CBE 411

Mobile Forensics And Security

LAB REPORT

NAME: K. REVANTH

ROLL NO: 2021BCY0037

Reverse Engineering the Created APK File

(**Cryptify App**)

Introduction:

The goal of this project was to perform reverse engineering on an Android APK, modify its encryption logic, and observe the effect of these modifications. The original APK contains an encryption mechanism using a static encryption key. By decompiling the APK, modifying the key, and implementing custom encryption and decryption functions, the system now uses different keys, resulting in different encrypted and decrypted outputs for the same plaintext.

App Used:

- **Android Studio:** An app used to create the APK file for the Cryptify app and emulate it.

Tools and Techniques Used:

- **APKTool:** A tool used for decompiling and recompiling APK files.
- **Zipalign:** A tool used to optimize the APK by aligning the APK file.
- **Keytool:** A utility for generating the keystore used for signing the APK.
- **Apksigner:** Used to sign the APK with the newly generated keystore.

Procedure:

I. CREATING APK FILE

1. Set Up a New Android Project

- Open **Android Studio**.
- Select "***Start a new Android Studio project***".

- Choose "***Empty Views Activity***" & name it (**EncryptionDemo**).
- Set the language to **Java** and finish creating the project.

2. Update activity_main.xml

- Design a simple UI for the app where users can input text to be encrypted/decrypted and a button to trigger encryption or decryption.

// EncryptionDemo/app/src/main/res/layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <!-- Title -->
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cryptify"
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_marginBottom="32dp"
        android:gravity="center" />

    <!-- Input Text Box with Border and Padding -->
    <EditText
        android:id="@+id/inputText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text to encrypt"
        android:inputType="textMultiLine"
        android:lines="5"
        android:layout_marginBottom="16dp"
        android:background="@drawable/border"
        android:paddingLeft="16dp" /> <!-- Padding to move
text to the right -->
```

```
    <!-- Encrypt Button -->
    <Button
        android:id="@+id/encryptButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Encrypt"
        android:layout_marginBottom="16dp" />

    <!-- Encrypted Text Box with Border and Padding -->
    <EditText
        android:id="@+id/encryptedText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Encrypted text will appear here"
        android:inputType="none"
        android:lines="5"
        android:layout_marginBottom="16dp"
        android:focusable="false"
        android:background="@drawable/border"
        android:paddingLeft="16dp" /> <!-- Padding to move
text to the right -->

    <!-- Decrypt Button -->
    <Button
        android:id="@+id/decryptButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Decrypt"
        android:layout_marginBottom="16dp" />

    <!-- Decrypted Text Box with Border and Padding -->
    <EditText
        android:id="@+id/decryptedText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Decrypted text will appear here"
        android:inputType="none"
        android:lines="5"
        android:focusable="false"
        android:background="@drawable/border"
        android:paddingLeft="16dp" /> <!-- Padding to move
text to the right -->

</LinearLayout>
```

3. Add Border Drawable (res/drawable/border.xml)

- Create a border style for the '*EditText*' components to make them look more defined. This file will define the border for the text boxes.

- Create a new XML file under '*res/drawable/*' named **border.xml** with the following content:

// EncryptionDemo/app/src/main/res/drawable/border.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FFFFFF"/>
    <stroke android:color="#000000" android:width="2dp"/>
    <corners android:radius="8dp"/>
</shape>
```

4. Java Code for Encryption/Decryption

- In your **MainActivity.java**, implement the *XOR encryption and decryption* logic, and wire up the buttons to perform the corresponding actions.

//EncryptionDemo/app/src/main/java/com.example.encryptiondemo/MainActivity.java

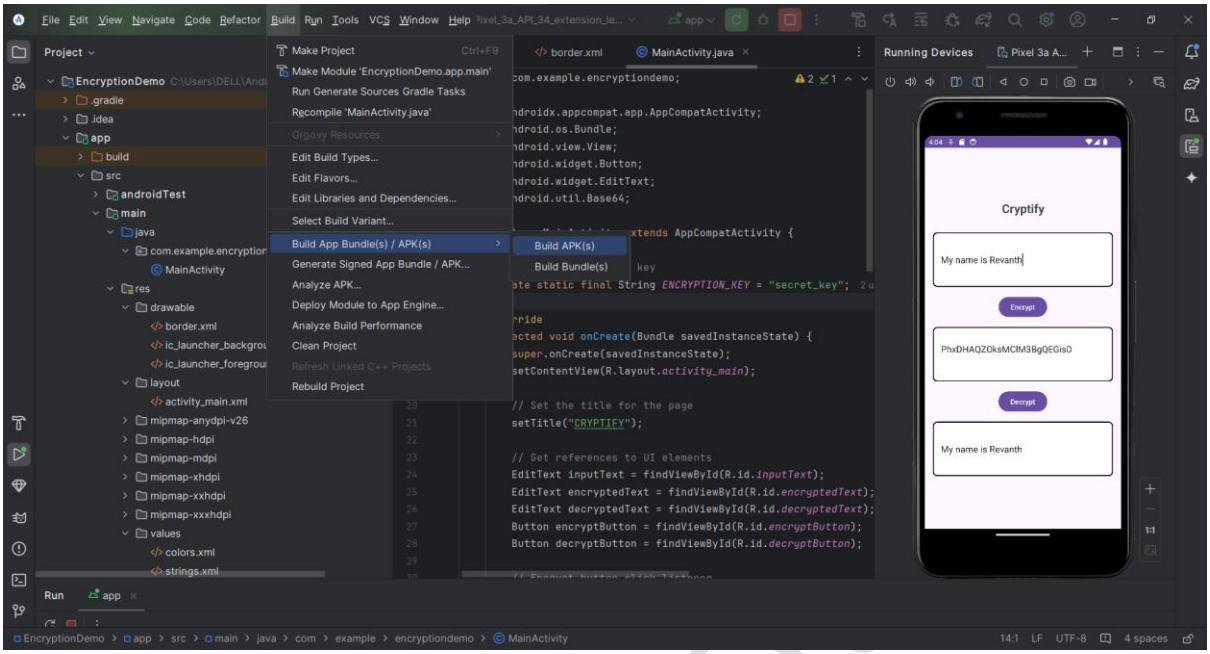
```
package com.example.encryptiondemo;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.util.Base64;

public class MainActivity extends AppCompatActivity {

    // Hardcoded encryption key
    private static final String ENCRYPTION_KEY =
```

```
"secret_key";

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

        // Set the title for the page
        setTitle("CRYPTIFY");

        // Get references to UI elements
        EditText inputText = findViewById(R.id.inputText);
        EditText encryptedText =
findViewById(R.id.encryptedText);
        EditText decryptedText =
findViewById(R.id.decryptedText);
        Button encryptButton =
findViewById(R.id.encryptButton);
        Button decryptButton =
findViewById(R.id.decryptButton);

        // Encrypt button click listener
        encryptButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String input =
inputText.getText().toString();
                String encrypted = encrypt(input); //
Encrypt the input text
                encryptedText.setText(encrypted); // Show
encrypted text in the encryptedText EditText
            }
        });

        // Decrypt button click listener
        decryptButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String encrypted =
encryptedText.getText().toString();
                String decrypted = decrypt(encrypted); //
Decrypt the encrypted text
                decryptedText.setText(decrypted); // Show
decrypted text in the decryptedText EditText
```

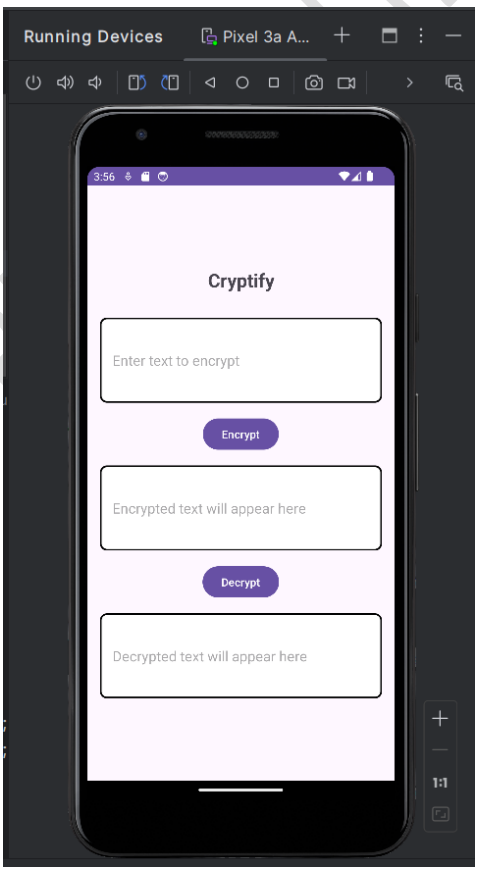
```
            }
        });
    }

    // XOR Encryption function with Base64 encoding
    private String encrypt(String input) {
        byte[] encryptedBytes =
xorOperation(input.getBytes(), ENCRYPTION_KEY.getBytes());
        return Base64.encodeToString(encryptedBytes,
Base64.DEFAULT); // Encode the encrypted bytes to Base64
    }

    // XOR Decryption function with Base64 decoding
    private String decrypt(String input) {
        byte[] decodedBytes = Base64.decode(input,
Base64.DEFAULT); // Decode the Base64 input
        byte[] decryptedBytes = xorOperation(decodedBytes,
ENCRYPTION_KEY.getBytes());
        return new String(decryptedBytes); // Convert bytes
back to string
    }

    // XOR Operation function
    private byte[] xorOperation(byte[] input, byte[] key) {
        byte[] result = new byte[input.length];
        for (int i = 0; i < input.length; i++) {
            result[i] = (byte) (input[i] ^ key[i %
key.length]);
        }
        return result;
    }
}
```

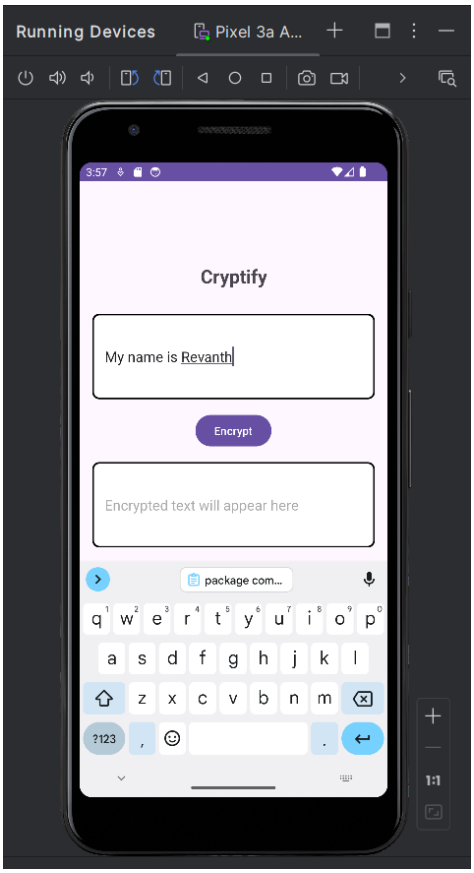
5. Build and Test the APK

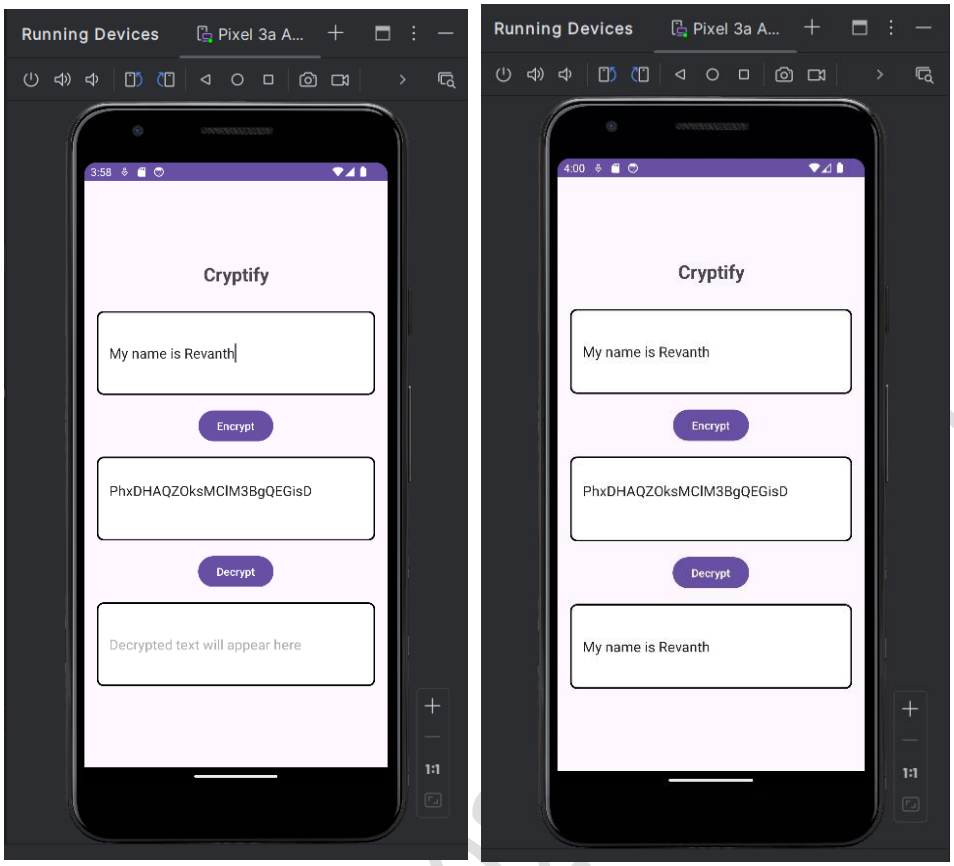
- Build the APK by going to **Build -> Build APK(s)** in Android Studio.
- Install the APK on an *Android device* or *emulator*.



- Test the functionality:
  - Enter text in the **input** box.
  - Click **Encrypt** and verify that the encrypted text is shown.
  - Click **Decrypt** and verify that the original text is restored.







II. REVERSE ENGINEERING THE APK FILE

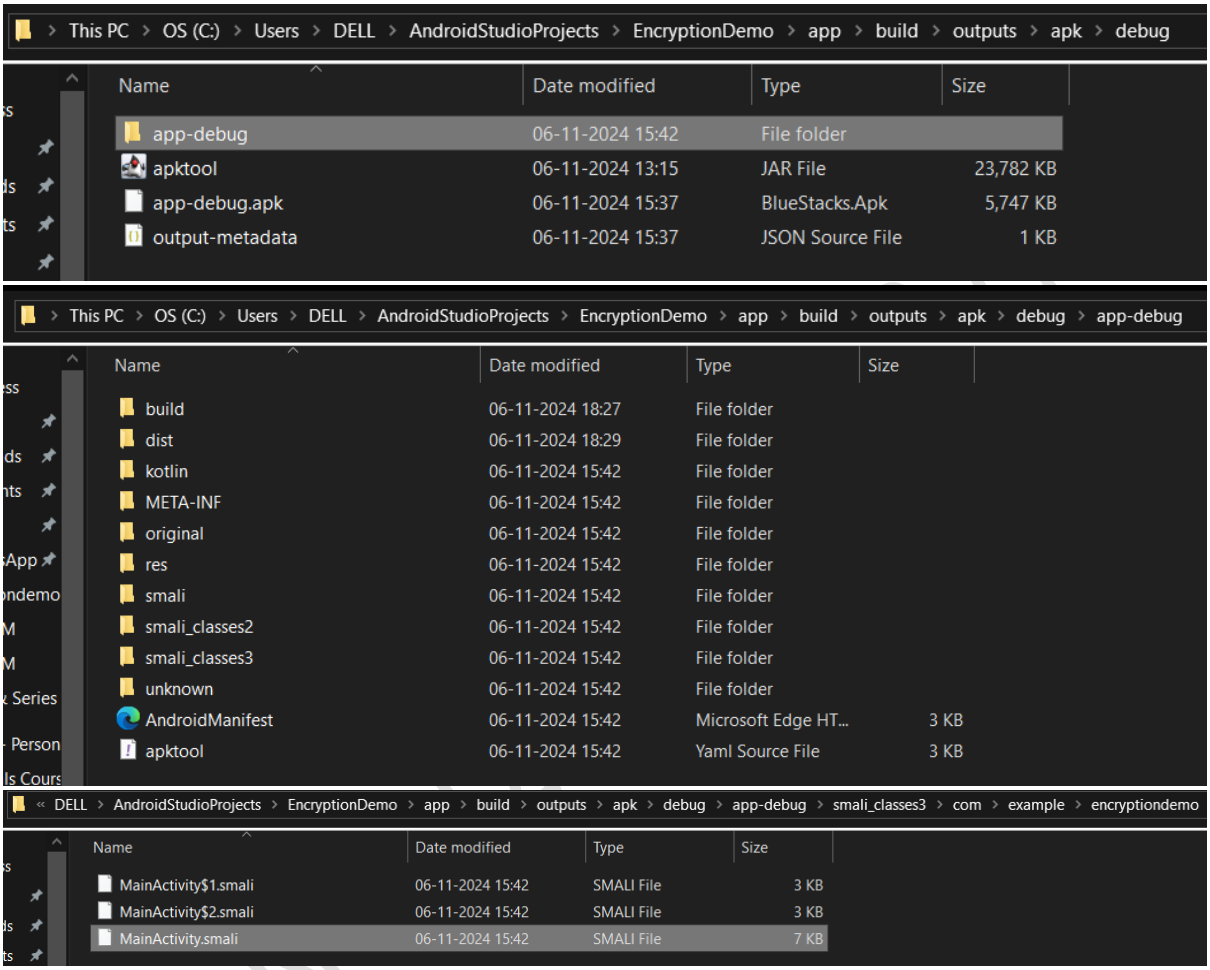
6. Decompiling the APK:

- The APK (app-debug.apk) was decompiled using APKTool:

`java -jar apktool.jar d app-debug.apk`

```
C:\Users\DELL\AndroidStudioProjects\EncryptionDemo\app\build\outputs\apk\debug>java -jar apktool.jar d app-debug.apk
I: Using Apktool 2.10.0 on app-debug.apk with 8 thread(s).
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Loading resource table...
I: Decoding file-resources...
I: Loading resource table from file: C:\Users\DELL\AppData\Local\apktool\framework\1.apk
I: Decoding values */* XMLs...
I: Decoding AndroidManifest.xml with resources...
I: Regular manifest package...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory
```

- This process extracted the smali files, which contain the low-level representation of the app's code.



Three smali files have been created; ***MainActivity.smali***, ***MainActivity$1.smali***, and ***MainActivity$2.smali***.

7. Identifying and modifying smali files:

// MainActivity.smali

```
.class public Lcom/example/encryptiondemo/MainActivity;
.super Landroidx/appcompat/app/AppCompatActivity;
.source "MainActivity.java"
```

```
# static fields

.field private static final ENCRYPTION_KEY:Ljava/lang/String;
= "secret_key"


# direct methods

.method public constructor <init>()V

    .locals 0

    .line 10

    invoke-direct {p0},
Landroidx/appcompat/app/AppCompatActivity;-><init>()V

    return-void

.end method


.method static synthetic
access$000(Lcom/example/encryptiondemo/MainActivity;Ljava/lang
/String;)Ljava/lang/String;

    .locals 1

    .param p0, "x0"    #
Lcom/example/encryptiondemo/MainActivity;

    .param p1, "x1"    # Ljava/lang/String;

    .line 10

    invoke-direct {p0, p1},
Lcom/example/encryptiondemo/MainActivity;-
>encrypt(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v0

    return-object v0

.end method


.method static synthetic
access$100(Lcom/example/encryptiondemo/MainActivity;Ljava/lang
/String;)Ljava/lang/String;

    .locals 1
```

```
    .param p0, "x0"    #
Lcom/example/encryptiondemo/MainActivity;

    .param p1, "x1"    # Ljava/lang/String;

    .line 10

    invoke-direct {p0, p1},
Lcom/example/encryptiondemo/MainActivity;-
>decrypt(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v0

    return-object v0

.end method


.method private decrypt(Ljava/lang/String;)Ljava/lang/String;

    .locals 3

    .param p1, "input"    # Ljava/lang/String;

    .line 59

    const/4 v0, 0x0

    invoke-static {p1, v0}, Landroid/util/Base64;-
>decode(Ljava/lang/String;I)[B

    move-result-object v0

    .line 60

    .local v0, "decodedBytes":[B

    const-string v1, "secret_key"

    invoke-virtual {v1}, Ljava/lang/String;->getBytes()[B

    move-result-object v1

    invoke-direct {p0, v0, v1},
Lcom/example/encryptiondemo/MainActivity;-
>xorOperation([B[B)[B

    move-result-object v1

    .line 61

    .local v1, "decryptedBytes":[B

    new-instance v2, Ljava/lang/String;

    invoke-direct {v2, v1}, Ljava/lang/String;-><init>([B)V
```

```
    return-object v2
.end method


.method private encrypt(Ljava/lang/String;)Ljava/lang/String;
    .locals 2
    .param p1, "input"    # Ljava/lang/String;
    .line 53
    invoke-virtual {p1}, Ljava/lang/String;->getBytes()[B
    move-result-object v0
    const-string v1, "secret_key"
    invoke-virtual {v1}, Ljava/lang/String;->getBytes()[B
    move-result-object v1
    invoke-direct {p0, v0, v1},
Lcom/example/encryptiondemo/MainActivity;-
>xorOperation([B[B)[B
    move-result-object v0
    .line 54
    .local v0, "encryptedBytes":[B
    const/4 v1, 0x0
    invoke-static {v0, v1}, Landroid/util/Base64;-
>encodeToString([BI)Ljava/lang/String;
    move-result-object v1
    return-object v1
.end method


.method private xorOperation([B[B)[B
    .locals 4
    .param p1, "input"    # [B
    .param p2, "key"    # [B
    .line 66
    array-length v0, p1
```

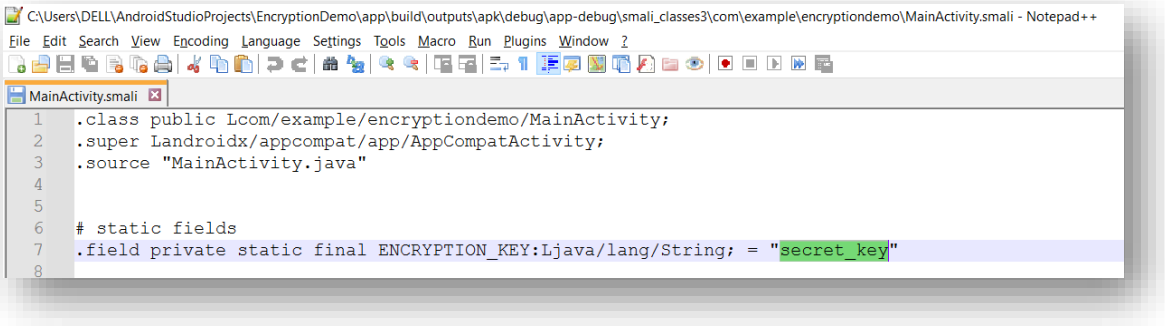
```
    new-array v0, v0, [B
    .line 67
    .local v0, "result":[B
    const/4 v1, 0x0
    .local v1, "i":I
    :goto_0
    array-length v2, p1
    if-ge v1, v2, :cond_0
    .line 68
    aget-byte v2, p1, v1
    array-length v3, p2
    rem-int v3, v1, v3
    aget-byte v3, p2, v3
    xor-int/2addr v2, v3
    int-to-byte v2, v2
    aput-byte v2, v0, v1
    .line 67
    add-int/lit8 v1, v1, 0x1
    goto :goto_0
    .line 70
    .end local v1    # "i":I
    :cond_0
    return-object v0
.end method

# virtual methods
.method protected onCreate(Landroid/os/Bundle;)V
    .locals 6
    .param p1, "savedInstanceState"    # Landroid/os/Bundle;
    .line 17
```

```
    invoke-super {p0, p1},
Landroidx/appcompat/app/AppCompatActivity;-
>onCreate(Landroid/os/Bundle;)V

    .line 18

    sget v0, Lcom/example/encryptiondemo/R$layout;-
>activity_main:I

    invoke-virtual {p0, v0},
Lcom/example/encryptiondemo/MainActivity;->setContentView(I)V

    .line 21

    const-string v0, "CRYPTIFY"

    invoke-virtual {p0, v0},
Lcom/example/encryptiondemo/MainActivity;-
>setTitle(Ljava/lang/CharSequence;)V

    .line 24

    sget v0, Lcom/example/encryptiondemo/R$id;->inputText:I

    invoke-virtual {p0, v0},
Lcom/example/encryptiondemo/MainActivity;-
>findViewById(I)Landroid/view/View;

    move-result-object v0

    check-cast v0, Landroid/widget/EditText;

    .line 25

    .local v0, "inputText":Landroid/widget/EditText;

    sget v1, Lcom/example/encryptiondemo/R$id;-
>encryptedText:I

    invoke-virtual {p0, v1},
Lcom/example/encryptiondemo/MainActivity;-
>findViewById(I)Landroid/view/View;

    move-result-object v1

    check-cast v1, Landroid/widget/EditText;

    .line 26

    .local v1, "encryptedText":Landroid/widget/EditText;

    sget v2, Lcom/example/encryptiondemo/R$id;-
>decryptedText:I
```

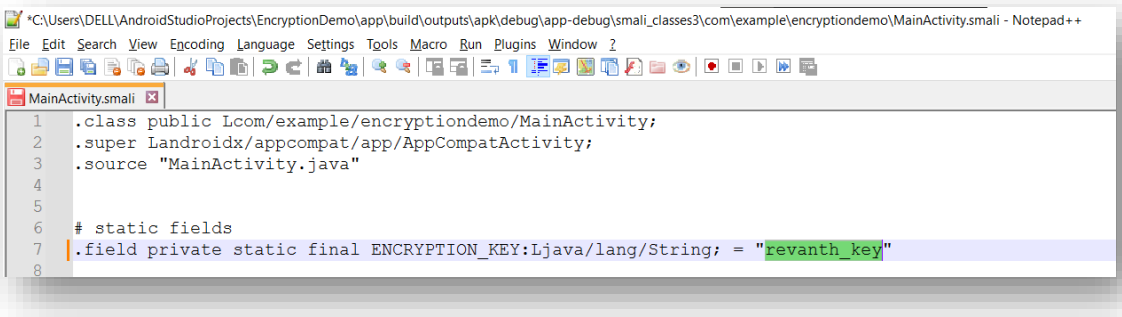
```
    invoke-virtual {p0, v2},
Lcom/example/encryptiondemo/MainActivity;-
>findViewById(I)Landroid/view/View;

    move-result-object v2

    check-cast v2, Landroid/widget/EditText;

    .line 27

    .local v2, "decryptedText":Landroid/widget/EditText;

    sget v3, Lcom/example/encryptiondemo/R$id;-
>encryptButton:I

    invoke-virtual {p0, v3},
Lcom/example/encryptiondemo/MainActivity;-
>findViewById(I)Landroid/view/View;

    move-result-object v3

    check-cast v3, Landroid/widget/Button;

    .line 28

    .local v3, "encryptButton":Landroid/widget/Button;

    sget v4, Lcom/example/encryptiondemo/R$id;-
>decryptButton:I

    invoke-virtual {p0, v4},
Lcom/example/encryptiondemo/MainActivity;-
>findViewById(I)Landroid/view/View;

    move-result-object v4

    check-cast v4, Landroid/widget/Button;

    .line 31

    .local v4, "decryptButton":Landroid/widget/Button;

    new-instance v5,
Lcom/example/encryptiondemo/MainActivity$1;

    invoke-direct {v5, p0, v0, v1},
Lcom/example/encryptiondemo/MainActivity$1;-
><init>(Lcom/example/encryptiondemo/MainActivity;Landroid/widg
et/EditText;Landroid/widget/EditText;)V

    invoke-virtual {v3, v5}, Landroid/widget/Button;-
>setOnClickListener(Landroid/view/View$OnClickListener;)V

    .line 41
```

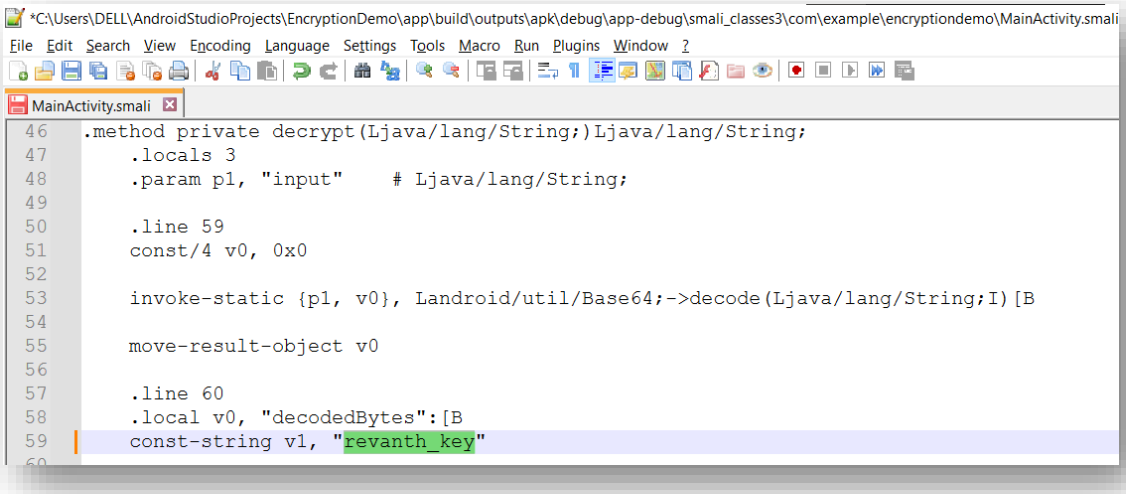
```
    new-instance v5,
Lcom/example/encryptiondemo/MainActivity$2;

    invoke-direct {v5, p0, v1, v2},
Lcom/example/encryptiondemo/MainActivity$2;-
><init>(Lcom/example/encryptiondemo/MainActivity;Landroid/widg
et/EditText;Landroid/widget/EditText;)V

    invoke-virtual {v4, v5}, Landroid/widget/Button;-
>setOnClickListener(Landroid/view/View$OnClickListener;)V

    .line 49

    return-void

.end method
```

Before modification:



After modification:





// MainActivity$1.smali

```
.class Lcom/example/encryptiondemo/MainActivity$1;

.super Ljava/lang/Object;

.source "MainActivity.java"


# interfaces

.implements Landroid/view/View$OnClickListener;



# annotations
```

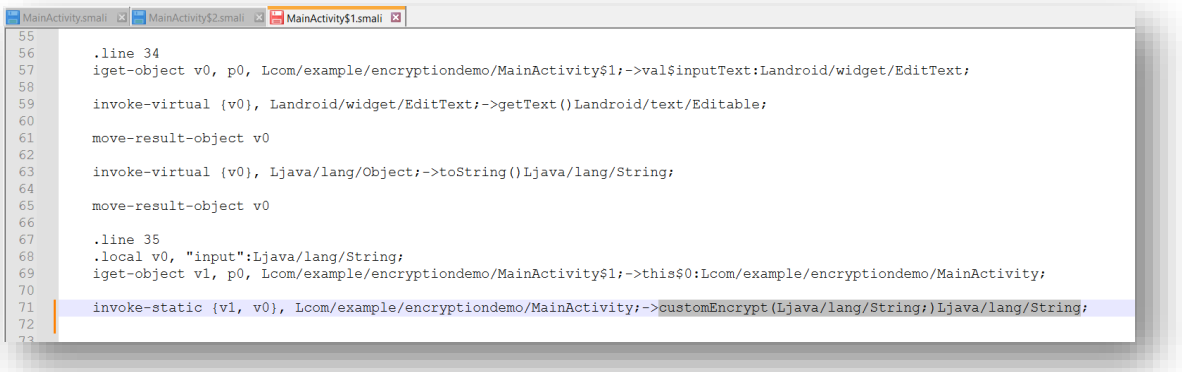
```
.annotation system Ldalvik/annotation/EnclosingMethod;
    value = Lcom/example/encryptiondemo/MainActivity;-
>onCreate(Landroid/os/Bundle;)V
.end annotation


.annotation system Ldalvik/annotation/InnerClass;
    accessFlags = 0x0
    name = null
.end annotation




# instance fields
.field final synthetic
this$0:Lcom/example/encryptiondemo/MainActivity;
.field final synthetic
val$encryptedText:Landroid/widget/EditText;
.field final synthetic val$inputText:Landroid/widget/EditText;




# direct methods
.method constructor
<init>(Lcom/example/encryptiondemo/MainActivity;Landroid/widge
t/EditText;Landroid/widget/EditText;)V
    .locals 0
    .param p1, "this$0"    #
Lcom/example/encryptiondemo/MainActivity;
    .annotation system Ldalvik/annotation/Signature;
        value = {
            "()V"
        }
    .end annotation
    .line 31
```

```
    iput-object p1, p0,
Lcom/example/encryptiondemo/MainActivity$1;-
>this$0:Lcom/example/encryptiondemo/MainActivity;

    iput-object p2, p0,
Lcom/example/encryptiondemo/MainActivity$1;-
>val$inputText:Landroid/widget/EditText;

    iput-object p3, p0,
Lcom/example/encryptiondemo/MainActivity$1;-
>val$encryptedText:Landroid/widget/EditText;

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void

.end method



# virtual methods

.method public onClick(Landroid/view/View;)V

    .locals 3

    .param p1, "v"    # Landroid/view/View;

    .line 34

    iget-object v0, p0,
Lcom/example/encryptiondemo/MainActivity$1;-
>val$inputText:Landroid/widget/EditText;

    invoke-virtual {v0}, Landroid/widget/EditText;-
>getText()Landroid/text/Editable;

    move-result-object v0

    invoke-virtual {v0}, Ljava/lang/Object;-
>toString()Ljava/lang/String;

    move-result-object v0

    .line 35

    .local v0, "input":Ljava/lang/String;

    iget-object v1, p0,
Lcom/example/encryptiondemo/MainActivity$1;-
>this$0:Lcom/example/encryptiondemo/MainActivity;
```

```
    invoke-static {v1, v0},
Lcom/example/encryptiondemo/MainActivity;-
>customEncrypt(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v1

    .line 36

    .local v1, "encrypted":Ljava/lang/String;

    iget-object v2, p0,
Lcom/example/encryptiondemo/MainActivity$1;-
>val$encryptedText:Landroid/widget/EditText;

    invoke-virtual {v2, v1}, Landroid/widget/EditText;-
>setText(Ljava/lang/CharSequence;)V

    .line 37

    return-void

.end method
```

Before modification:

After modification:



// MainActivity$2.smali

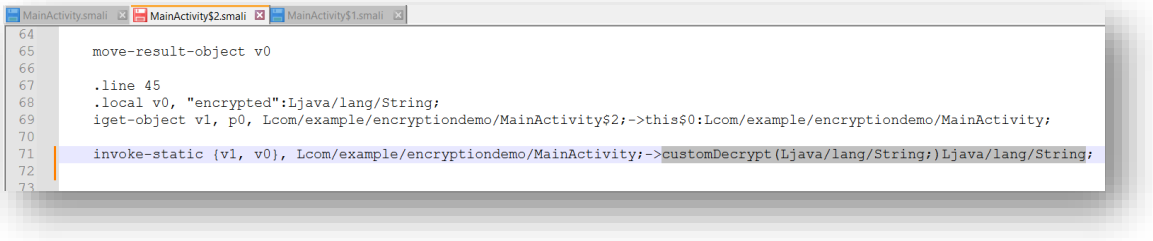
```
.class Lcom/example/encryptiondemo/MainActivity$2;

.super Ljava/lang/Object;

.source "MainActivity.java"


# interfaces

.implements Landroid/view/View$OnClickListener;




# annotations

.annotation system Ldalvik/annotation/EnclosingMethod;

    value = Lcom/example/encryptiondemo/MainActivity;-
>onCreate(Landroid/os/Bundle;)V

.end annotation


.annotation system Ldalvik/annotation/InnerClass;

    accessFlags = 0x0

    name = null

.end annotation




# instance fields

.field final synthetic
this$0:Lcom/example/encryptiondemo/MainActivity;

.field final synthetic
val$decryptedText:Landroid/widget/EditText;

.field final synthetic
val$encryptedText:Landroid/widget/EditText;




# direct methods
```

```
.method constructor
<init>(Lcom/example/encryptiondemo/MainActivity;Landroid/widge
t/EditText;Landroid/widget/EditText;)V

    .locals 0

    .param p1, "this$0"    #
Lcom/example/encryptiondemo/MainActivity;

    .annotation system Ldalvik/annotation/Signature;

        value = {

            "()V"

        }

    .end annotation

    .line 41

    iput-object p1, p0,
Lcom/example/encryptiondemo/MainActivity$2;-
>this$0:Lcom/example/encryptiondemo/MainActivity;

    iput-object p2, p0,
Lcom/example/encryptiondemo/MainActivity$2;-
>val$encryptedText:Landroid/widget/EditText;

    iput-object p3, p0,
Lcom/example/encryptiondemo/MainActivity$2;-
>val$decryptedText:Landroid/widget/EditText;

    invoke-direct {p0}, Ljava/lang/Object;-><init>()V

    return-void

.end method



# virtual methods

.method public onClick(Landroid/view/View;)V

    .locals 3

    .param p1, "v"    # Landroid/view/View;

    .line 44

    iget-object v0, p0,
Lcom/example/encryptiondemo/MainActivity$2;-
>val$encryptedText:Landroid/widget/EditText;
```

```
    invoke-virtual {v0}, Landroid/widget/EditText;-
>getText()Landroid/text/Editable;

    move-result-object v0

    invoke-virtual {v0}, Ljava/lang/Object;-
>toString()Ljava/lang/String;

    move-result-object v0

    .line 45

    .local v0, "encrypted":Ljava/lang/String;

    iget-object v1, p0,
Lcom/example/encryptiondemo/MainActivity$2;-
>this$0:Lcom/example/encryptiondemo/MainActivity;

    invoke-static {v1, v0},
Lcom/example/encryptiondemo/MainActivity;-
>customDecrypt(Ljava/lang/String;)Ljava/lang/String;

    move-result-object v1

    .line 46

    .local v1, "decrypted":Ljava/lang/String;

    iget-object v2, p0,
Lcom/example/encryptiondemo/MainActivity$2;-
>val$decryptedText:Landroid/widget/EditText;

    invoke-virtual {v2, v1}, Landroid/widget/EditText;-
>setText(Ljava/lang/CharSequence;)V

    .line 47

    return-void

.end method
```

Before modification:

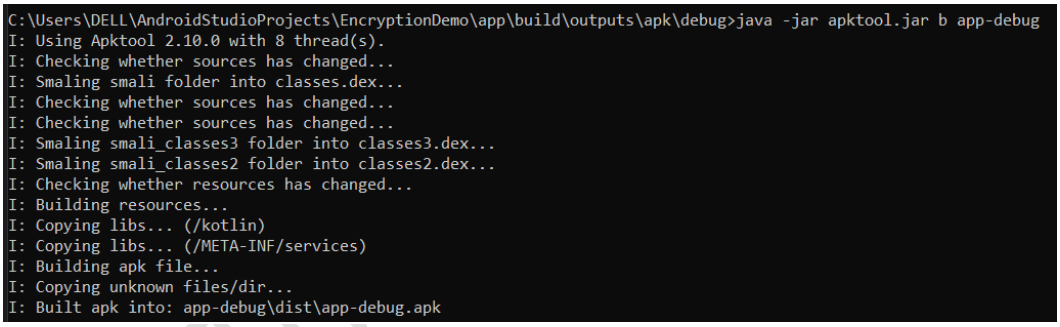
After modification:



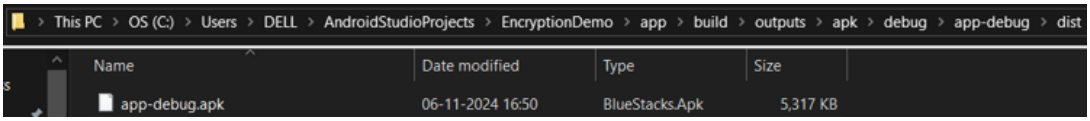
8. Rebuilding the APK:

- The APK was rebuilt with the modified smali files using APKTool:

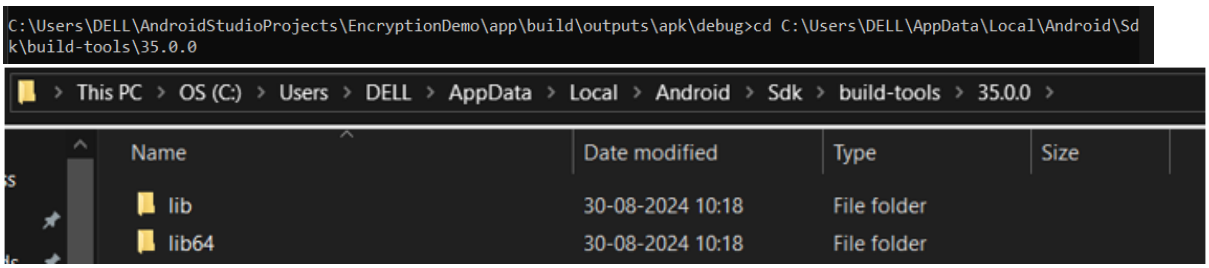
  `java -jar apktool.jar b app-debug`

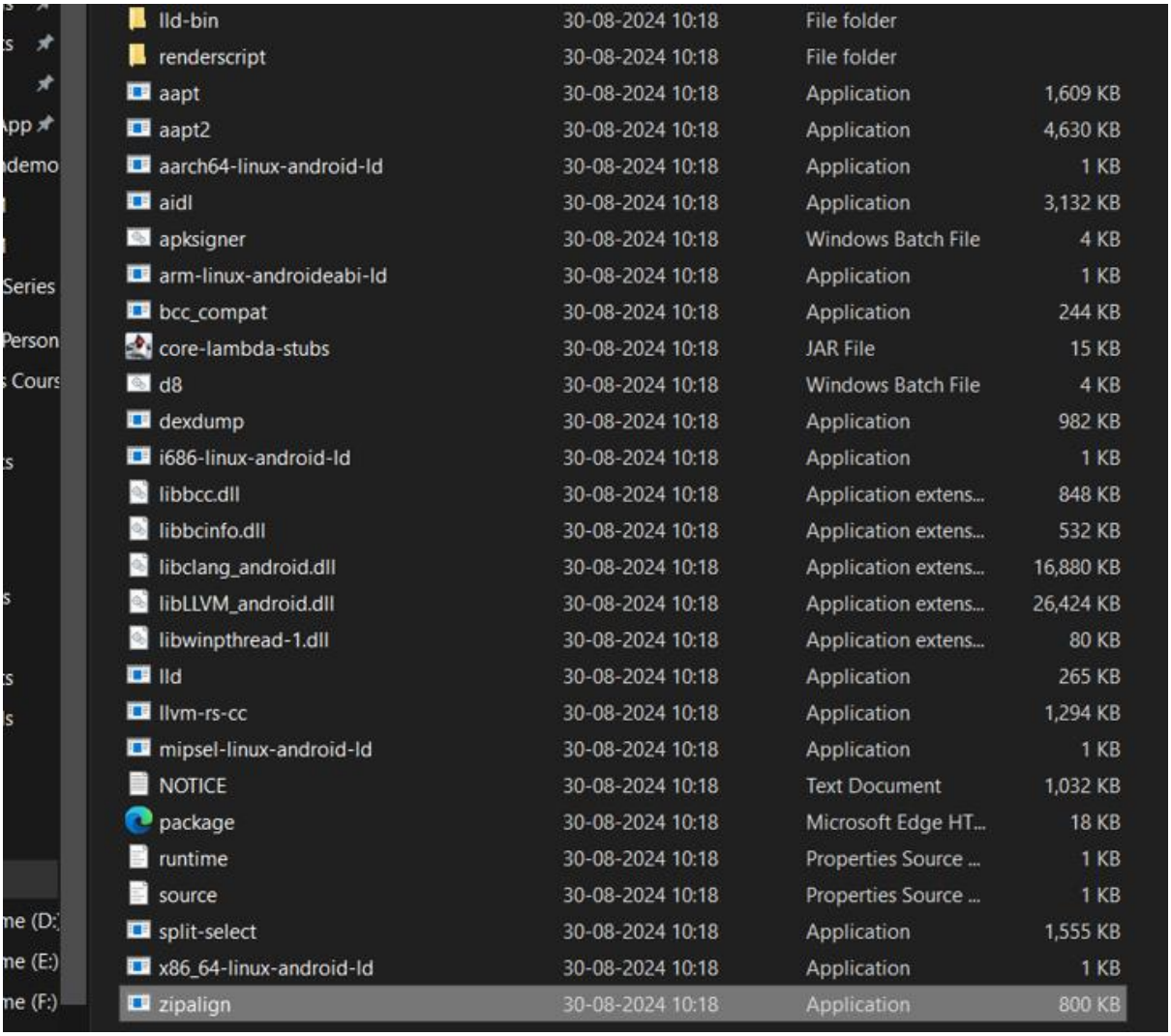


This generated a new APK with the updated logic and resources.



9. Aligning the APK:





lld-bin	30-08-2024 10:18	File folder	
renderscript	30-08-2024 10:18	File folder	
aapt	30-08-2024 10:18	Application	1,609 KB
aapt2	30-08-2024 10:18	Application	4,630 KB
aarch64-linux-android-ld	30-08-2024 10:18	Application	1 KB
aidl	30-08-2024 10:18	Application	3,132 KB
apksigner	30-08-2024 10:18	Windows Batch File	4 KB
arm-linux-androideabi-ld	30-08-2024 10:18	Application	1 KB
bcc_compat	30-08-2024 10:18	Application	244 KB
core-lambda-stubs	30-08-2024 10:18	JAR File	15 KB
d8	30-08-2024 10:18	Windows Batch File	4 KB
dexdump	30-08-2024 10:18	Application	982 KB
i686-linux-android-ld	30-08-2024 10:18	Application	1 KB
libbcc.dll	30-08-2024 10:18	Application extens...	848 KB
libbcinfo.dll	30-08-2024 10:18	Application extens...	532 KB
libclang_android.dll	30-08-2024 10:18	Application extens...	16,880 KB
libLLVM_android.dll	30-08-2024 10:18	Application extens...	26,424 KB
libwinpthread-1.dll	30-08-2024 10:18	Application extens...	80 KB
lld	30-08-2024 10:18	Application	265 KB
llvm-rs-cc	30-08-2024 10:18	Application	1,294 KB
mipsel-linux-android-ld	30-08-2024 10:18	Application	1 KB
NOTICE	30-08-2024 10:18	Text Document	1,032 KB
package	30-08-2024 10:18	Microsoft Edge HT...	18 KB
runtime	30-08-2024 10:18	Properties Source ...	1 KB
source	30-08-2024 10:18	Properties Source ...	1 KB
split-select	30-08-2024 10:18	Application	1,555 KB
x86_64-linux-android-ld	30-08-2024 10:18	Application	1 KB
zipalign	30-08-2024 10:18	Application	800 KB

- The APK was aligned using the Zipalign tool to optimize its size and performance:

```
zipalign -v 4 app-debug.apk app-debug-aligned.apk
```

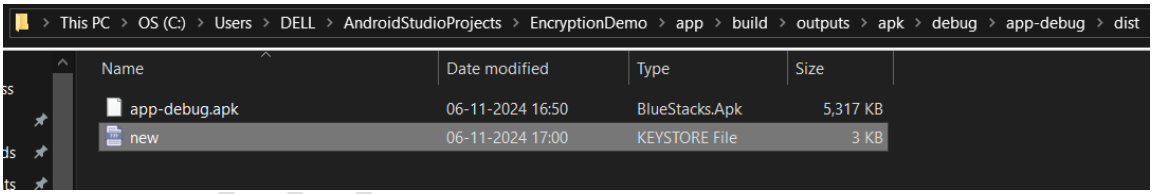
```
C:\Users\DELL\AppData\Local\Android\Sdk\build-tools\35.0.0>zipalign -v 4 C:\Users\DELL\AndroidStudioProjects\EncryptionD
emo\app\build\outputs\apk\debug\app-debug\dist\app-debug.apk
Zip alignment utility
Copyright (C) 2009 The Android Open Source Project

Usage: zipalign [-f] [-p] [-P <pagesize_kb>] [-v] [-z] <align> infile.zip outfile.zip
       zipalign -c [-p] [-P <pagesize_kb>] [-v] <align> infile.zip

  <align>: alignment in bytes, e.g. '4' provides 32-bit alignment
  -c: check alignment only (does not modify file)
  -f: overwrite existing outfile.zip
  -p: 4kb page-align uncompressed .so files
  -v: verbose output
  -z: recompress using Zopfli
  -P <pagesize_kb>: Align uncompressed .so files to the specified
                    page size. Valid values for <pagesize_kb> are 4, 16
                    and 64. '-P' cannot be used in combination with '-p'.
```

10. Signing the APK:

- A new keystore is generated using Keytool:

```
keytool -genkey -v -keystore new.keystore -alias app-debug -keyalg RSA -keysize 2048 -validity 20000
```

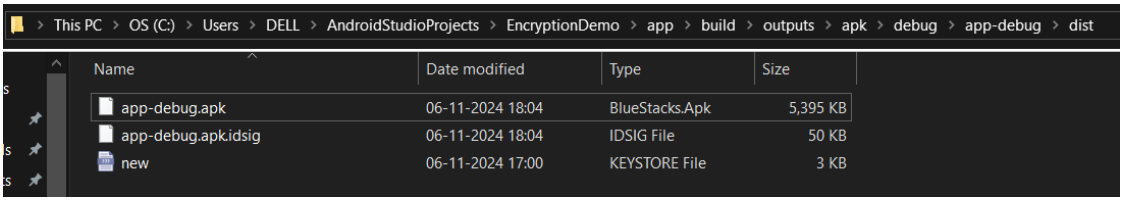
```
C:\Users\DELL\AppData\Local\Android\Sdk\build-tools\35.0.0>keytool -genkey -v -keystore C:\Users\DELL\AndroidStudioProje
cts\EncryptionDemo\app\build\outputs\apk\debug\app-debug\dist\new.keystore -alias app-debug -keyalg RSA -keysize 2048 -v
alidity 20000
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Revanth Kondabathula
What is the name of your organizational unit?
  [Unknown]:  IIIT Kottayam
What is the name of your organization?
  [Unknown]:  Cyber Security
What is the name of your City or Locality?
  [Unknown]:  Kottayam
What is the name of your State or Province?
  [Unknown]:  Kerala
What is the two-letter country code for this unit?
  [Unknown]:  IN
Is CN=Revanth Kondabathula, OU=IIIT Kottayam, O=Cyber Security, L=Kottayam, ST=Kerala, C=IN correct?
  [no]:  Yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 20,000 days
        for: CN=Revanth Kondabathula, OU=IIIT Kottayam, O=Cyber Security, L=Kottayam, ST=Kerala, C=IN
[Storing C:\Users\DELL\AndroidStudioProjects\EncryptionDemo\app\build\outputs\apk\debug\app-debug\dist\new.keystore]
```



- The APK was signed with the newly generated keystore using Apksigner:

```
apksigner sign --ks new.keystore --v1-signing-enabled true --v2-signing-enabled true app-debug-aligned.apk
```

```
C:\Users\DELL\AppData\Local\Android\Sdk\build-tools\35.0.0>apksigner sign --ks C:\Users\DELL\AndroidStudioProjects\Encry
ptionDemo\app\build\outputs\apk\debug\app-debug\dist\new.keystore --v1-signing-enabled true --v2-signing-enabled true C:
\Users\DELL\AndroidStudioProjects\EncryptionDemo\app\build\outputs\apk\debug\app-debug\dist\app-debug.apk
Keystore password for signer #1:
```



11. Running the new APK

- First, go to Android Studio and go to its Terminal.

- Relocate to the path where the old adb device was present.

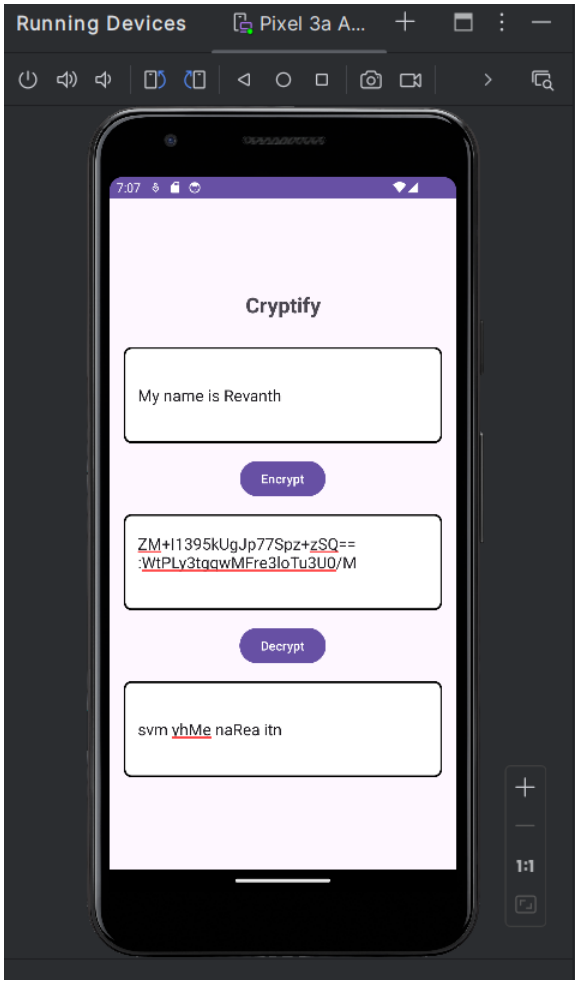
  `cd C:\Users\DELL\AppData\Local\Android\Sdk\platform-tools`

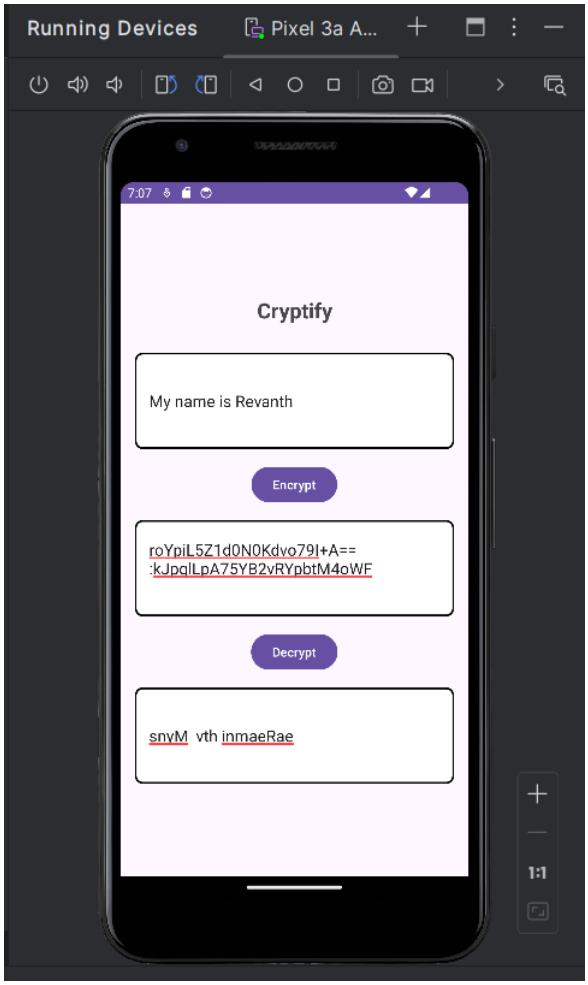
```
PS C:\Users\DELL\AndroidStudioProjects\EncryptionDemo> cd C:\Users\DELL\AppData\Local\Android\Sdk\platform-tools
```

- Now uninstall the old adb device.
  `.\adb uninstall com.example.encryptiondemo`

```
PS C:\Users\DELL\AppData\Local\Android\Sdk\platform-tools> .\adb uninstall com.example.encryptiondemo
Success
```

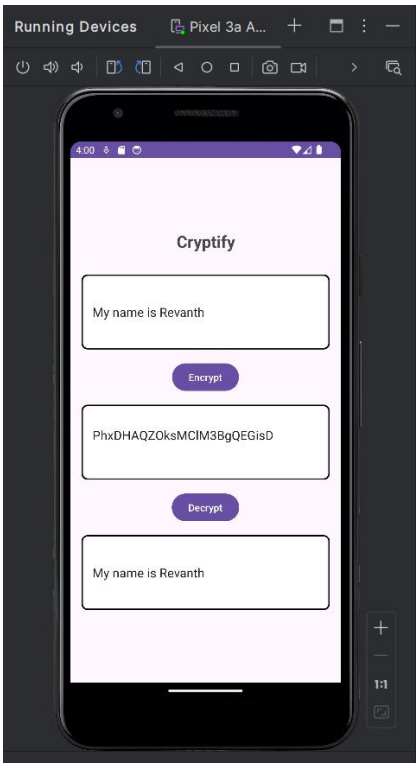
- Drag the new apk file and place it in the emulator and run it.



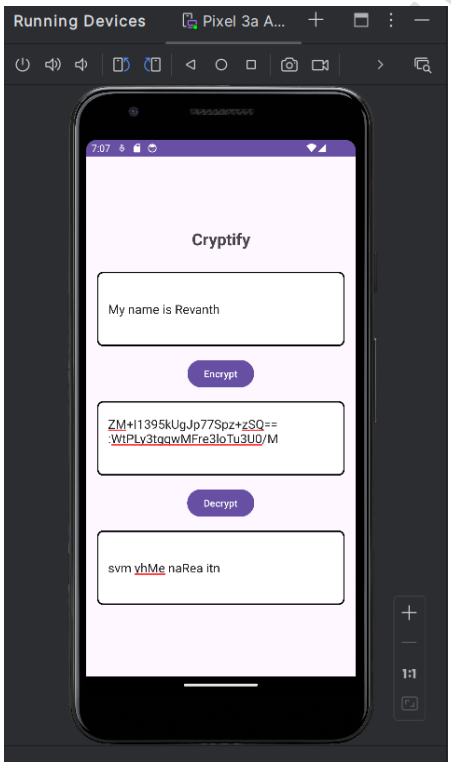


Results:

1. **Different Encrypted Outputs:**
    - When the same plaintext was encrypted with the new key and IV, the output encrypted text was different compared to the original APK. This demonstrates the effectiveness of the key modification.



(Original APK)



(Reverse Engineered APK)

    - The decryption process also returns a different result, ensuring that the new encryption mechanism is working as expected.

2. **Final Output:**
    - The modified APK behaves as expected: when the same plaintext is input, the resulting encrypted and decrypted texts are different from the original.

Conclusion:

The reverse engineering and modification of the encryption mechanism in the APK was successful. By changing the encryption key and implementing custom encryption and decryption methods, the system now provides different encrypted and decrypted outputs for the same plaintext input. This modification enhances the security of the application, as it makes the encryption harder to reverse-engineer with the default static key.

This report highlights the process of reverse engineering, modifying encryption logic, and testing the APK in a detailed and professional manner.