A Project Report

on

Securing a message using AES Encrypto app

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Technology in Computer Science & Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. Kavita
Associate Professor

Submitted By

Jayant Joshi
18SCSE1010510

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

JULY, 2021-2022

SCHOOL OF COMPUTING SCIENCE AND - ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"SECURING A MESSAGE USING AES ENCRYPTO APP"** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **July 2021** to **December 2021**, under the supervision of **Dr. Kavita Associate Professor, Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Jayant Joshi,18SCSE1010510

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Kavita

Associate Professor

<u>**CERTIFICATE**</u>

The Final Thesis/Project/ Dissertation Viva-Voce examination of **Jayant Joshi: 18SCSE1010510** has been held on _________________ and his/her work is recommended for the award of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.**

Signature of Examiner(s) **Signature of Supervisor(s)**

Date: December, 2021

Place: Greater Noida

Abstract

As we all know that, everyday we pass we loose some bit of our social security in this tech world and as the time passes the Security measures and IT rules are getting more and more stricter these days. This is a good thing for us but if we see it in the other way they are eating out on our personal information slowly and without us being of it.

And to comply with these issues and to have some secret conversation, communication or sending or receiving any information which can't be by any chance exposed to the public or anyone except its owners we made an application for our personal well being,

So keeping in mind the Advanced Encryption Standards algorithm (also known as one of the most difficult encryption algorithm to breach into) I have tried to create an Android Application which can encrypt and decrypt our messages and secret codes with one unique key.

This application is currently only being in use for a personal businesses and works of delivering messages and can successfully encrypt and decrypt messages using AES algorithm with its unique key but in the near future where we are losing every bit of our personal information daily this application could be a really helpful method to keep a bit of our information to ourselves.

Table of Contents

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
SCSE.	School of Computing Science and Engineering
AES	Advanced Encryption Standard
DES	Data Encryption Standard
DFD	Data Flow Diagram

CHAPTER-1

Introduction

The idea of making this application is not that much unique but its really useful these days. We all have heard of various data leaks and information leaks from various different big companies like WhatsApp, Domino’s etc.

You can see on the news or different platforms that your information shared on these platforms are leaked and are shared publicly. Although the amount of data leaks are very few on WhatsApp but on Domino’s literally everything is out in public and there was this domain through which you can access your information online. You have also heard of You tube accounts getting hacked these days and what not.

Although we say WhatsApp is secured through end to end encryption so no one could see our chats but there is a way to access these chats and because not everything is end to end secured. The messages are secured on the sender’s and the receiver’s side but when a chat is being backed up to our cloud storage it is not secured with end to end encryption and that is the place where hacker’s target to get that message information from the person.

And also you would have noticed that 4-5 months back there were some changes in the IT rules for social media platforms from the Ministry of Information and Technology where every social media platform has to appoint a specific set of people who will have to right to monitor your social status, conversations and every information on that platform.

Although it was for our safety but as you can see this is clear violations of right to privacy and free speech, particularly in the absence of the robust data protection law. But still every social media platform agreed to this law as they had no other choice but to agree to the government norms and policies.

So everyone is trying to get into our privacy and personal space so what if we want to have some private or super secret conversation which we don’t want anyone else to know so in that situation we can use this application as AES algorithm is one of the best and only the sender and receiver has to know the key which is not a very difficult task. So Since our platform is private only so it won’t be monitored by anyone and we can encrypt and decrypt our messages there while sharing that messages in any social media application,

Technologies Used:

- Android XML : Page layout has been designed in Android XML.
- Android : This project has been developed over the Android Platform.
- Java : All the coding has been written in Java.
- Encryption algorithm: Advanced Encryption Standards(AES) algorithm.
- Android Studio : We have used Android Studio for developing the project.

CHAPTER-2 Introduction

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Points to remember:

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each

.
That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

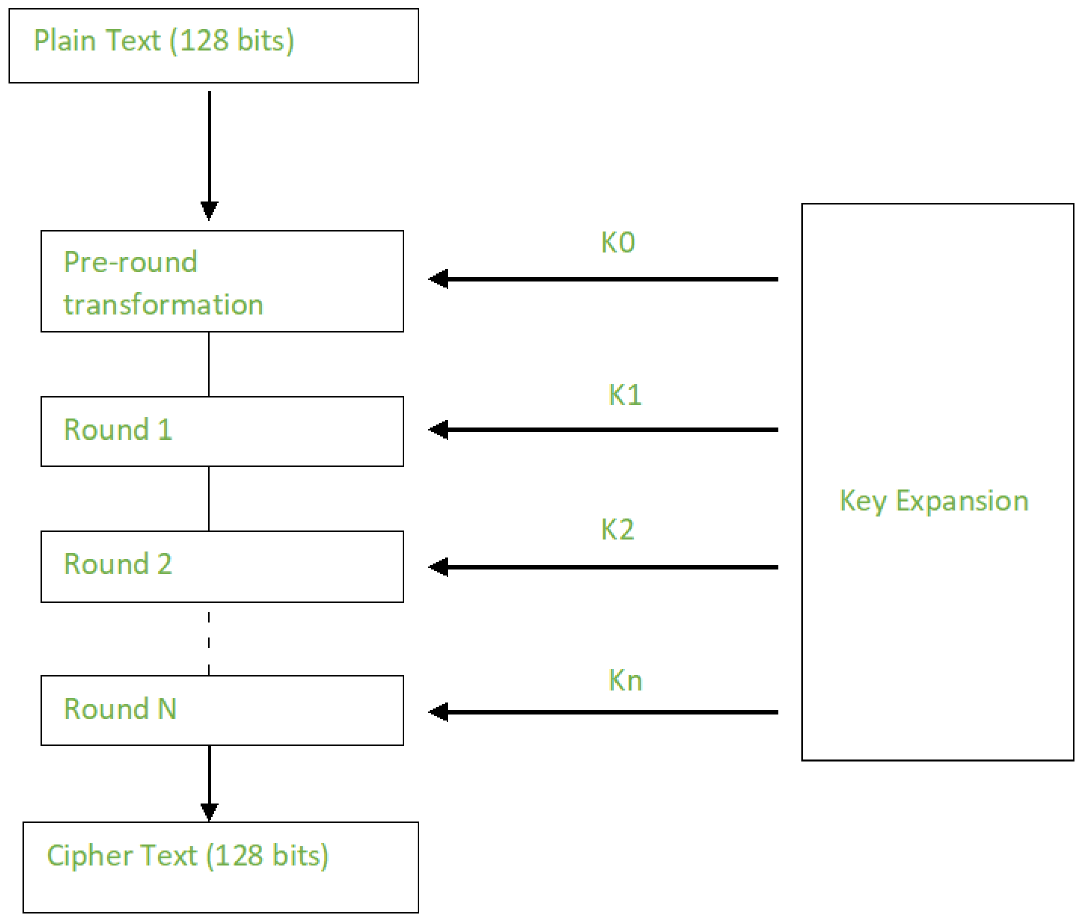
Working of the cipher :
AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows :

- 128 bit key – 10 rounds
- 192 bit key – 12 rounds
- 256 bit key – 14 rounds

Creation of Round keys :
A Key Schedule algorithm is used to calculate all the round keys from the key.

So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.



Encryption :

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```

Each round comprises of 4 steps :

- SubBytes

- ShiftRows
- MixColumns
- Add Round Key

The last round doesn’t have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes :

This step implements the substitution.

In this step each byte is substituted by another byte.(Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

ShiftRows :

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

```
[ b0  | b1  | b2  | b3  ]         [ b0  | b1  | b2  | b3  ]
| b4  | b5  | b6  | b7  |    ->   | b5  | b6  | b7  | b4  |
| b8  | b9  | b10 | b11 |         | b10 | b11 | b8  | b9  |
[ b12 | b13 | b14 | b15 ]         [ b15 | b12 | b13 | b14 ]
```

MixColumns :

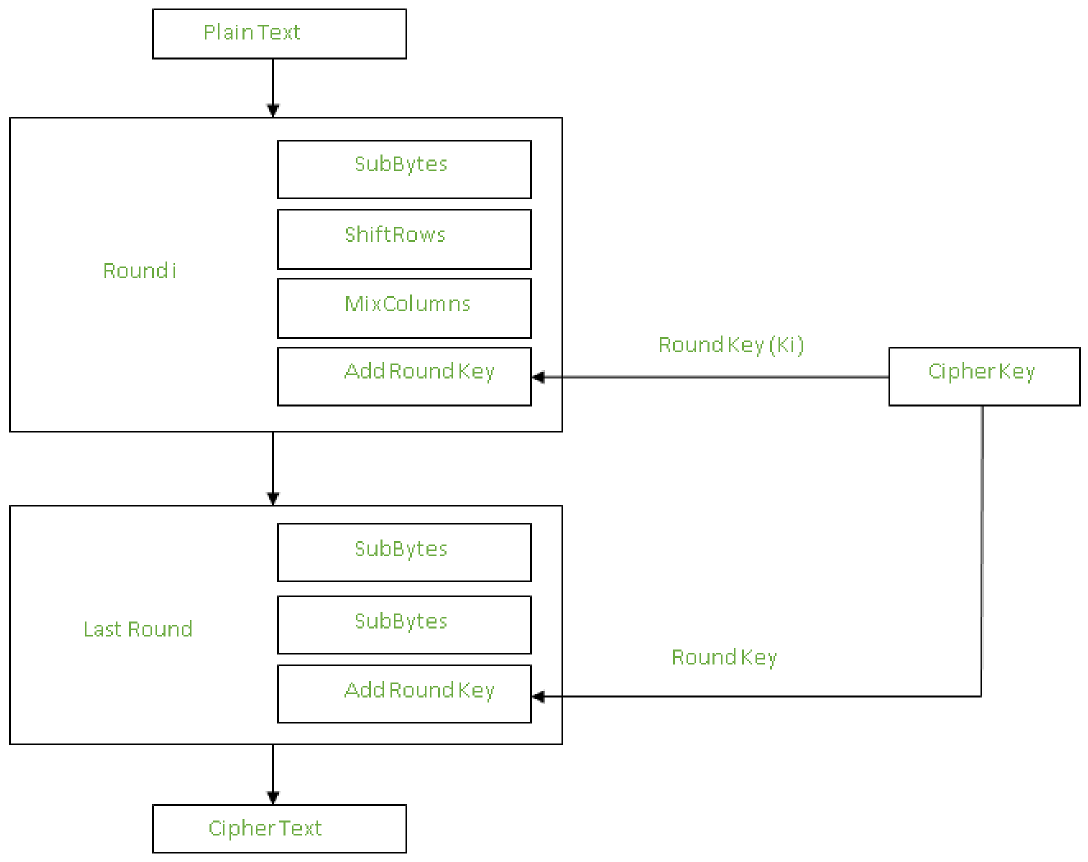
This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

```
[ c0 ]          [ 2   3   1   1 ]  [ b0 ]
| c1 |   =      | 1   2   3   1 |      | b1 |
| c2 |       | 1   1   2   3 |      | b2 |
[ c3 ]       [ 3   1   1   2 ]      [ b3 ]
```

Add Round Keys :

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.



After all these rounds 128 bits of encrypted data is given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Decryption :

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes. Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse MixColumns :

This step is similar to the MixColumns step in encryption, but differs in the matrix used to carry out the operation.

```
[ b0 ]        [ 14  11  13  9  ]  [ c0 ]
| b1 |  =     | 9   14  11  13 |     | c1 |
| b2 |      | 13  9   14  11 |     | c2 |
[ b3 ]        [ 11  13  9   14 ]     [ c3 ]
```

Inverse SubBytes :

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

Summary :

AES instruction set is now integrated into the CPU (offers throughput of several

GB/s)to improve the speed and security of applications that use AES for encryption and decryption. Even though its been 20 years since its introduction we have failed to break the AES algorithm as it is infeasible even with the current technology. Till date the only vulnerability remains in the implementation of the algorithm.

ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support

- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

This page provides an introduction to basic Android Studio features. For a summary of the latest changes, see Android Studio release notes.

Project structure

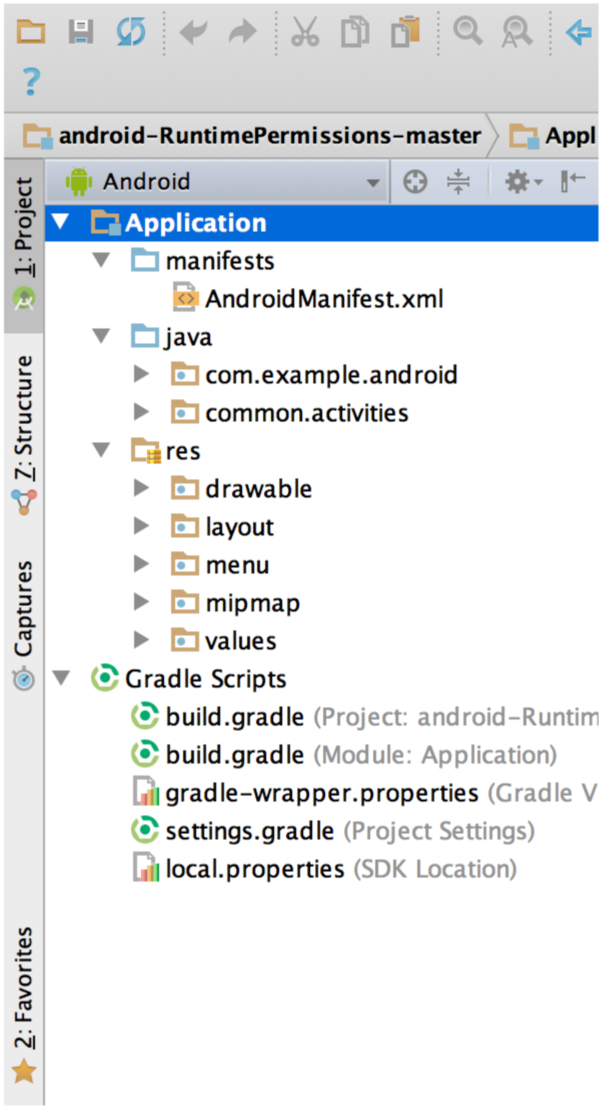

android-RuntimePermissions-master
Appl
Android
1: Project
7: Structure
Captures
2: Favorites
Application
manifests
AndroidManifest.xml
java
com.example.android
common.activities
res
drawable
layout
menu
mipmap
values
Gradle Scripts
build.gradle (Project: android-Runtim
build.gradle (Module: Application)
gradle-wrapper.properties (Gradle V
settings.gradle (Project Settings)
local.properties (SDK Location)

Figure 1. The project files in Android view.

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

- **manifests**: Contains the AndroidManifest.xml file.
- **java**: Contains the Java source code files, including JUnit test code.
- **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in figure 1, it's showing as **Android**).

You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the **Problems** view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

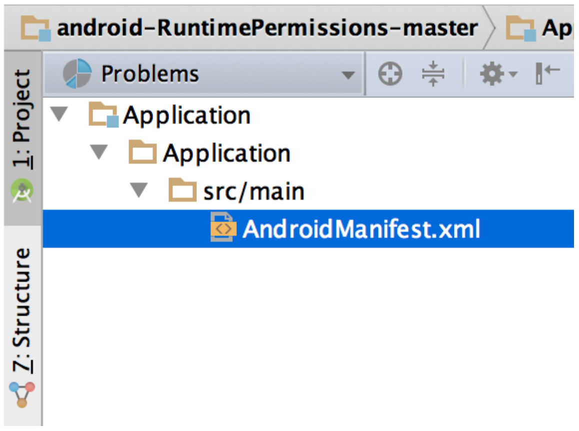
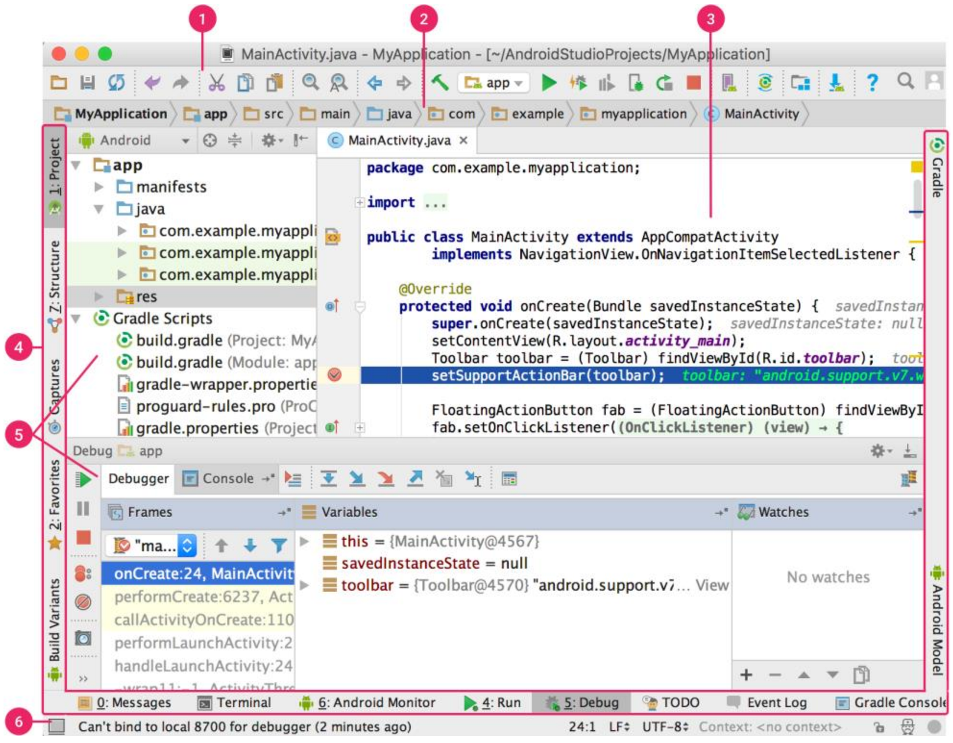


Figure 2. The project files in Problems view, showing a layout file with a problem.

For more information, see Projects overview.

The user interface

The Android Studio main window is made up of several logical areas identified in figure 3.



```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {  savedInstan
        super.onCreate(savedInstanceState);  savedInstanceState: null
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  tool
        setSupportActionBar(toolbar);  toolbar: "android.support.v7.w

        FloatingActionButton fab = (FloatingActionButton) findViewByI
        fab.setOnClickListener((OnClickListener) (view) → {
```

```
"ma...
onCreate:24, MainActivit
performCreate:6237, Act
callActivityOnCreate:110
performLaunchActivity:2
handleLaunchActivity:24
```

```
this = {MainActivity@4567}
savedInstanceState = null
toolbar = {Toolbar@4570} "android.support.v7... View
```

```
Can't bind to local 8700 for debugger (2 minutes ago)    24:1  LF÷  UTF-8÷  Context: <no context>
```

Figure 3. The Android Studio main window.

1. The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
2. The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
3. The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.

5. The **tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
6. The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

You can organize the main window to give yourself more screen space by hiding or moving toolbars and tool windows. You can also use keyboard shortcuts to access most IDE features.

At any time, you can search across your source code, databases, actions, elements of the user interface, and so on, by double-pressing the Shift key, or clicking the magnifying glass in the upper right-hand corner of the Android Studio window. This can be very useful if, for example, you are trying to locate a particular IDE action that you have forgotten how to trigger.

Tool windows

Instead of using preset perspectives, Android Studio follows your context and automatically brings up relevant tool windows as you work. By default, the most commonly used tool windows are pinned to the tool window bar at the edges of the application window.

- To expand or collapse a tool window, click the tool's name in the tool window bar. You can also drag, pin, unpin, attach, and detach tool windows.

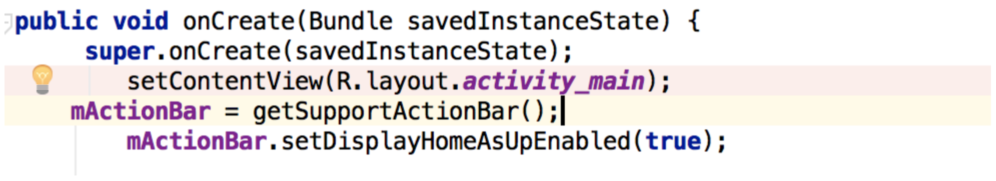
- To return to the current default tool window layout, click **Window > Restore Default Layout** or customize your default layout by clicking **Window > Store Current Layout as Default**.
- To show or hide the entire tool window bar, click the window icon in the bottom left-hand corner of the Android Studio window.
- To locate a specific tool window, hover over the window icon and select the tool window from the menu.

You can also use keyboard shortcuts to open tool windows. Table 1 lists the shortcuts for the most common windows.

Style and formatting

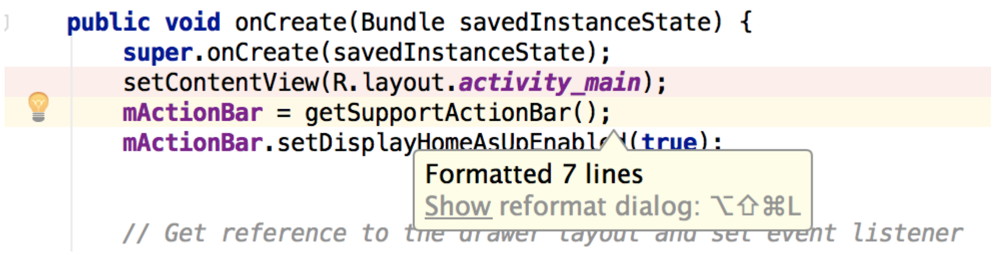
As you edit, Android Studio automatically applies formatting and styles as specified in your code style settings. You can customize the code style settings by programming language, including specifying conventions for tabs and indents, spaces, wrapping and braces, and blank lines. To customize your code style settings, click File > Settings > Editor > Code Style (Android Studio > Preferences > Editor > Code Style on a Mac.)

Although the IDE automatically applies formatting as you work, you can also explicitly call the Reformat Code action by pressing Control+Alt+L (Opt+Command+L on a Mac), or auto-indent all lines by pressing Control+Alt+L (Control+Option+I on a Mac).



```
public void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
        mActionBar.setDisplayHomeAsUpEnabled(true);
```

Figure 4. Code before formatting.



```
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mActionBar = getSupportActionBar();
        mActionBar.setDisplayHomeAsUpEnabled(true);


        // Get reference to the drawer layout and set event listener
```

Formatted 7 lines
Show reformat dialog: ⌥⇧⌘L

Figure 5. Code after formatting.

Gradle build system

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

- Customize, configure, and extend the build process.
- Create multiple APKs for your app, with different features using the same project and modules.
- Reuse code and resources across sourcesets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build.gradle. They are plain text files that use Groovy syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

To learn more about the build system and how to configure, see Configure your build.

Build variants

The build system can help you create different versions of the same application from a single project. This is useful when you have both a free version and a paid version of your app, or if you want to distribute multiple APKs for different device configurations on Google Play.

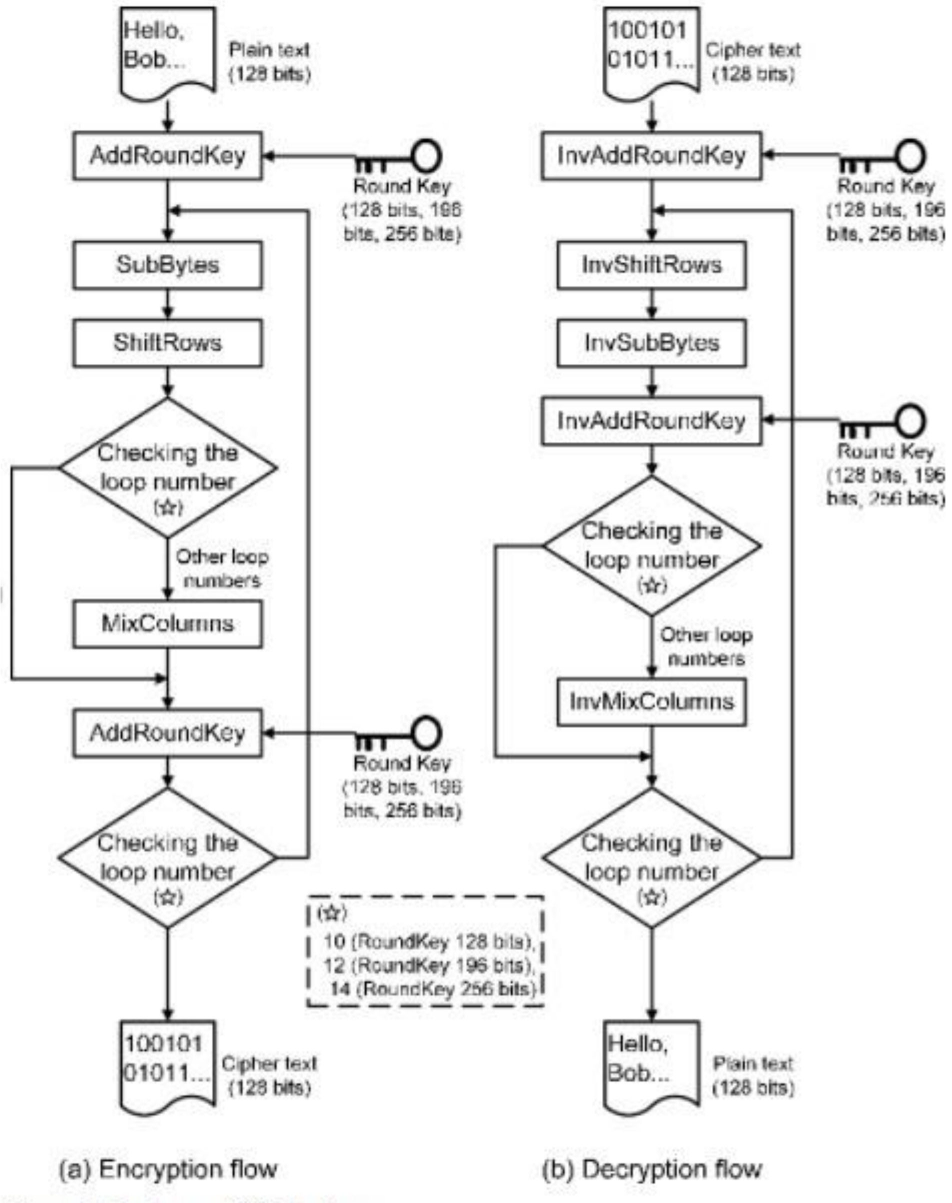
For more information about configuring build variants, see Configure build variants.

Multiple APK support

Multiple APK support allows you to efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and allowing them to share test APK, javac, dx, and ProGuard settings.

Chapter-3

Functionality/Working of the project

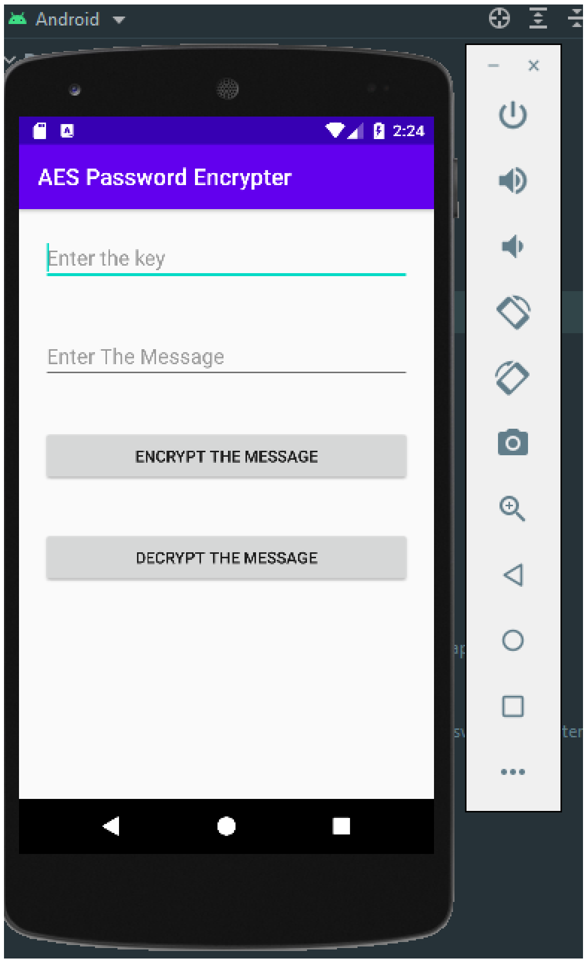


Hello, Bob...
Plain text (128 bits)
AddRoundKey
Round Key (128 bits, 196 bits, 256 bits)
SubBytes
ShiftRows
Checking the loop number (☆)
Other loop numbers
MixColumns
AddRoundKey
Round Key (128 bits, 196 bits, 256 bits)
Checking the loop number (☆)
100101 01011...
Cipher text (128 bits)
(a) Encryption flow
(☆) 10 (RoundKey 128 bits), 12 (RoundKey 196 bits), 14 (RoundKey 256 bits)
100101 01011...
Cipher text (128 bits)
InvAddRoundKey
Round Key (128 bits, 196 bits, 256 bits)
InvShiftRows
InvSubBytes
InvAddRoundKey
Round Key (128 bits, 196 bits, 256 bits)
Checking the loop number (☆)
Other loop numbers
InvMixColumns
Checking the loop number (☆)
Hello, Bob...
Plain text (128 bits)
(b) Decryption flow

Processing flow diagram of AES algorithm.

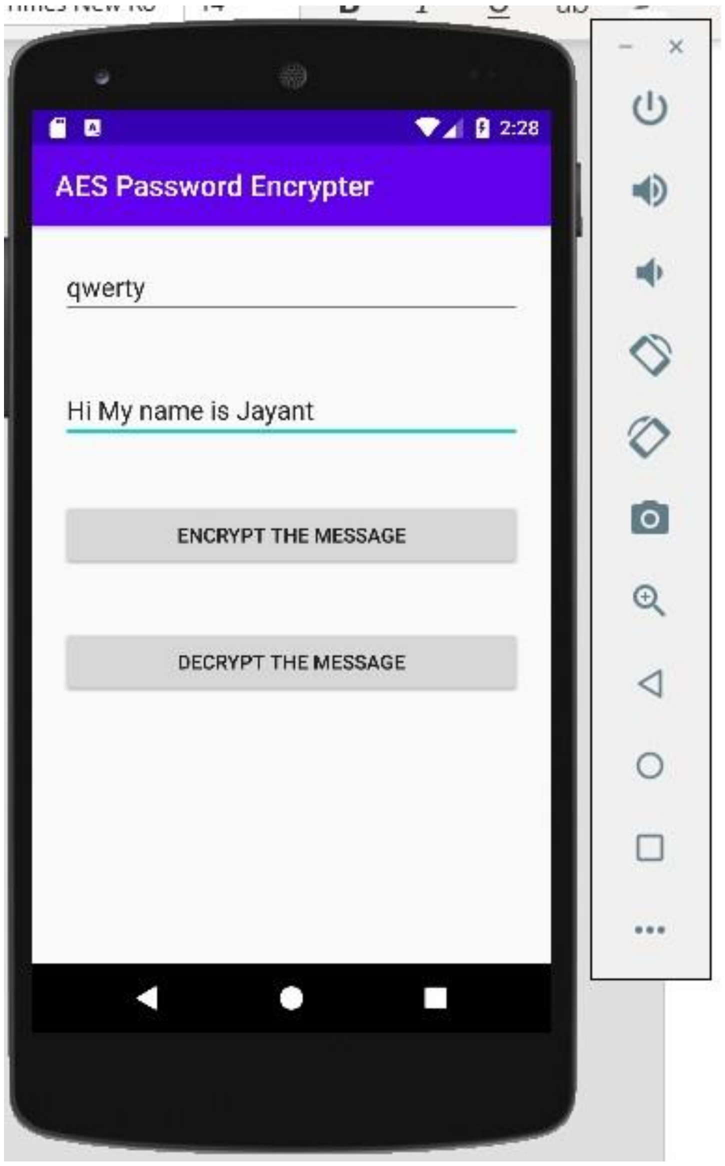
- Data Flow diagram of AES algorithm

1) **Interface:** This is the interface of the application where you can see we have two columns and two buttons. Columns to out message and key and buttons for the functions whether to encrypt or decrypt

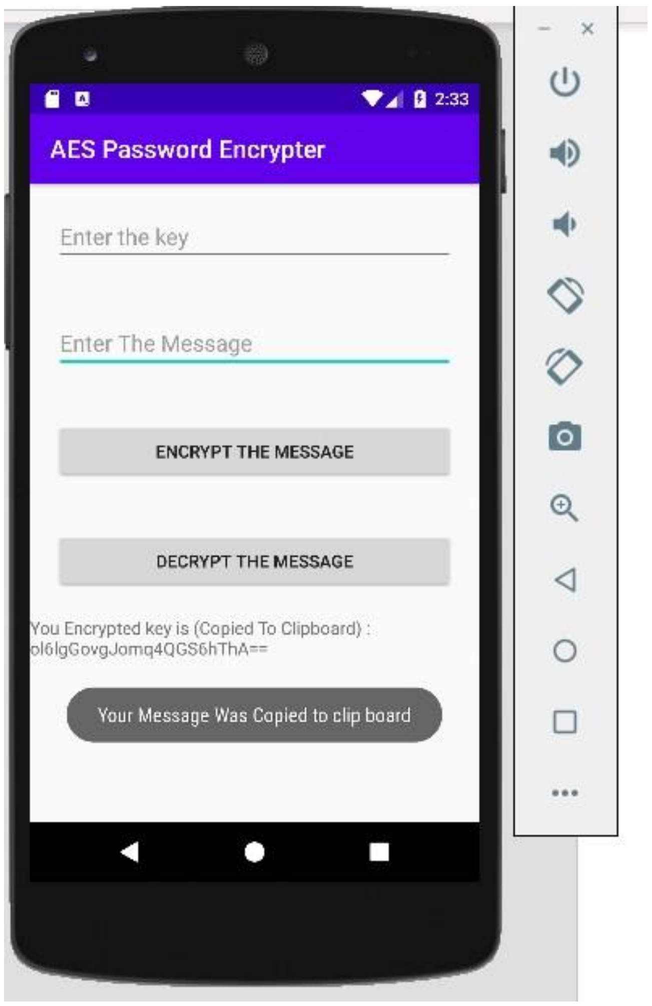


Sender's end

2) **Key and Message:** We then have to set a key for the message and enter the message in the given column.



3) **Encryption**: Now we will encrypt the message and it will be copied to our clipboard and then we can send that through any desired application.

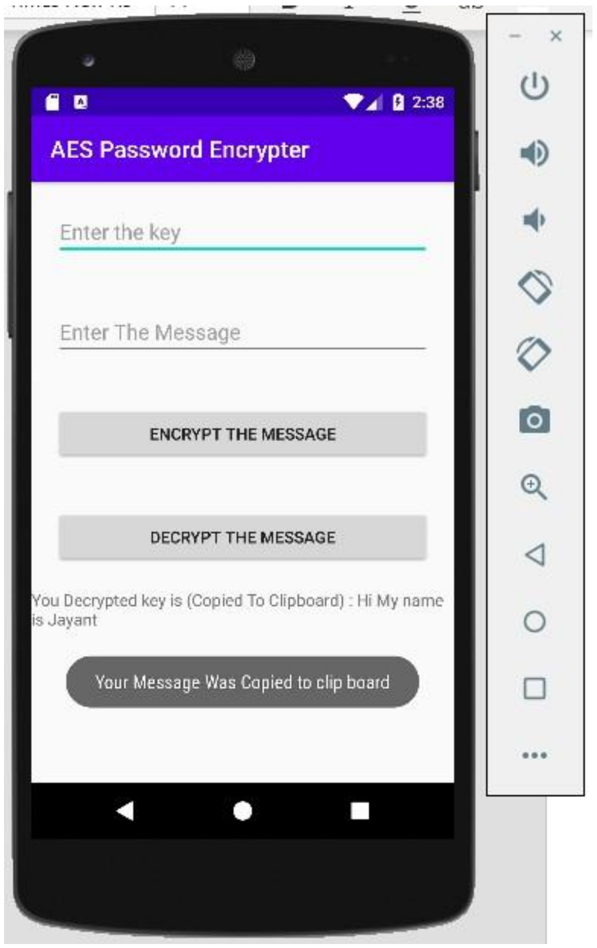


You Encrypted key is (Copied To Clipboard) :
ol6lgGovgJomq4QGS6hThA==

Your Message Was Copied to clip board

Receiver’s end

4) **Decryption:** To decrypt the message, receiver has to put the encrypted text in the message column and should know the key to the message.



You Decrypted key is (Copied To Clipboard) : Hi My name is Jayant

Your Message Was Copied to clip board

- ➢ From the above pictures we have seen the entire working and functionality of the project.

Chapter-4

Results and discussion

We have successfully completed our application program in android studio and with the help of the source code anyone can run this application.

SOURCE CODE(JAVA):

```java
package com.aespasswordencrypter;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.imdigitalashish.aespasswordencrypter.R;
import com.scottyab.aescrypt.AESCrypt;

import java.security.GeneralSecurityException;

public class MainActivity extends AppCompatActivity {

    EditText et_key;
    EditText et_value;
    TextView message;

    @Override
    protected void onCreate(Bundle savedInstanceState)
{
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et_key = findViewById(R.id.et_text_key);
        et_value = findViewById(R.id.et_value);
        message = findViewById(R.id.message);

    }

    public void encrypt(View view) throws
GeneralSecurityException {

        String encrypted =
AESCrypt.encrypt(et_key.getText().toString(),
et_value.getText().toString());
        ClipboardManager clipboardManager =
(ClipboardManager)getSystemService(CLIPBOARD_SERVICE);
        ClipData clipData =
ClipData.newPlainText("label", encrypted);
        clipboardManager.setPrimaryClip(clipData);
        Toast.makeText(this, "Your Message Was Copied
to clip board", Toast.LENGTH_SHORT).show();
        et_value.setText("");
        et_key.setText("");
        message.setText(String.format("You Encrypted
key is (Copied To Clipboard) : %s", encrypted));

    }

    public void decrypt(View view) throws
GeneralSecurityException {
        String encrypted =
AESCrypt.decrypt(et_key.getText().toString(),
et_value.getText().toString());
        ClipboardManager clipboardManager =
(ClipboardManager)getSystemService(CLIPBOARD_SERVICE);
        ClipData clipData =
ClipData.newPlainText("label", encrypted);
        clipboardManager.setPrimaryClip(clipData);
        Toast.makeText(this, "Your Message Was Copied
to clip board", Toast.LENGTH_SHORT).show();
        et_value.setText("");
        et_key.setText("");
        message.setText(String.format("You Decrypted
key is (Copied To Clipboard) : %s", encrypted));


    }
}
```

SOURCE CODE(XML):

```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/andro
id"
    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=".MainActivity">

    <EditText
        android:id="@+id/et_text_key"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:hint="Enter the key"/>

    <TextView
        android:id="@+id/message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/decrypt"
        android:text="" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:layout_below="@id/et_text_key"
        android:id="@+id/et_value"
        android:hint="Enter The Message" />
    <Button
        android:id="@+id/encrypt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Encrypt the message"
        android:onClick="encrypt"
        android:layout_below="@id/et_value"
        android:layout_margin="20dp"/>
    <Button
        android:id="@+id/decrypt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Decrypt the message"
        android:onClick="decrypt"
        android:layout_below="@id/encrypt"
        android:layout_margin="20dp"/>


</RelativeLayout>
```

From the above discussions and documentation we have seen that this app is not for public exploitation but its sole purpose is to keep the privacy in its users and receivers hand.

There will be a lot of questions as If there would be any breach of security the hackers could even hack the mobile phone or other devices and it is very much possible that they can do that in-fact anything is possible in this world nothing is safe, But what we can do is be precautious and take safety measures for this is the only thing that is in our hand to do.

Chapter-5

Conclusion and Future Scope

Conclusion: From our efforts and hard work finally we came to the conclusion of this report as we can't do anything of the government and the rules because they are there to protect us and to some extent we also feel the same but, not everything is meant to be exploited and it is also our right to keep our privacy so we will let the officials do what they want and keep on doing what we have to do with it. We are not doing it in any wrong way. It's just a matter pf privacy and security and we choose to stand for it and have it in our own way by using this application. It might be a little time consuming as you have to encrypt a message then send it then receive and then decrypt it but it could be worth the wait.

Future Scope:

- Currently these applications are only being used by us. But if conceptualized in a more industrial purpose it could be one of those applications which can be bought off by the big security firms which could provide their client the privacy they need.
- Time consuming thing could be reduced in the future time.
- Currently this app is limited to only mobile application so a web application could be made out of this which would be very handy.
- This application could is android limited only so someone with swift language knowledge can convert this into a iOS application too.

Hence there are numerous number of scopes for this project all we need is a good vision and strong shoulder to carry it forward.

References

- **dfd:-** https://www.researchgate.net/figure/Processing-flow-diagram-of-AES-algorithm_fig2_270443179
- https://www.indiatoday.in/magazine/cover-story/story/20201012-how-private-are-your-whatsapp-chats-1727605-2020-10-03
- https://medium.com/asecuritysite-when-bob-met-alice/sometimes-it-feels-like-only-cybercrimals-know-how-to-use-encryption-properly-3dcebcd39ddb
- https://www.thehindu.com/news/national/govt-announces-new-social-media-rules/article33931290.ece
- https://www.geeksforgeeks.org/advanced-encryption-standard-aes/
- https://embeddedsw.net/Cipher_Reference_Home.html- Cryptography – 256 bit Ciphers: Reference source code and submissions to international cryptographic designs contests.
- https://searchmobilecomputing.techtarget.com/definition/Android-Studio#:~:text=Android%20Studio%20is%20the%20official,code%20editing%20and%20developer%20tools.
- https://developer.android.com/studio/intro
- https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm
- https://www.educative.io/edpresso/what-is-the-aes-algorithm
- https://www.youtube.com/watch?v=o-MZgVJP-rI
- https://www.youtube.com/watch?v=o-MZgVJP-rI