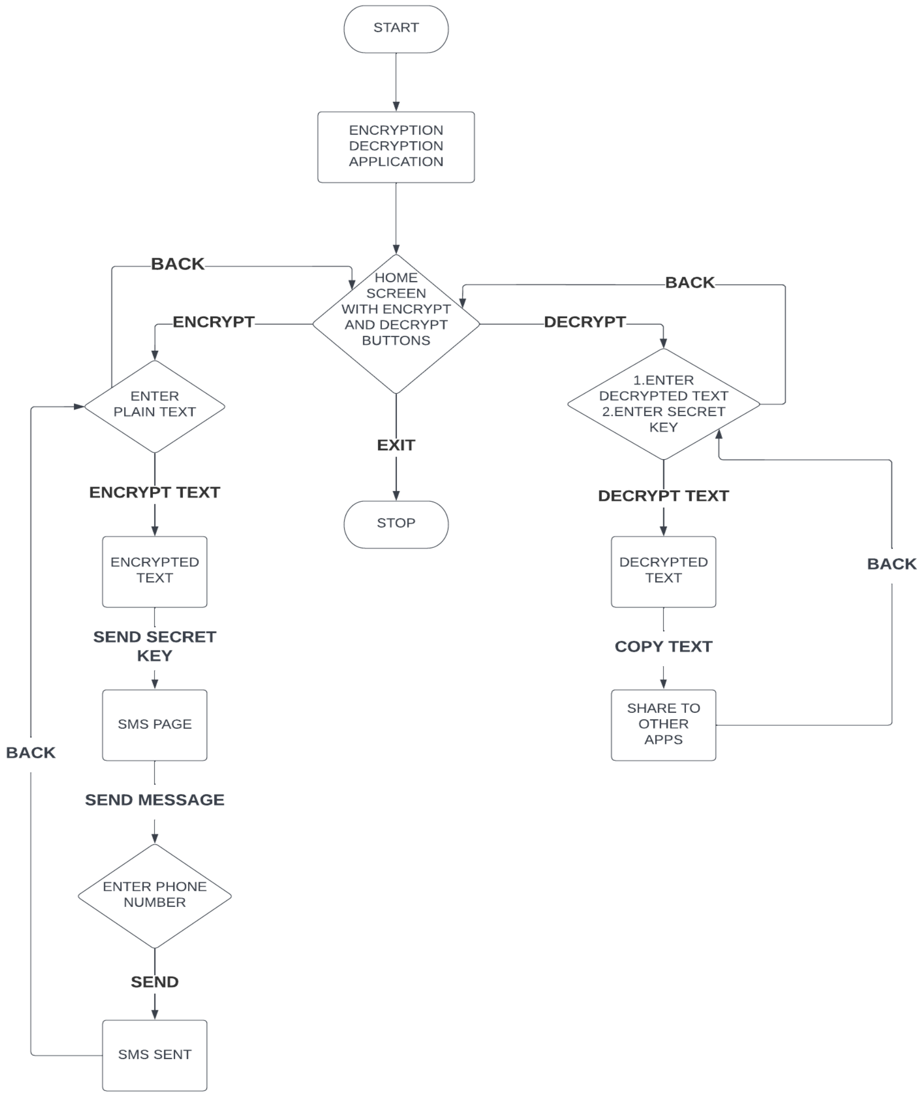
2.4 Project Operation Constraints

- User only can use or install this app on android devices.

CHAPTER-3

FLOW DIAGRAM

3.1 Flow Chart



3.2 Description of the Flow chart

1. Start
2. First, we view the app homepage where the “ENCRYPT TEXT” and “DECRYPT TEXT” options are available
3. Then we select the required option that is either encryption or decryption
4. If the “ENCRYPT TEXT” option is selected we are leaded to a page where we need to enter the data to be encrypted.
5. Then we press the “ENCRYPT TEXT” button which generates a secret key which is sharable to others using the share to other apps option
6. We send the secret key to the required user by clicking the “SEND SECRET KEY” via SMS by giving their phone number
7. On the “DECRYPT TEXT” side we enter the encrypted text that is shared and the secret key that is shared via SMS by the sender
8. When we click on the “DECRYPT TEXT” option we get the decrypted version of the data being sent which is being displayed on the screen.
9. Stop

3.3 Module Description

The functionality of the encryption application is to give the ability to share the data with the authorized person such that he/she can decrypt the encrypted data with the secret key.

Users

The users will be anyone who has the encryption decryption application.

Main Menu

When the user runs the encryption application, the user will see the main menu, which will have two options either to select “ENCRYPT TEXT” or “DECRYPT TEXT”.

Encrypt Text

Here we enter the data to be encrypted and click the Encrypt text button. In the same module we can share the encrypted data to the authorized users through other apps. We have a Send Secret Key option which shares the secret key through SMS page.

SMS page

Here the secret key is automatically copied to the text area and we enter the phone number of the authorized person.

Decrypt Text

Here we enter the encrypted text shared and the secret key received. The decrypted text is displayed on the screen.

CHAPTER-4

IMPLEMENTATION

4.1 Built-in Functions

1)**getSystemService(Context.CLIPBOARD_SERVICE)-** for accessing and modifying the contents of the global clipboard.

2)**Cipher.getInstance("AES")-** creates a Cipher instance using the encryption algorithm called AES.

3) **cipher.init(Cipher.ENCRYPT_MODE, key, iv)-** initialize this cipher with a key and a set of algorithm parameters.

4) **cipher.doFinal(input.getBytes())-** perform the encryption or decryption operation. It also resets the *Cipher* object to the state it was in when previously initialized via a call to *init()* method, making the *Cipher* object available to encrypt or decrypt additional messages.

5)**Base64.getEncoder().encodeToString(cipherText)-** It returns a Base64.Encoder that encodes using the Basic type base64 encoding scheme.

6) **Base64.getDecoder().decode(encodedKey)-** It returns a Base64.Decoder that decodes using the Basic type base64 encoding scheme.

7)**smsManager.setTextMessage(phoneNum,null,sms,null,null)-** sends text message to the given number.

8)**ContextCompat.checkSelfPermission()-** To check if the user has already granted your app a particular permission. This method returns either PERMISSION_GRANTED or PERMISSION_DENIED, depending on whether your app has the permission.

9)**ActivityCompat.requestPermissions()-** It says what permission your app wants, but doesn't say why.

10)**cp.setPrimaryClip()-** Sets the current primary clip on the clipboard. This is the clip that is involved in normal cut and paste operations.

4.2 Android Manifest and Special Permission

Every app project must have an AndroidManifest.xml file (with precisely that name) at the root of the project source set. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

While building our app into the final application package (APK), the Android build tools use the package attribute for two things:

- It applies this name as the namespace for our app's generated activity_main.java class (used to access our app resources).
- It uses this name to resolve any relative class names that are declared in the manifest file.

4.2.1 AndroidManifest.XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.encryptdecrypt" >
    <uses-permission android:name="android.permission.SEND_SMS"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.EncryptDecrypt" >
        <activity
            android:name=".SMSActivity"
            android:exported="false" />
        <activity
            android:name=".Decrypt"
            android:exported="false" />
        <activity
            android:name=".Encrypt"
            android:exported="false" />
        <activity
            android:name=".MainActivity”
            android:exported="true" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.3 XML and JAVA code:

Activity_Main.xml-

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
    android:background="#f00"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="centerCrop"
        android:src="@drawable/background"
        />


    <!--Title of the application-->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="5dp"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginTop="20dp"
            android:text="Encryption Decryption App”
            android:textAlignment="center"
            android:paddingTop="45dp"
            android:paddingBottom="30dp"
            android:textColor="#FFFFFF"
            android:textSize="40dp"
            android:textStyle="bold" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_gravity="center"
        android:layout_marginTop="30dp"
        android:orientation="vertical">

        <!--Button for encryption-->
        <Button
            android:id="@+id/btVar1"
            android:layout_width="220dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_margin="50dp"
            android:background="#000000"
            android:text="Encrypt text"
            android:textColor="#FFFFFF"
            android:textSize="25dp"
            android:textStyle="bold"
            android:onClick="encrypt"
            />

        <!--Button for decryption-->
        <Button
            android:id="@+id/btVar2"
            android:layout_width="220dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_margin="50dp"
            android:background="#000000"
            android:text="Decrypt text"
            android:textColor="#FFFFFF"
            android:textSize="25dp"
```

```
            android:textStyle="bold"
            android:onClick="decrypt"
            />

    </LinearLayout>

</RelativeLayout>
```

MainActivity.java-

```
package com.example.encryptdecrypt;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    }
    public void encrypt(View v){
        Intent i = new Intent(this,Encrypt.class);
        startActivity(i);
    }
    public void decrypt(View v){
        Intent i = new Intent(this,Decrypt.class);
        startActivity(i);
    }
}
```

ActivityEncrypt.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginLeft="15dp"
android:layout_marginTop="60dp"
android:layout_marginRight="15dp"
android:layout_marginBottom="20dp"
android:orientation="vertical">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:layout_marginTop="5dp"
    android:text="Plain text :"
    android:textColor="#FFFFFF"
    android:textSize="30dp"
    android:textStyle="bold" />

<!--input text for encryption-->
<EditText
    android:id="@+id/etVar1"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_gravity="center"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp"
    android:background="@drawable/shape"
    android:hint="  Enter the text here"
    android:paddingLeft="20dp"
    android:shadowColor="@color/white"
    android:textColor="#FFFFFF"
    android:textColorHint="@color/white"
    android:textSize="20sp" />

<!--start encryption-->
<Button
    android:id="@+id/btVar1"
    android:layout_width="230dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"
    android:background="#000000"
    android:onClick="enc"
```

```
            android:id="@+id/tvVar1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="left"
            android:layout_marginTop="5dp"
            android:textColor="#FAFAFA"
            android:textSize="20dp" />

        <!--button to copy encrypted code-->
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:orientation="horizontal">

            <Button
                android:id="@+id/btVar3"
                android:layout_width="160dp"
                android:layout_height="wrap_content"
                android:background="#000000"
                android:onClick="copy"
                android:text="Copy to clipboard"
                android:layout_gravity="center"
                android:textColor="#FFFFFF"
                android:layout_marginRight="13dp"
                android:textSize="20sp"
                android:textStyle="bold" />

            <Button
                android:id="@+id/btVar4"
                android:layout_width="150dp"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:background="#000000"
                android:onClick="share"
                android:text="Share to other apps"
                android:textColor="#FFFFFF"
                android:textSize="20sp"
                android:textStyle="bold" />
        </LinearLayout>

    </LinearLayout>
```

```
                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_marginLeft="15dp"
                    android:layout_marginTop="5dp"
                    android:layout_marginRight="15dp"
                    android:orientation="vertical">

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="left"
                        android:layout_marginTop="5dp"
                        android:gravity="left"
                        android:text="Secret key: "
                        android:textColor="#FFFFFF"
                        android:textSize="30dp"
                        android:textStyle="bold" />
                    <TextView
                        android:id="@+id/tvVar2"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="left"
                        android:layout_marginTop="5dp"
                        android:textColor="#FAFAFA"
                        android:textIsSelectable="true"
                        android:textSize="20dp" />
                    <!--button to copy encrypted code-->
                    <Button
                        android:id="@+id/sms"
                        android:layout_width="230dp"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center"
                        android:layout_marginTop="30dp"
                        android:background="#000000"
                        android:onClick="sms"
                        android:text="Send secret key"
                        android:textColor="#FFFFFF"
                        android:textSize="20sp"
                        android:textStyle="bold" />
                </LinearLayout>
    </LinearLayout>
</RelativeLayout>
```

Encrypt.java-

```
package com.example.encryptdecrypt;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.util.Base64;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;

public class Encrypt extends AppCompatActivity {
    private static final int PERMISSION_REQUEST_CODE = 1;

    TextView encryptedText,secretkey;
    ClipboardManager cp;
    EditText txt1;
    String K;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_encrypt);
        encryptedText = findViewById(R.id.tvVar1);
        txt1 = findViewById(R.id.etVar1);
        secretkey = findViewById(R.id.tvVar2);
        cp = (ClipboardManager) getSystemService(Context.CLIPBOARD_SERVICE);
    }

    @RequiresApi(api = Build.VERSION_CODES.O)
    public void enc(View v) throws NoSuchAlgorithmException,
IllegalBlockSizeException, InvalidKeyException,
            BadPaddingException, InvalidAlgorithmParameterException,
NoSuchPaddingException{
        String input = txt1.getText().toString();
        String[] arr = Encrypt.givenString_whenEncrypt_thenSuccess(input);
        Toast.makeText(this, "Encrypted Successfully!",
Toast.LENGTH_SHORT).show();
        encryptedText.setText(arr[0]);
        secretkey.setText(arr[1]);
        K = arr[1];

    }

    public void sms(View v){
        Intent i = new Intent(this,SMSActivity.class);
        Bundle b = new Bundle();
        b.putString("key",K);
        i.putExtras(b);
        startActivity(i);
    }
    public void copy(View v){
        String data = encryptedText.getText().toString().trim();
        if(!data.isEmpty()){
            ClipData temp = ClipData.newPlainText("text",data);
            cp.setPrimaryClip(temp);
            Toast.makeText(this,"Copied",Toast.LENGTH_SHORT).show();
        }
    }
    public void share(View v){
        Intent i = new Intent();
        i.setAction(Intent.ACTION_SEND);
        i.putExtra(Intent.EXTRA_TEXT,encryptedText.getText().toString());
        i.setType("text/plain");
        Intent shareIntent = Intent.createChooser(i,null);
        startActivity(shareIntent);
    }
    public static SecretKey generateKey(int n) throws NoSuchAlgorithmException {
        KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
        keyGenerator.init(n);
        SecretKey key = keyGenerator.generateKey();
        return key;
    }

    public static IvParameterSpec generateIv() {
        byte[] iv = new byte[16];
        new SecureRandom().nextBytes(iv);
        return new IvParameterSpec(iv);
    }

    @RequiresApi(api = Build.VERSION_CODES.O)
    public static String encrypt(String algorithm, String input, SecretKey key,
                                 IvParameterSpec iv) throws NoSuchPaddingException,
NoSuchAlgorithmException,
            InvalidAlgorithmParameterException, InvalidKeyException,
            BadPaddingException, IllegalBlockSizeException {

        Cipher cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.ENCRYPT_MODE, key, iv);
        byte[] cipherText = cipher.doFinal(input.getBytes());
        return Base64.getEncoder().encodeToString(cipherText);
    }

    @RequiresApi(api = Build.VERSION_CODES.O)
    public static String[] givenString_whenEncrypt_thenSuccess(String input)
            throws NoSuchAlgorithmException, IllegalBlockSizeException,
InvalidKeyException,
            BadPaddingException, InvalidAlgorithmParameterException,
NoSuchPaddingException {


        SecretKey key = Encrypt.generateKey(128);
        String seckey = Base64.getEncoder().encodeToString(key.getEncoded());

        //IvParameterSpec ivParameterSpec = Encrypt.generateIv();
        String encodedKey = "e7yGowuZf0W29W6oslUQAg==";
        byte[] decodedKey = Base64.getDecoder().decode(encodedKey);
        IvParameterSpec ivParameterSpec = new IvParameterSpec(decodedKey, 0,
decodedKey.length);
        String algorithm = "AES/CBC/PKCS5Padding";
        String cipherText = Encrypt.encrypt(algorithm, input, key, ivParameterSpec);
        return new String[]{cipherText, seckey};
        // String plainText = Encrypt.decrypt(algorithm, cipherText, key,
ivParameterSpec);
        // Assertions.assertEquals(input, plainText);
    }


}
```

Activity_decrypt.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="#f00"
    android:orientation="vertical"
    tools:context=".Decrypt">

    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="centerCrop"
        android:src="@drawable/background" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:orientation="vertical">

        <!--Textview to display title of the activity-->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Decryption"
            android:textColor="#FFFFFF"
            android:textSize="40dp"
            android:layout_marginBottom="5dp"
            android:textStyle="bold" />

    </LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:layout_margin="5dp"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="5dp"
        android:orientation="vertical">

        <!--view to create margin-->
        <View
            android:id="@+id/viewVar1"
            android:layout_width="wrap_content"
            android:layout_height="5dp"
            android:background="#fff" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="10dp"
        android:layout_marginRight="15dp"
        android:layout_marginBottom="50dp"
        android:orientation="vertical">

        <!--enter the code to be decrypted here-->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="left"
            android:layout_marginTop="5dp"
            android:text="Encrypted Text :"
            android:textColor="#FFFFFF"
            android:textSize="30dp"
            android:textStyle="bold" />
```

```
<EditText
    android:background="@drawable/shape"
    android:id="@+id/etVar1"
    android:layout_marginLeft="5dp"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_marginBottom="5dp"
    android:layout_gravity="center"
    android:paddingLeft="20dp"
    android:layout_marginTop="5dp"
    android:hint="  Enter the encrypted here"
    android:textColorHint="@color/white"
    android:textColor="#FFFFFF"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="left"
    android:layout_marginTop="5dp"
    android:text="Secret key :"
    android:textColor="#FFFFFF"
    android:textSize="30dp"
    android:textStyle="bold" />

<EditText
    android:id="@+id/etVar2"
    android:background="@drawable/shape"
    android:layout_marginLeft="5dp"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_gravity="center"
    android:layout_marginTop="5dp"
    android:paddingLeft="20dp"
    android:layout_marginBottom="5dp"
    android:hint="  Enter the secret key here"
    android:textColorHint="@color/white"
    android:textColor="#FFFFFF"
    android:textSize="20sp" />

<!--button to decrypt the code-->
<Button
    android:id="@+id/btVar1"
    android:layout_width="230dp"
    android:layout_height="wrap_content"
```

```
        android:layout_gravity="center"
        android:layout_marginTop="10dp"
        android:background="#000000"
        android:onClick="dec"
        android:text="Decrypt Text"
        android:textColor="#FFFFFF"
        android:textSize="20sp"
        android:textStyle="bold" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="15dp"
    android:orientation="vertical">

    <!--view to create margin-->
    <View
        android:id="@+id/viewVar2"
        android:layout_width="wrap_content"
        android:layout_height="5dp"
        android:background="#fff" />

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="5dp"
    android:layout_marginRight="15dp"
    android:orientation="vertical">

    <!--display the decrypted text here-->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="left"
        android:layout_marginTop="5dp"
        android:text="Your decrypted text : "
        android:textStyle="bold"
```

```
            android:textColor="#FFFFFF"
            android:textSize="30sp" />

        <TextView
            android:id="@+id/tvVar2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="left"
            android:layout_marginTop="5dp"
            android:textColor="#FFFFFF"
            android:layout_marginBottom="5dp"
            android:textSize="20sp" />

    </LinearLayout>

    <LinearLayout

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="5dp"
        android:layout_marginRight="15dp"
        android:orientation="vertical">

        <!--button to copy the decrypted text-->
        <Button
            android:id="@+id/btVar2"
            android:layout_width="230dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="#000000"
            android:layout_marginTop="5dp"
            android:onClick="copy1"
            android:text="Copy Text"
            android:textColor="#FFFFFF"
            android:textSize="20sp"
            android:textStyle="bold" />

    </LinearLayout>

  </LinearLayout>


</RelativeLayout>
```

Decrypt.java

```
package com.example.encryptdecrypt;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.security.SecureRandom;
import java.util.Base64;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class Decrypt extends AppCompatActivity {
    EditText txt1,txt2;
    TextView decryptedText;
    ClipboardManager cp;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_decrypt);
        txt1 = findViewById(R.id.etVar1);
        txt2 = findViewById(R.id.etVar2);
        decryptedText = findViewById(R.id.tvVar2);
        cp = (ClipboardManager) getSystemService(Context.CLIPBOARD_SERVICE);

    }
    @RequiresApi(api = Build.VERSION_CODES.O)
    public void dec(View v) throws NoSuchAlgorithmException,
IllegalBlockSizeException, InvalidKeyException,
            BadPaddingException, InvalidAlgorithmParameterException,
NoSuchPaddingException {
        String ciphertext = txt1.getText().toString();
        String secretkey = txt2.getText().toString();
        String plaintext =
Decrypt.givenString_whenEncrypt_thenSuccess(ciphertext,secretkey);
        Toast.makeText(this, "Decrypted successfully!",
Toast.LENGTH_SHORT).show();
        decryptedText.setText(plaintext);
    }
    public void copy1(View v){
        String data = decryptedText.getText().toString().trim();
        if(!data.isEmpty()){
            ClipData temp = ClipData.newPlainText("text",data);
            cp.setPrimaryClip(temp);
            Toast.makeText(this,"Copied",Toast.LENGTH_SHORT).show();
        }
    }
    public static SecretKey generateKey(int n) throws NoSuchAlgorithmException {
        KeyGenerator keyGenerator = KeyGenerator.getInstance("AES");
        keyGenerator.init(n);
        SecretKey key = keyGenerator.generateKey();
        return key;
    }



    @RequiresApi(api = Build.VERSION_CODES.O)
    public static String decrypt(String algorithm, String cipherText, SecretKey key,
                                 IvParameterSpec iv) throws NoSuchPaddingException,
NoSuchAlgorithmException,
            InvalidAlgorithmParameterException, InvalidKeyException,
            BadPaddingException, IllegalBlockSizeException {

        Cipher cipher = Cipher.getInstance(algorithm);
        cipher.init(Cipher.DECRYPT_MODE, key, iv);
        byte[] plainText = cipher.doFinal(Base64.getDecoder().decode(cipherText));
        return new String(plainText);
    }
    @RequiresApi(api = Build.VERSION_CODES.O)
    public static String givenString_whenEncrypt_thenSuccess(String
cipherText,String secretkey)
            throws NoSuchAlgorithmException, IllegalBlockSizeException,
InvalidKeyException,
            BadPaddingException, InvalidAlgorithmParameterException,
NoSuchPaddingException {

        byte[] decoKey = Base64.getDecoder().decode(secretkey);
        SecretKey key = new SecretKeySpec(decoKey, 0, decoKey.length, "AES");


        String encodedKey = "e7yGowuZf0W29W6oslUQAg==";
        byte[] decodedKey = Base64.getDecoder().decode(encodedKey);
        IvParameterSpec ivParameterSpec = new IvParameterSpec(decodedKey, 0,
decodedKey.length);
        String algorithm = "AES/CBC/PKCS5Padding";

         String plainText = Decrypt.decrypt(algorithm, cipherText, key, ivParameterSpec
);
        return plainText;
    }
}
```

Activity_smsactivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    tools:context=".SMSActivity"
    android:background="#f00"
    android:orientation="vertical">
    <ImageView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:scaleType="centerCrop"
        android:src="@drawable/background"
        />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:orientation="vertical">

        <!--title of the activity-->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="SMS"
            android:textColor="#FFFFFF"
            android:textSize="40sp"
            android:textStyle="bold" />

    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="15dp"
        android:layout_marginTop="10dp"
        android:layout_marginRight="15dp"
        android:layout_marginBottom="50dp"
        android:orientation="vertical"
        android:gravity="center"
        android:layout_centerInParent="true">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="left"
            android:layout_marginTop="5dp"
            android:text="Phone number :"
            android:textColor="#FFFFFF"
            android:textSize="30dp"
            android:textStyle="bold" />
        <EditText
            android:background="@drawable/shape"
            android:id="@+id/phoneNumber"
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:paddingLeft="20dp"
            android:hint="  Enter the phone number here"
            android:inputType="phone"
            android:textColor="@color/white"
            android:textColorHint="@color/white" />
        <TextView
```

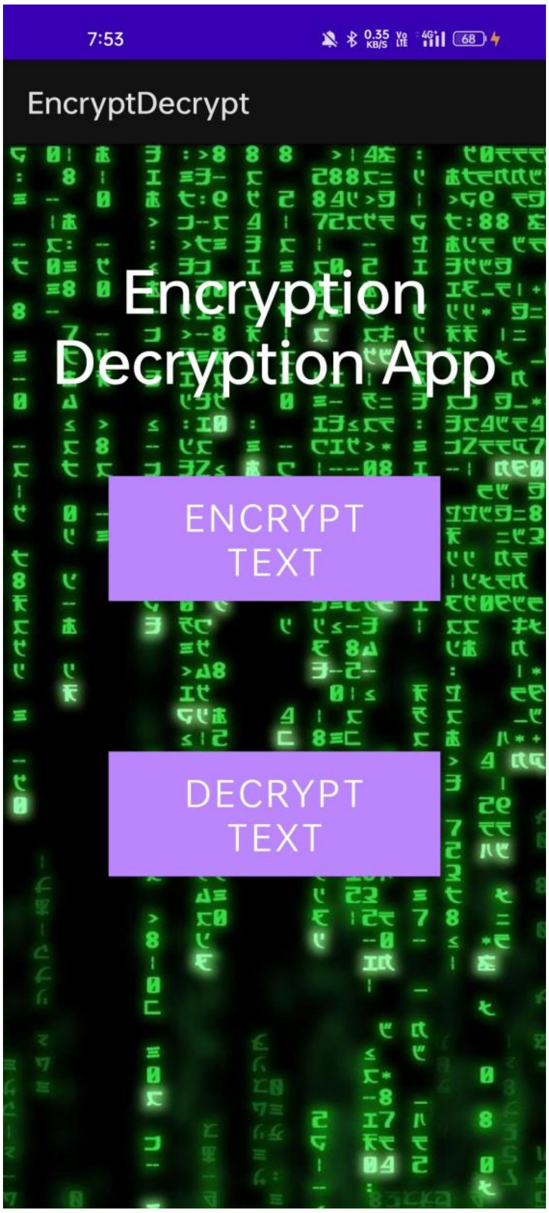
```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="left"
            android:layout_marginTop="5dp"
            android:text="Message :"
            android:textColor="#FFFFFF"
            android:textSize="30dp"
            android:textStyle="bold" />


        <EditText
            android:id="@+id/message"
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:layout_gravity="center"
            android:background="@drawable/shape"
            android:hint="  Enter the message here"
            android:paddingLeft="20dp"
            android:textColor="@color/white"
            android:textColorHint="@color/white" />

    <Button
        android:id="@+id/sendSMS"
        android:text="Send SMS"
        android:layout_gravity="center"
        android:layout_width="230dp"
        android:textSize="20dp"
        android:layout_height="wrap_content"/>
    </LinearLayout>

</RelativeLayout>
```

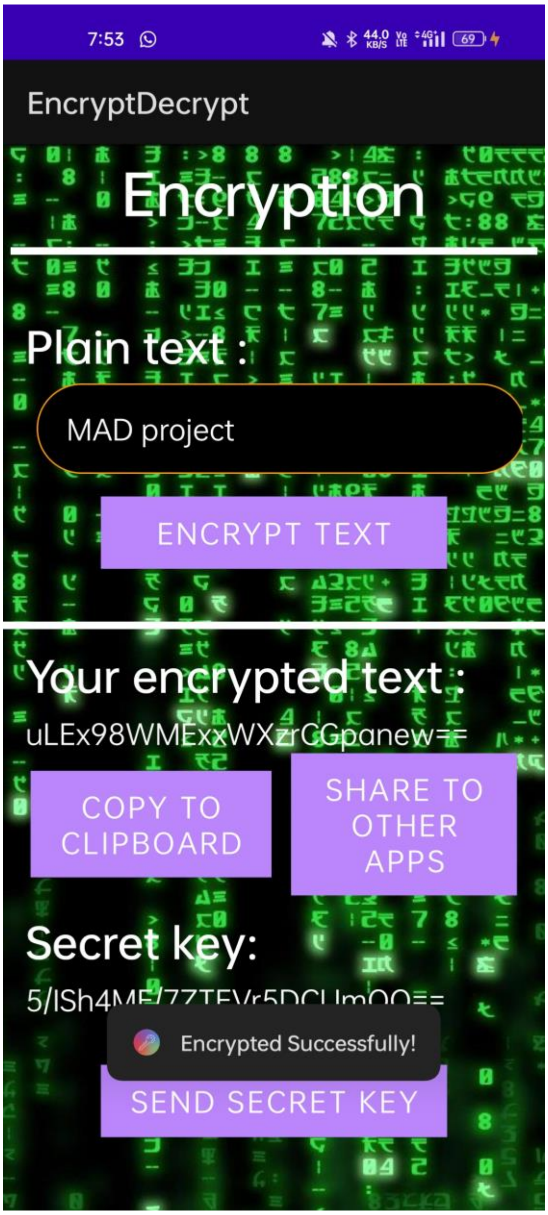
SMSActivity.java

```
package com.example.encryptdecrypt;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SMSActivity extends AppCompatActivity {
    private static final int PERMISSION_REQUEST_CODE = 1;
    private Button sendSMS;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_smsactivity);
        final EditText phoneNumber = (EditText) findViewById(R.id.phoneNumber);
        if (Build.VERSION.SDK_INT >= 23) {
            if (checkPermission()) {
                Log.e("permission", "Permission already granted.");
            } else {
                requestPermission();
            }
        }
        final EditText smsText = (EditText) findViewById(R.id.message);
        Bundle b = getIntent().getExtras();
        String key = b.getString("key");
        smsText.setText(key);
        sendSMS = (Button) findViewById(R.id.sendSMS);
        sendSMS.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {
                String sms = smsText.getText().toString();
                String phoneNum = phoneNumber.getText().toString();
                if(!TextUtils.isEmpty(sms) && !TextUtils.isEmpty(phoneNum)) {
                    if(checkPermission()) {

//Get the default SmsManager//

                        SmsManager smsManager = SmsManager.getDefault();

//Send the SMS//

                        smsManager.sendTextMessage(phoneNum, null, sms, null, null);
                        Toast.makeText(SMSActivity.this, "Message sent successfully",
Toast.LENGTH_SHORT).show();
                    }else {
                        Toast.makeText(SMSActivity.this, "Permission denied",
Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });

    }
    private boolean checkPermission() {
        int result = ContextCompat.checkSelfPermission(SMSActivity.this,
Manifest.permission.SEND_SMS);
        if (result == PackageManager.PERMISSION_GRANTED) {
            return true;
        } else {
            return false;
        }
    }
    private void requestPermission() {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.SEND_SMS}, PERMISSION_REQUEST_CODE);

    }
    @Override
    public void onRequestPermissionsResult(int requestCode, String permissions[],
int[] grantResults) {
        switch (requestCode) {
            case PERMISSION_REQUEST_CODE:
                if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {

                    Toast.makeText(SMSActivity.this,
                            "Permission accepted", Toast.LENGTH_LONG).show();

                } else {
                    Toast.makeText(SMSActivity.this,
                            "Permission denied", Toast.LENGTH_LONG).show();
                    Button sendSMS = (Button) findViewById(R.id.sendSMS);
                    sendSMS.setEnabled(false);
}
```

CHAPTER-5

SNAPSHOTS



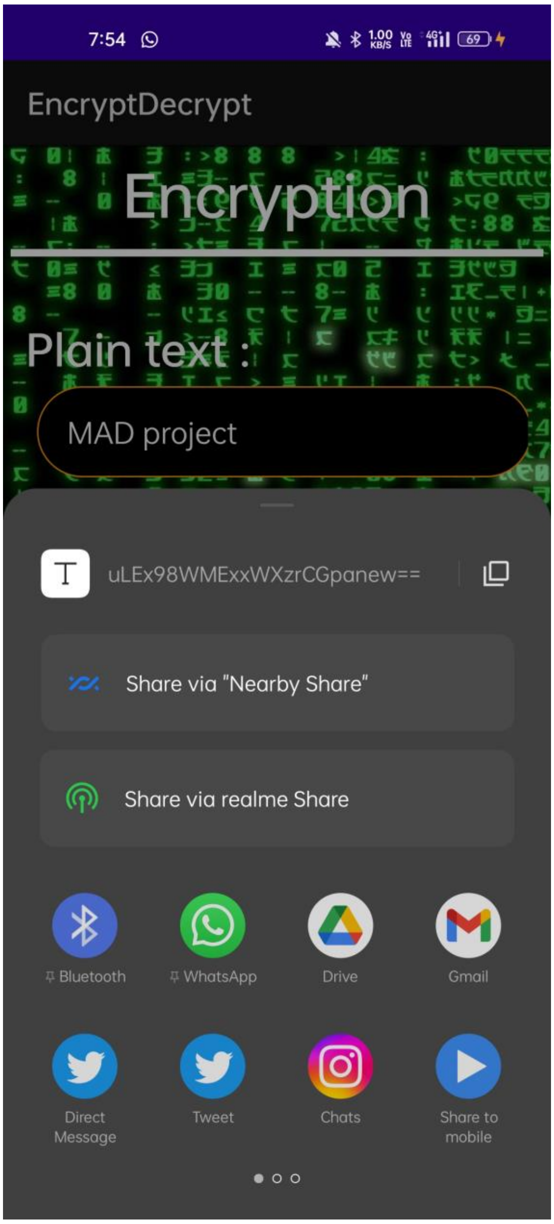


MAD project

uLEx98WMExxWXzrCGpanew==

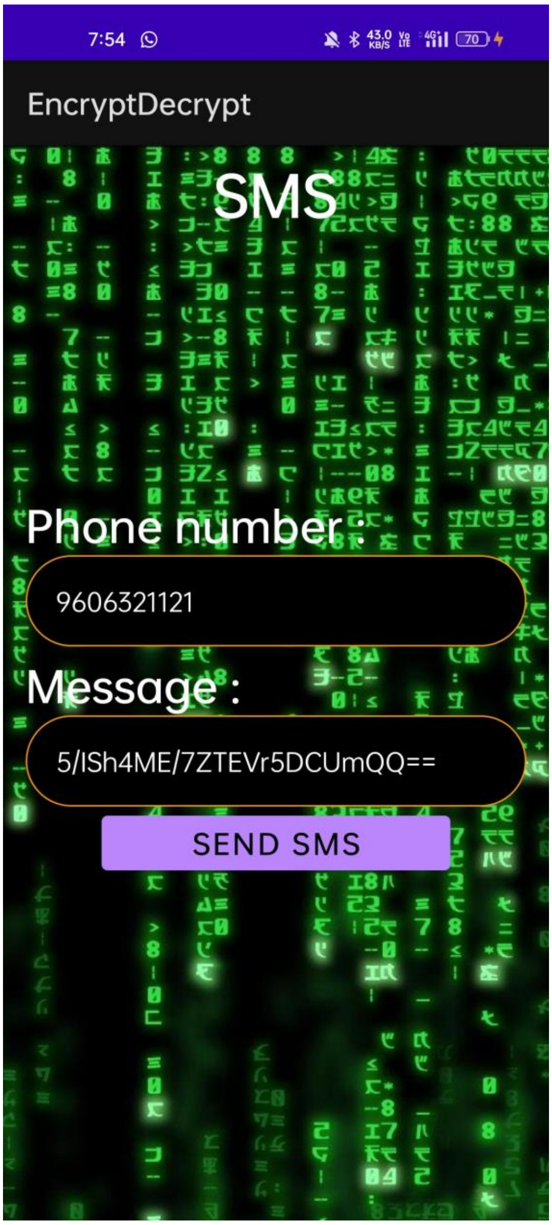
5/lSh4MF/77TFVr5DCLlmOO==

Encrypted Successfully!



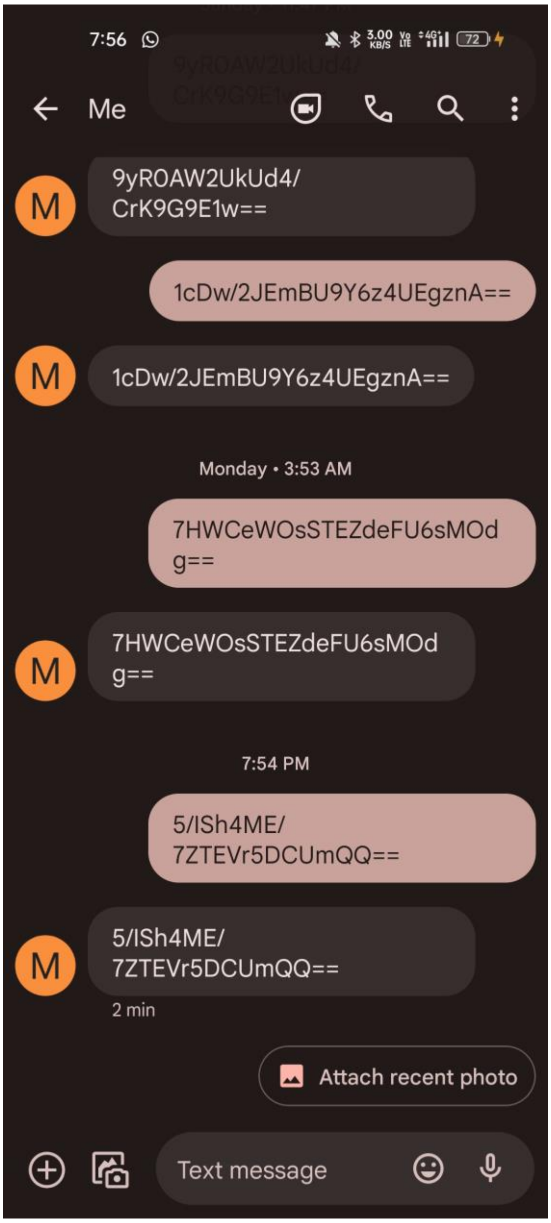
Plain text : MAD project

uLEx98WMExxWXzrCGpanew==



Phone number : 9606321121

Message : 5/ISh4ME/7ZTEVr5DCUmQQ==



9yROAW2UkUd4/
CrK9G9E1w==

1cDw/2JEmBU9Y6z4UEgznA==

1cDw/2JEmBU9Y6z4UEgznA==

Monday • 3:53 AM

7HWCeWOsSTEZdeFU6sMOd
g==

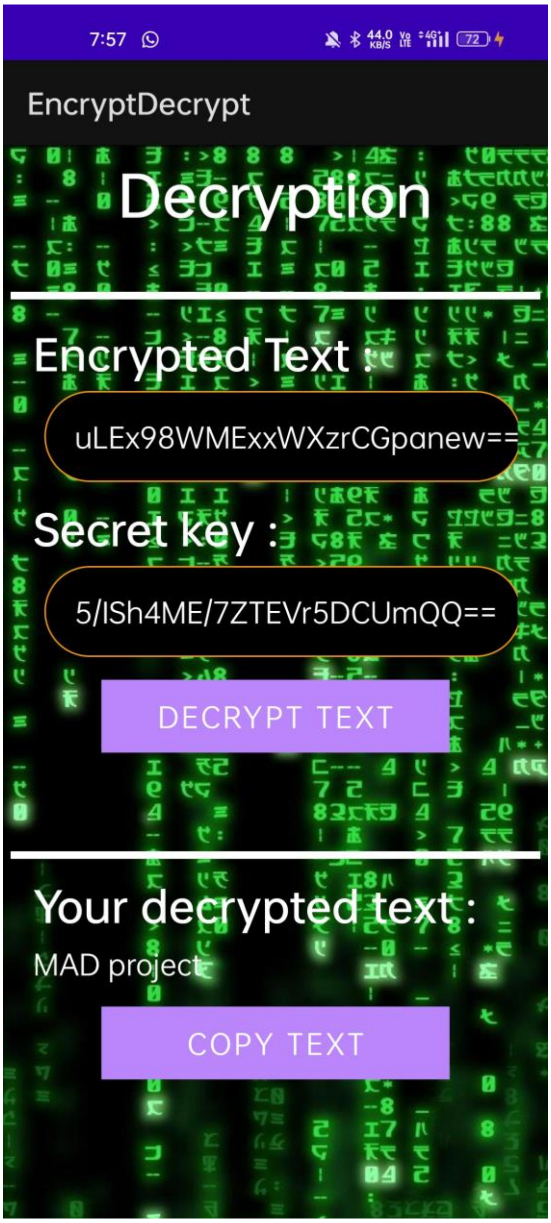
7HWCeWOsSTEZdeFU6sMOd
g==

7:54 PM

5/ISh4ME/
7ZTEVr5DCUmQQ==

5/ISh4ME/
7ZTEVr5DCUmQQ==

2 min



Encrypted Text :

uLEx98WMExxWXzrCGpanew==

Secret key :

5/ISh4ME/7ZTEVr5DCUmQQ==

Your decrypted text :

MAD project

CHAPTER-6

CONCLUSION

The Encryption Decryption App is a user-friendly app to people who require secure transfer of data. Secure transfer of data is nowadays a very important aspect of life. Overall system works without any bugs.

6.1 Future Enhancements

- In the future the same project can be enhanced in such a way that that the application is more interactive.
- Making Cloud Base Storage where user can upload data directly from the phone and share the secret key to the authorized user such that he can decrypt and use it.
- Add security enhancements and making the code more loosely coupled.

BIBLIOGRAPHY

- "Android studio": https://developer.android.com/studio
- “JDK”: www.oracle.com/java/downloads.
- “Advanced Encryption Standard[AES] algorithm”: https://www.geeksforgeeks.org/advanced-encryption-standard-aes/