Nethaji Chockkalingam

USER PRIVACY AND SECURITY IN MOBILE DEVICES USING BIOMETRICS

Master of Engineering

Degree Programme in Information Technology

2018

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

USER PRIVACY AND SECURITY IN MOBILE DEVICES USING BIOMETRICS

Nethaji chockkalingam

Satakunta University of Applied Sciences

Master of Engineering

Degreee Programme in Information Technology – Jan 2018

Number of pages: 56 Appendices: 0

Keywords: SecureText, Encryption, Fingerprint, Texting App.

Abstract

The purpose of this thesis was to develop an Android app which ensures user privacy and security in mobile devices using Biometrics. This app was designed to perform traditional texting function without use of internet in the android platform. It was intended to use for everyone This thesis mainly discuss the need of storing our mobile data more secure, possible methods and impacts, and why it more important in the modern world, Existing passwords are not secure enough, Encryption techniques, and encryption standard. Fingerprint played the major role for authentication in this CText app.

The development tools used for this project is Android studio, which is most important for design an Android App. And it is recommended and acceptable by all android device manufactures to support maximum function. Java programming language is used for logical and complex coding, AES Encryption standard was used to provide high security and break free the key from brutal-attacks by super computer. It made the CText App encryption is stronger and more reliable and forever secure.

Biometric Fingerprint is used as main verification method for legitimate access in this app. Fingerprint is the most secure biometrics compared to face and Voice recognition and iris scanner. Most of the biometrics required specialized sensors or readers for enrollment. Most of the smartphone comes with fingerprint. So it's easy to use to work with more devices. CText Android came with Fingerprint to ensure user privacy and security.

.

This project is designed and developed using agile software development methodologies, from the planning stage to evaluation stage. Testing performed numerous repeated set of test cases in real android devices to make sure the app functions working without lack or performance degradation. The issues and challenges are faced by the author both in terms of technical issues and other deadlines to meet also presented.

After this CText App launched in Google Play store, feedback will be collected from the end users and potential scenarios and then planned to develop the future development needs to make it more user friendly and better for everyone.

Contents

Abbreviations

SDK –Software develop kit. Tools to develop software programs.

IDE -Integrated Development Environment. Tools to develop software programs.

RAM- Random Access Memory. Runtime memory for applications in computer

APP- Applications. Set of operation defined in a single program for mobile portable devices.

DES – Data Encryption Standard. One type of algorithm used for ciphering/encoding.

AES – Advanced Encryption Standard. Latest algorithm used for ciphering/encoding

XML- Extensible Markup Language- are used both as a form of data communication between systems

SMS – Short Message Service. Used in mobile as communication service.

PIN -Personal Identification Number. Kind of password to protect the system/mobile device.

1 Introduction

Modern world, Mobile phone become vital and has become a part of our body; having no access to a phone is unimaginable for everyone nowadays. Despite the fact phone is discovered for communication. It evolved into next stages now its smart phones. Smart phones come with plenty of features and services. We use phone for stores personal information and memories in the form of text, images, and videos. We can say phone is our digital diary. Millions of mobile applications are available in market to enhance more features in the phone. One of the most important features is texting. We share our personal data and thoughts to other person through chat apps (Texting, WhatsApp, WeChat, and so on). Texting is more convenient and elegant than talking to someone new. If anyone accessed texts in your phone, your personal life will be revealed. There is no modern security to stop this. Once your phone is unlocked, it's like an open diary.

In this Thesis, am going to discuss, how can we improve our personal data is more secure way to store and communicate using an Android App **CText.** Am presenting an app which lets you do texting in a high secured fashion and presenting data always encrypted in phone as well as over air communication. It ensures only you can access and read your personal information.

1.1 Texting

Texting is inevitable in modern world. Important facts about texting are

- Texting is the most widely-used and frequently used app on a Smartphone, with 97% of Americans using it at least once a day.
- People worldwide will send 8.3 trillion text messages in just this year alone. That's almost 23 billion messages per day, or almost 16 million messages per minute.
- Over 80% of American adult's text, making it the most common cell phone activity. Only 43% of Smartphone owners use their phone to make calls, but over 70% of Smartphone users text

- 55% of heavy text message users (50+ texts per day) say they would prefer to receive a text over a phone call
- It takes the average person 90 minutes to respond to email, but only 90 seconds to respond to a text message. (Tumbleson, 2016)

Is texting safe and secure? Does it provide privacy? Most likely, the answer is No. Even our latest smart phones are failing to provide more security. Whoever can access the phone, can read your text. And eventually will come to know your private life.

How this CText App differ from existing million texting app.

- ✓ This app ciphers your message to unreadable format makes it hard to decrypt.
- ✓ It uses strong 128bit AES Encryption
- ✓ Decryption happens after the biometrics (Finger print) authentication. After that it shows the original contents

1.2 Encryption

Cipher (or Cypher) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure to encrypt the original message, make it more secure and provides confidentiality. It is sent over medium, and the receiving side will perform the sequential set of processes to decrypt the encoded text which makes it read the original text. We have used aspects of history to help us invent this algorithm.

Caesar Cipher History. The Roman ruler Julius Caesar (100 B.C. – 44 B.C.) used a very simple cipher for secret communication. He substituted each letter of the alphabet with a letter three positions further along. Today, we call it "cryptography", nowadays we have strong Encryption method DES (Data Encryption standard) and AES (Advanced Encryption Standard). AES more is secure than DES. This app uses AES 128-bit encryption. (National Institute of Standards and Technology, 2017)

In this Project, fingerprint is going to be used as a primary authentication method to prove legitimate users. This project is going to implement biometrics security, so both Biometrics must satisfy to decrypt or see the original contents.

1.3 Fingerprint security

We have fingerprints; the tiny **friction ridges** on the ends of our fingers and thumbs make it easier to grip things. You have fingerprints even before you're born. In fact, fingerprints are completely formed by the time you're seven months old in the womb. Unless you have accidents with your hands, your fingerprints remain the same throughout your life.

What makes fingerprints a brilliant way of telling people a part is that they are virtually unique. Fingerprints develop through an essentially random process according to the code in your DNA. Because the environment in the womb also has an effect, even the prints of identical twins are slightly different. While it's *possible* that two people could be found who had identical fingerprints, the chances of this happening are so small as to be virtually negligible.

1.4 Android

This app is developed in Android platform to support the major phone vendors. Am going to discuss all the security features and technologies in detail. Design and workflow .and how and why CText is unique and important to have in your phone.

2 Framework

2.1. Android

Android is a mobile operating system developed by google using an Linux kernel environment for smartphone and touch enabled devices. And it is also open source project under non-copyleft Apache License version 2.0. Which allows modification and redistribution. According to Wikipedia, android is the largest OS base with 2 billion active devices as of March 2017 and It has more than 4 million apps grows everyday.it has different versions Gingerbread 2.3, Ice cream, Jelly bean and latest one is Oreo released on Dec 5, 2017, google keep on updating and improving the existing environment better and user friendly.

2.1 Android Architecture design

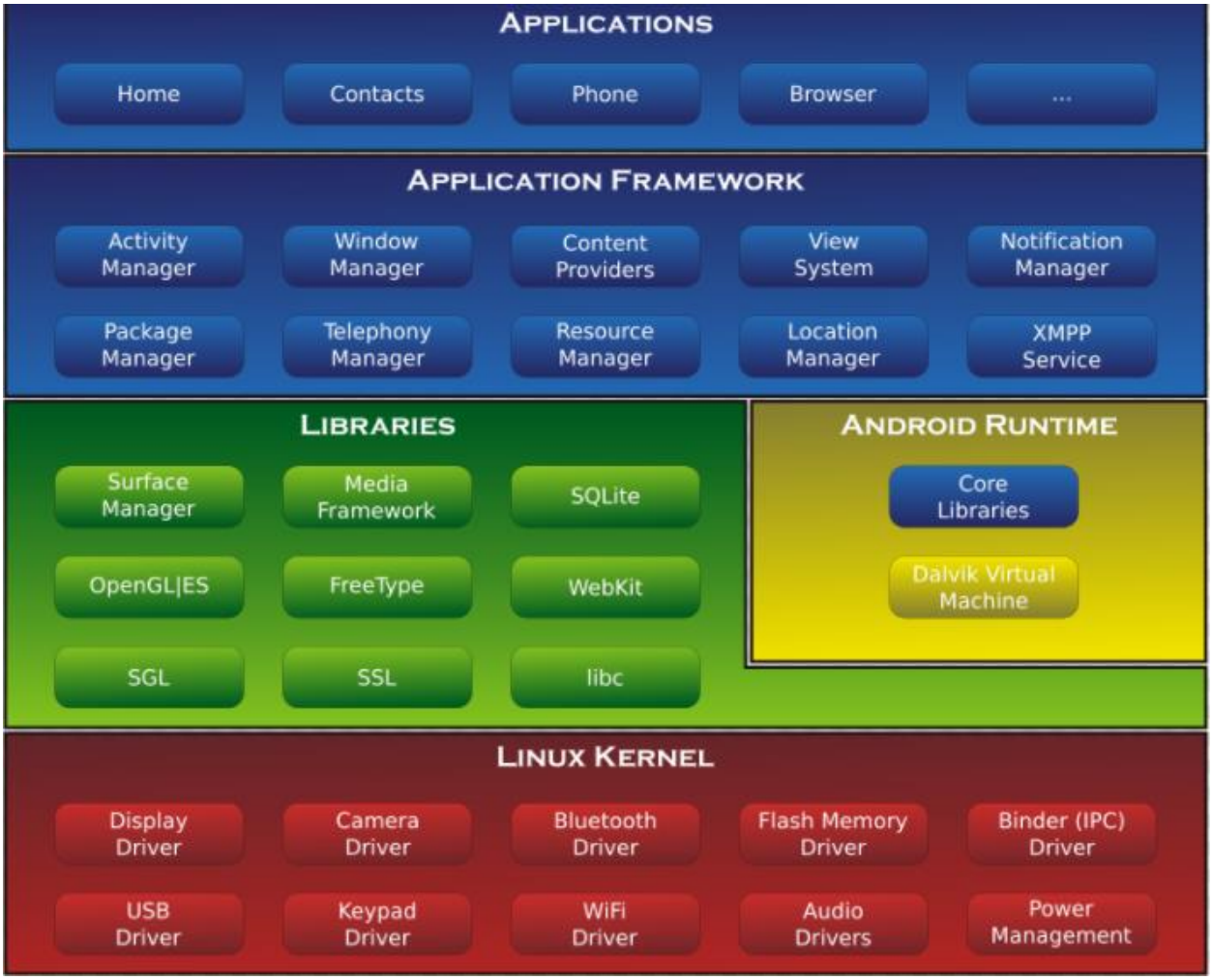


Figure 2. 1 Android architecture. (Anon., 2017)

To develop android App we need Android studio, we can develop app using java programming language and with and without the support of C++. Latest google announced android will support Kotlin programming language as well in future,

2.2 Android Studio

Android studio is the official software development kit (SDK) or IDE (integrated development environment for android operating system specially designed for mobile devices by inteliJ IDEA software.it provides the specialized features for mobile app development. This software is heavy weight, so the system requirements must be high configuration at least 8GB RAM to run smooth and faster. It lets you emulate virtual devices and do sample testing and verify before deploying into actual device. Each app has development involves three different major functions Design, programing code and Build.

2.2.1 Design

Design requires set of visual components to build an app such as Buttons, TextBox, ListView, ImageView and much more. And you can write or design a new shape using XML Layout file and XML resource file.

Layout: Layout provides the visual structure of user interface activity, five types of layout available in android are Linear, Relative, WebView, GridView and ListView.

```
RelativeLayout
<?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:background="@android:color/darker_gray"
    tools:context=".MainActivity">
```

Each component has respective properties to design and manipulate better as per our needs. Below shown the design view of Android studio.

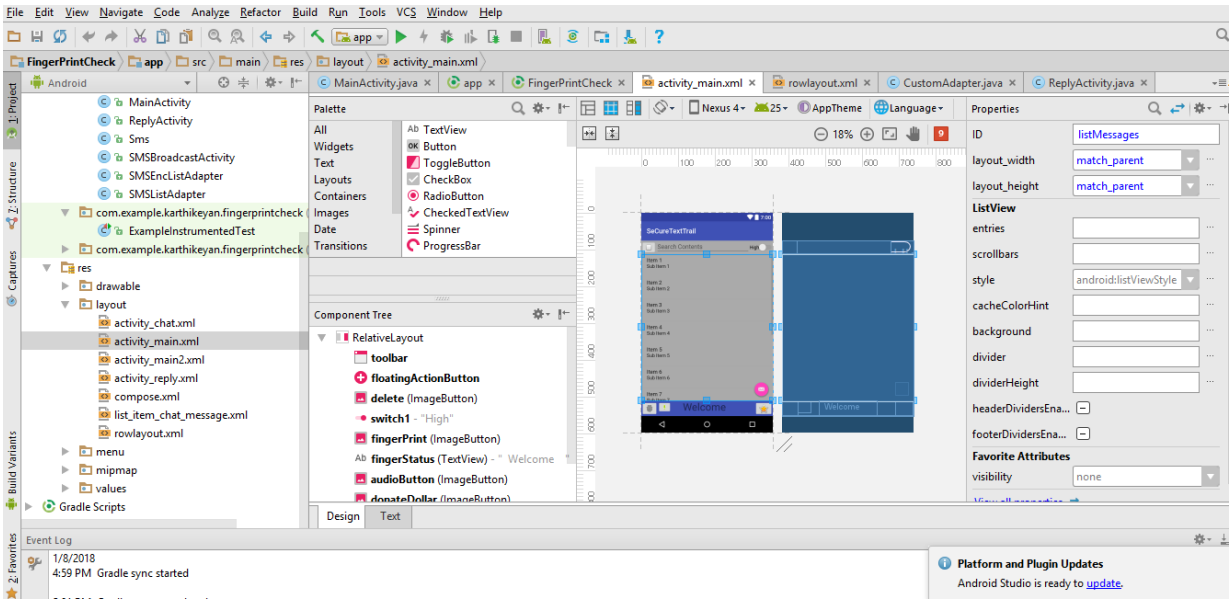


Figure 2. 2 Android Studio SDK

2.2.2 Coding part

CText app is written in java programming language, and few android library extensions. In general Android app is supports Java, and C++ for native libraries.

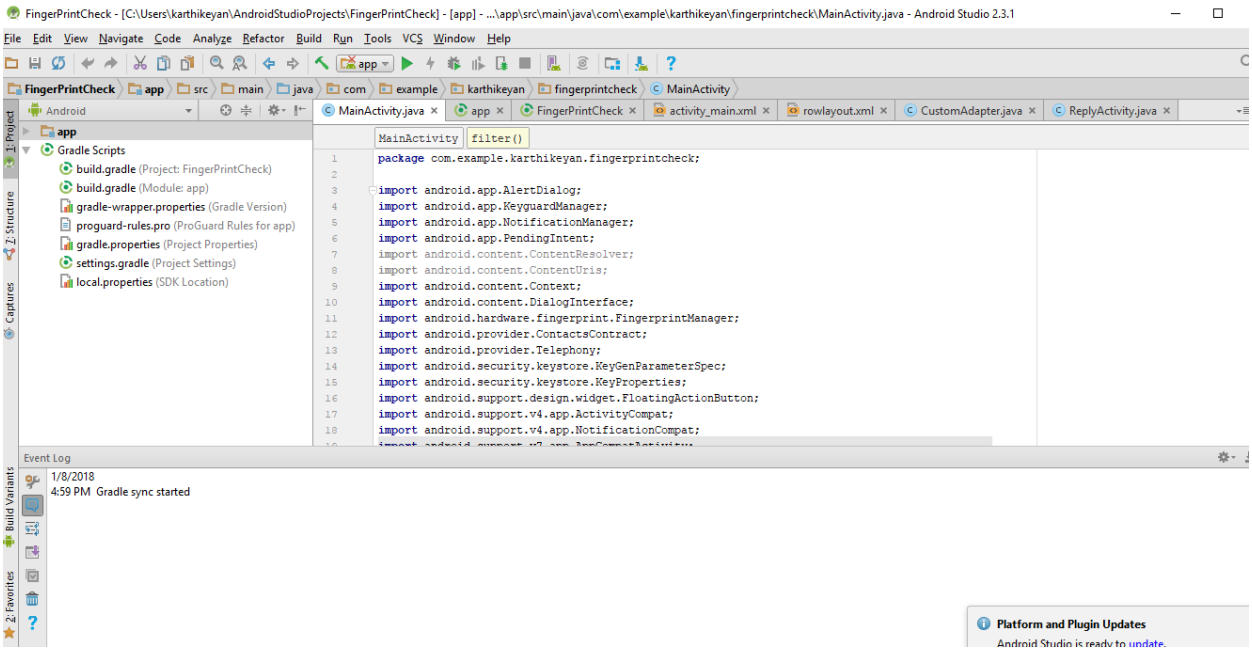


Figure 2. 3 Android studio. Main Activity. Java class

Android Intents.

Intents are the objects of android, it is used to call another activity or action from the current activity .in simple, you are logging in home page, once you click login, one intent called and displays the main contents of another activity.

Android Views

All the UI elements are created in View which the user interact to perform actions, viewGroup is the collection of views, this object is used to define a layout of user interface.

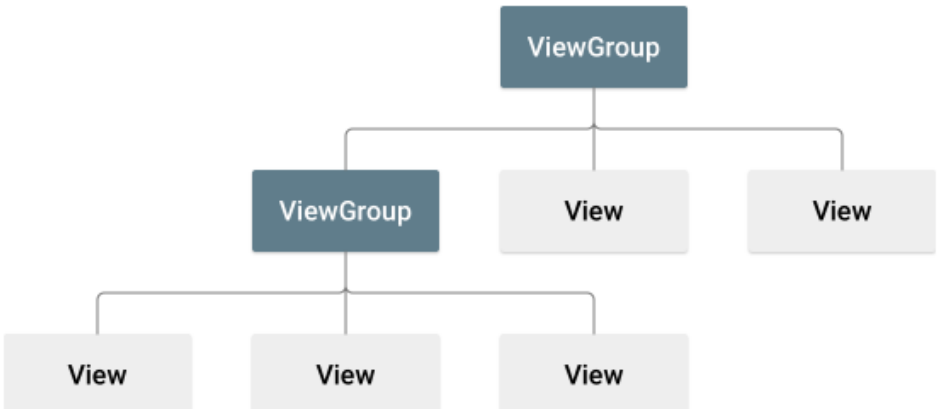


Figure 2. 4 Android View structure and models

Android Manifests

It is one of the most important file in app creation, App will run only if this file exists in the main directory in the same name convention AndroidManifest.xml. It defines the important properties of the app required to run such as the permission requirements and activity names and launcher activity declarations. App icon and labels and themes and much more. All are defined in XML structure as follows.

Android manifest Structure

```
<?xml version="1.0" encoding="utf-8"?>

<manifest>
```

```
<uses-permission />
<permission />
<permission-tree />
<permission-group />
<instrumentation />
<uses-sdk />
<uses-configuration />
<uses-feature />
<supports-screens />
<compatible-screens />
<supports-gl-texture />

<application>
   <activity>
      <intent-filter>
         <action />
         <category />
         <data />
      </intent-filter>
      <meta-data />
   </activity>
  <service>
      <intent-filter> . . . </intent-filter>
      <meta-data/>
   </service>


   <receiver>
      <intent-filter> . . . </intent-filter>
      <meta-data />
   </receiver>
```

```
        <provider>
            <grant-uri-permission />
            <meta-data />
            <path-permission />
        </provider>
        <uses-library />
    </application>
</manifest>
```

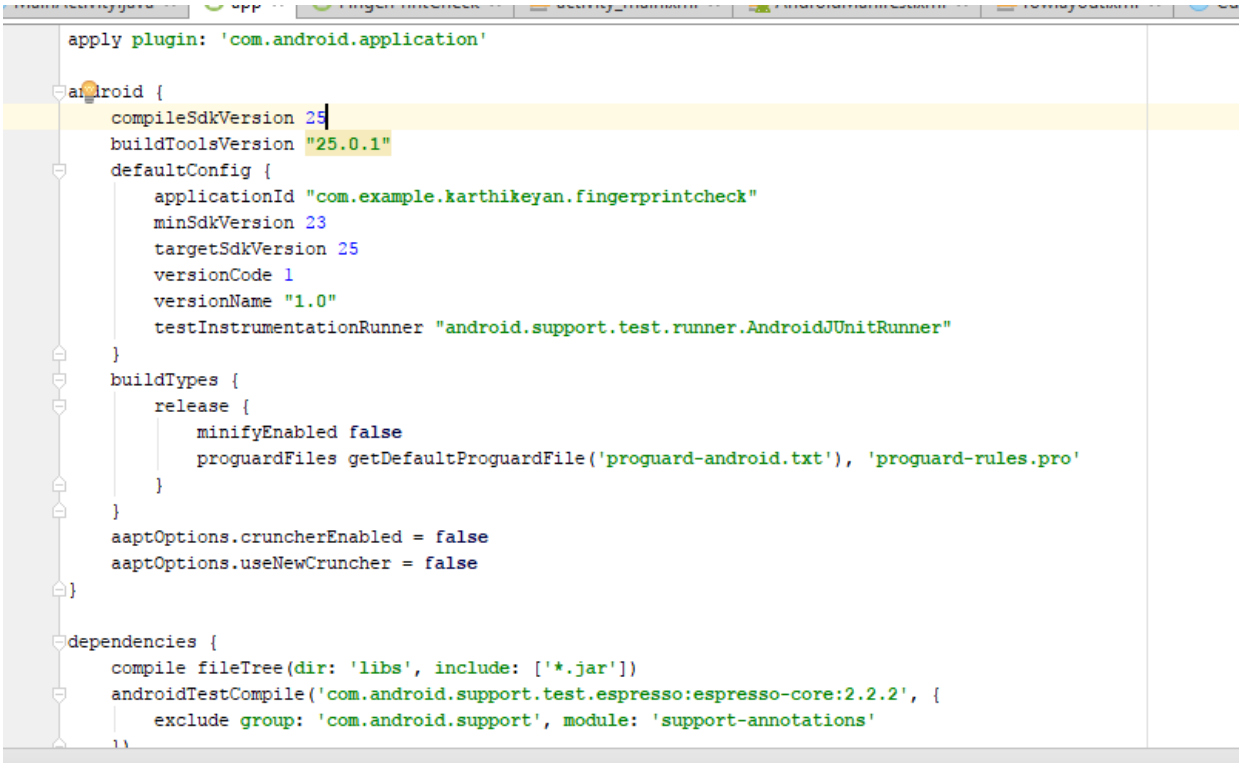
```
manifest | application
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/textapp"
        android:label="SeCureTextTrail"
        android:roundIcon="@drawable/textapp"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Main2Activity" />
        <activity android:name=".ReplyActivity" />
        <activity android:name=".ChatActivity"></activity>
    </application>
```

2.2.3 Build an App

Android studio uses gradle to build an android app, it's the final stage of app development. It ensures to choose the right sdk version and the required resources and then compile and give the output as Build Successful. And then we have two options for us to run application in virtual devices using an emulator or generate an .apk file to install in our phone directly.



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.1"
    defaultConfig {
        applicationId "com.example.karthikeyan.fingerprintcheck"
        minSdkVersion 23
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
    aaptOptions.cruncherEnabled = false
    aaptOptions.useNewCruncher = false
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
```

Cryptography

It's the art of storing and transmitting a data in a secure way to the intended recipient only can read and process using an Encryption (process of converting actual text to unreadable cipher text using a complex mathematical operations) and the receiver perform the decryption (processing the cipher data to the actual text using the same set of operation).

2.3 Encryption Standards

2.3.1 Data Encryption Standard

Its symmetric **key block cipher** that was developed by **National Institute of Standard and Technology** in **1977**. DES follows **Feistel structure** where the plaintext is divided into two halves. DES takes input as 64-bit plain text and 56-bit key to produce 64-bit Cipher text.

- DES was published in 1977 .it uses symmetric block cipher using 56-bit key used to store the confidential digital information it was used in US until 1996, later it was proved that ineffective, using brute-force they can crack the original content.

- Two organization distributed.net and Electronic frontier foundation are played a major role in cracking the DES. The DES 1 contest took 84 days to use a brute force attack to break the encrypted message. Later DES III challenge issued in 1999, and they break a code in just 22 hours and 15 mins.
- Triple DES: it uses the DES process three times to make it stronger, still it is breakable by Brute-force method. And it causes the system to slow down due to repeated encryption and decryption in triple DES process. (Larson, 2104)

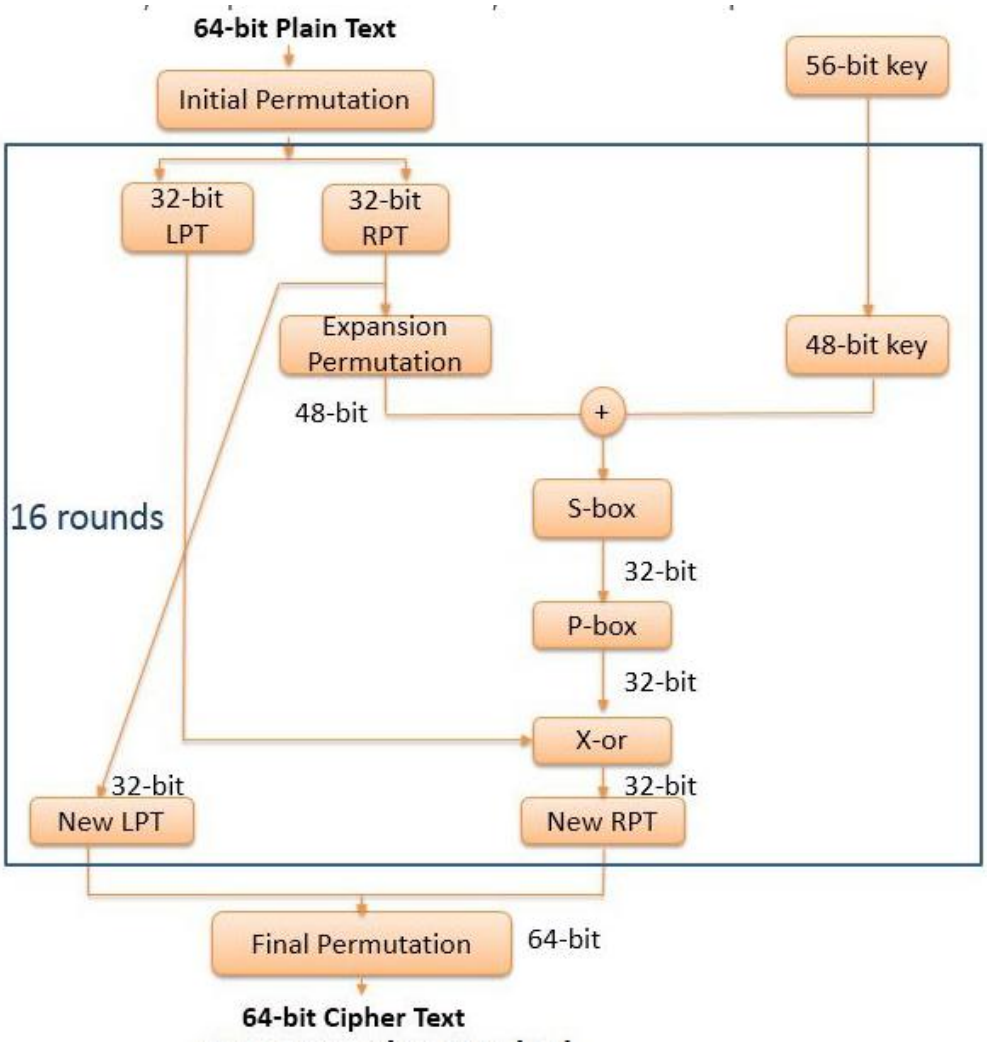


Figure 2. 5 DES Encryption Process (Anon., October 2016)

2.3.2 Advanced Encryption Standard

AES Advanced Encryption standard was developed in 2001, unlike DES, it has option to choose different key length like 128, 194 ,256 bit., it performs the functions subbyte, shiftrows, mixcolumns and addRoundkey to encrypt the message.so far it is known as strongest Encryption algorithm. So CText app designed to implement AES with 128 bits key

It provides more security in the user messages. Even with a supercomputer, it would take **1 billion years** to crack the 128-bit AES key using brute force attack. This is more than the age of the universe (**13.75 billion years**).

AES flow:

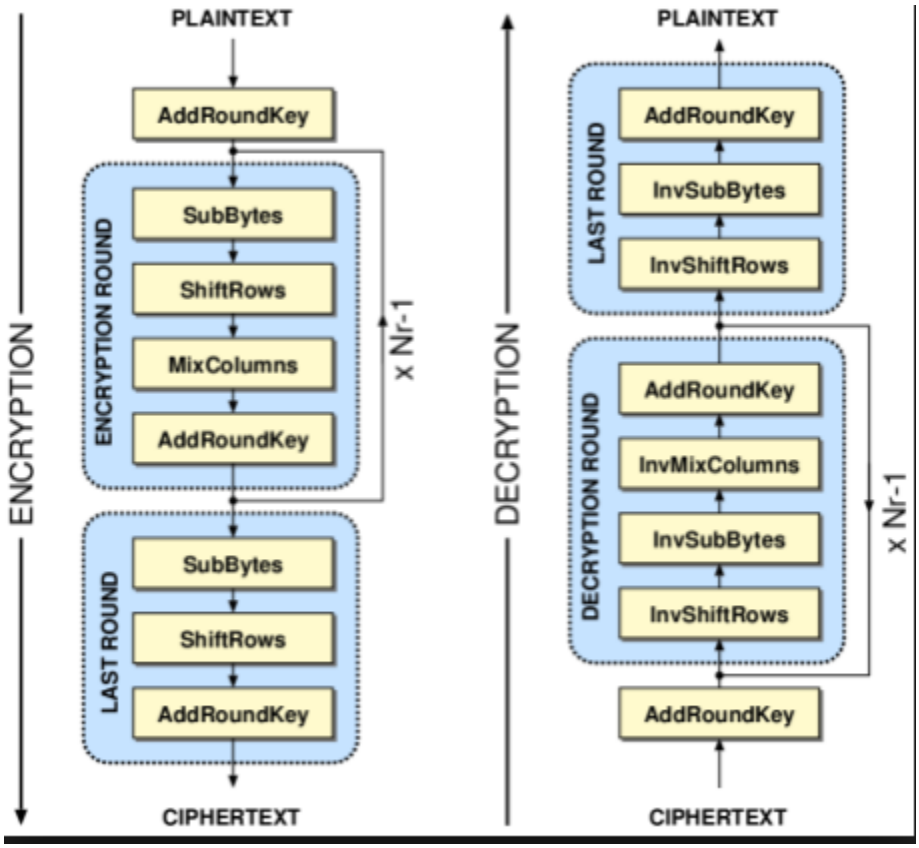


Figure 2. 6 AES Encryption Process (Leith, n.d.)

SubBytes

This is simple substitution process, which converts each byte into different values using AES defined table for 256 substitution value arrays, which can be restored using an inverse substitution table. The contents of the table are computed using mathematical formula. Substitution table will be stored in memory.

ShiftRows

Name suggest that shifting rows in a matrix data format by n bytes. Each row is rotated to the right by a certain number of bytes.

MixColumns

Each column in an array processed separately and generate a new column, and new data replaces the old column data. It involves matrix multiplication.

RoundKey

It is very simple. Perform XOR Operation with the existing array, XORs the value of the appropriate round key, and replaces data with the result. It is done once before the rounds start and then once per round, using each of the round keys in turn

Table 2. 1 Why AES is over DES (Anon., October 2016)

BASIS FOR COMPARISON	DES (DATA ENCRYPTION STANDARD)	AES (ADVANCED ENCRYPTION STANDARD)
Basic	In DES the data block is divided into two halves.	In AES the entire data block is processed as a single matrix.

BASIS FOR COMPARISON	DES (DATA ENCRYPTION STANDARD)	AES (ADVANCED ENCRYPTION STANDARD)
Principle	DES work on Feistel Cipher structure.	AES works on Substitution and Permutation Principle.
Key size	DES in comparison to AES has smaller key size.	AES has larger key size as compared to DES.
Rounds	16 rounds	10 rounds for 128-bit algo 12 rounds for 192-bit algo 14 rounds for 256-bit algo
Rounds Names	Expansion Permutation, Xor, S-box, P-box, Xor and Swap.	Subbytes, Shiftrows, Mix columns, Addroundkeys.
Security	DES has a smaller key which is less secure.	AES has large secret key comparatively hence, more secure.

BASIS FOR COMPARISON	DES (DATA ENCRYPTION STANDARD)	AES (ADVANCED ENCRYPTION STANDARD)
Speed	DES is comparatively slower.	AES is faster.

2.4 Biometrics Authentication

It is widely used in the field of computer science for access control using the biometrics. It's nothing but the body characteristics and measurements into matrix and match with unique every person, prominent biometrics are fingerprint, retina (Iris), voice recognition, facial recognition. It is more secure than traditional password and PIN (personal identification Number). Since biometric identifiers are unique.

Most of the biometrics are requires specialized sensors or devices to register and verify the legitimate access. Normal biometric validation follows this.

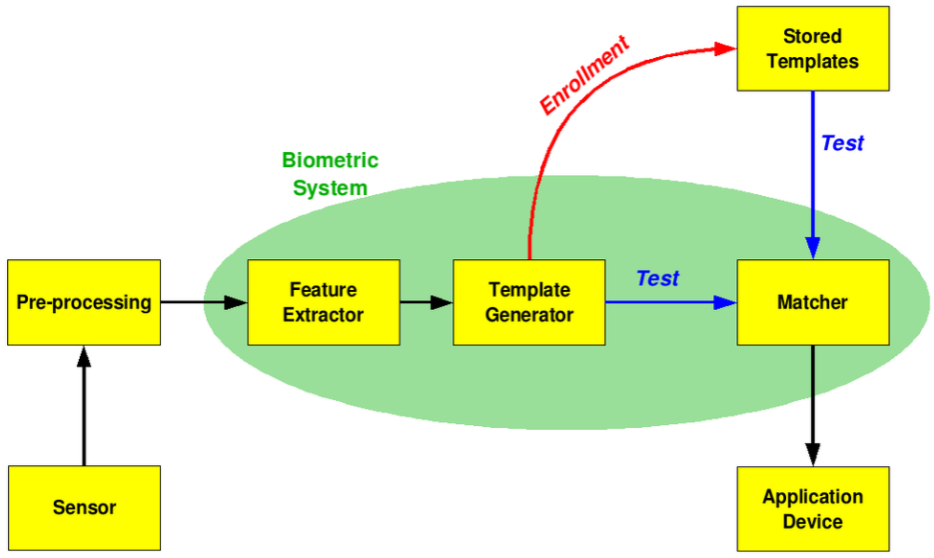


Figure 2. 7 simple biometric process model (Anon., 2017)

Like all other technologies, it has also some flaws. That is false match rate (FMR).

- **False Accept Rate-** the system accepts the false user as legitimate user due to the threshold match of the original data stored in the database, then system treats as genuine, it purely depends on threshold value set by the system.
- **False non-match Rate** – systems reject the legitimate user due to fail to detect the match between input pattern and template in the database. It depends on the percent of the valid input read or high threshold limit set by the system.

2.4.1 Fingerprint

Impression left by the friction ridges of a human finger, it is widely used to identify a person by unique. This fingerprint match identification is used for centuries. First usage recorded in 200 BC in china records from the Qin Dynasty (221-206 BC) include details about using handprints as evidence during burglary investigations. Nowadays all the police stations are using finger print to identify criminals, in the field of forensic science fingerprint plays major role in crime scene investigation. USA holds the fingerprint database for 50 million visitors to the US. And the largest biometric fingerprint database is hold by Aadhar (/wiki/Aadhaar), 2018)Indian government unique identification system which enables to search rapidly using an automated system for comparison of 1.1 billion records. No two fingers are alike even a million humans fingerprint taken, compared by automatic computer or human. Even twins have different pattern in fingerprints, so fingerprint biometric is very unique.

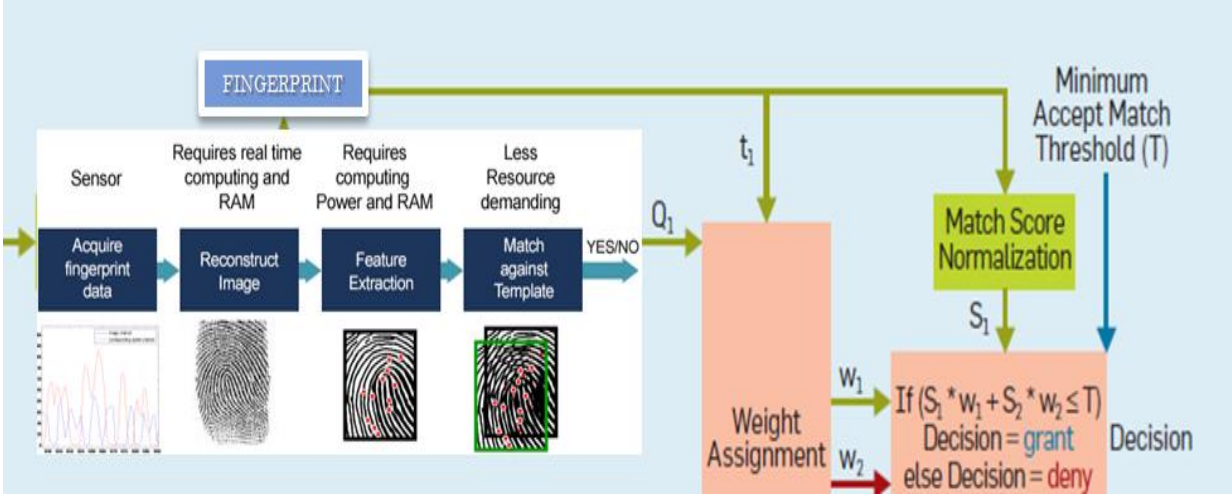


Figure 2. 8 Fingerprint access work model

Fingerprint is taken using scanners, **Optical Scanner**. Works by shining a bright light over your fingerprint and take a digital photo. Digital image feed into a computer scanner. The scanner uses a light-sensitive microchip to produce a digital image. The computer analyzes the image automatically, selecting just the fingerprint, and then uses sophisticated pattern-matching software to turn it into a code.

Unlike capacitive scanner, measures your finger electrically. When your finger rests on a surface, the ridges in your fingerprints touch the surface while the hollows between the ridges stand slightly clear of it. In other words, there are varying distances between each part of your finger and the surface below. A capacitive scanner builds up a picture of your fingerprint by measuring these distances. This type of scanners is used in iPhones and iPads and other touch enabled devices.

2.4.2 Face Recognition

A **facial recognition system** is used for secure systems to identify the person from the image or video frame in live monitoring crowd where crowd people gathering and requires without subject cooperation. Like in airport or big commuter stations. Key advantage of this face recognition is biometrics like fingerprints, iris scans, and speech recognition cannot perform this kind of mass identification. Latest face-id is coming on iPhoneX, it allows user to unlock their phone using Face. Every face has numerous, distinguishable landmarks, the different peaks and valleys that make up facial features.

Each human face has approximately 80 nodal points. Some of these measured by the Facial Recognition Technology are:

- Distance between the eyes
- Width of the nose
- Depth of the eye sockets
- The shape of the cheekbones
- The length of the jaw line

These nodal points are measured creating a numerical code, called a faceprint, representing the face in the database. This Face recognition is vulnerable to twins and it's often referred as inefficient compared to other biometrics.

3 Research Methodology

The Research method of this study closely related to agile software development method. Agile manifesto introduced into 2001. Later it goes through more development and widely used in most of the software development firm, due to anticipated changes flexibility for modification than other methods. In this thesis, agile software methodology (Scrum) is used to support new random ideas whenever happens and flexible to modify and it support from planning to Evaluation stage.

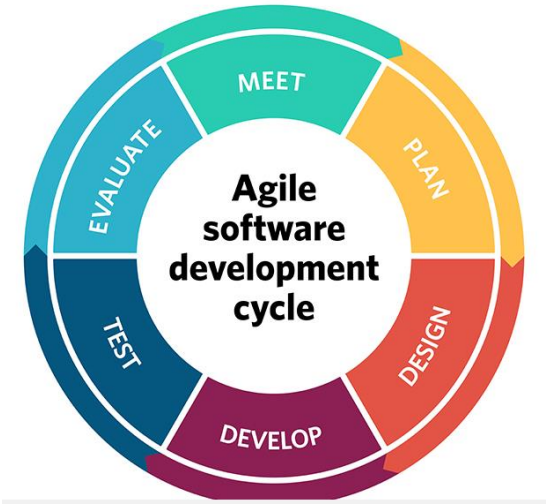


Figure 3 Agile software methodology (Rouse, n.d.)

3.1 Planning

This is the first stage of software development cycle. Planning section we are going to discuss is what we going to do, for whom we are going to do, why we are doing this and how we are doing this. We plan to develop a Mobile App in Android platform initially to provide user privacy in chat communication, in the modern times, everyone prefers to text than long boring voice conversation in call, it gives them enough time to think before they respond back. It avoids lot of misunderstanding especially if in the start of relationship or communication with new people.so Texting an app motive, so what's so special about this, already in the market thousands of Texting/chat apps there. This CText App provides more privacy than any other app which is now

in the market. It's just not the end to end encryption service like all other chat apps. This app provides all time encryption. It assures you only read your phone contents, not anyone else. CText uses strong Encryption algorithm AES to secure your data with 128/256 key length. The beauty is you don't know about anything key and some complex technical term. Anyone can use this. Am creating an app for all, who is normal people who wants more privacy in their phone. Almost everyone. Why am doing this app. everyone must keep their secrets safer, even with their partners. If someone can go through your phone contents especially text conversation, your personal life will be exposed.

How am doing this., CText is a simple app it will work without internet now, in future work it will be enabled with internet, it uses the regular phone service to send SMS, half of the world now, texting is free. After Installing the App, this app replaces the default SMS app. No sign in required. It encrypts all the existing messages to encrypted unreadable format. To read the original contents of the message you need to authenticate with your fingerprint. It doesn't use Pin or password concept, because both are easily guessable and eavesdrop by others. When you want to send a text message, it will have option to send regular message or encrypted message. If you send an encrypted a message, the Receiver must have this app to read the contents, this message will not decrypt by any other forms without CText app. It lets you send the multimedia content. This app will not preview the contents, but it does notify the user.

CText provides 100 percent privacy in all aspects than any other existing app. No middleman can't intercept, even if they did, they cannot decode without user key, key is not stored anywhere other than the respective user phone. CText is designed to be more secure so some Feature is disabled. CText doesn't have the feature to forward the text Message, and does not let you copy the contents of the text. It will not allow you to do perform backup either. Remaining all other features required for performing normal texting is available, you can reply, delete, select contacts, attach picture from gallery and take photos directly from the Camera. Based on user support, will come additional features in future.

3.2 Design

It is the second stage of software development lifecycle of Agile Methodology. Design part is very important, since it must attract the user even before start using the app. To initiate a design process, first set up an Android Studio development Environment.

1. Install the Android Studio is a heavy weight software development kit, system must be high configuration to smooth running application.
2. Java SDK will be downloaded and linked to android platform automatically
3. Android studio is about 3.5 to 4 Gigabytes, it requires some additional plugins based on project model and support.
4. To run emulator and virtual devices, it needs to download some resources files.

Creating a first project is a time taking process, but it is simple few steps needs to follow, one good thing is android studio does most of the things automatically.

Select File → New project →Name →Select Empty Activity → Activity Name → Finish.

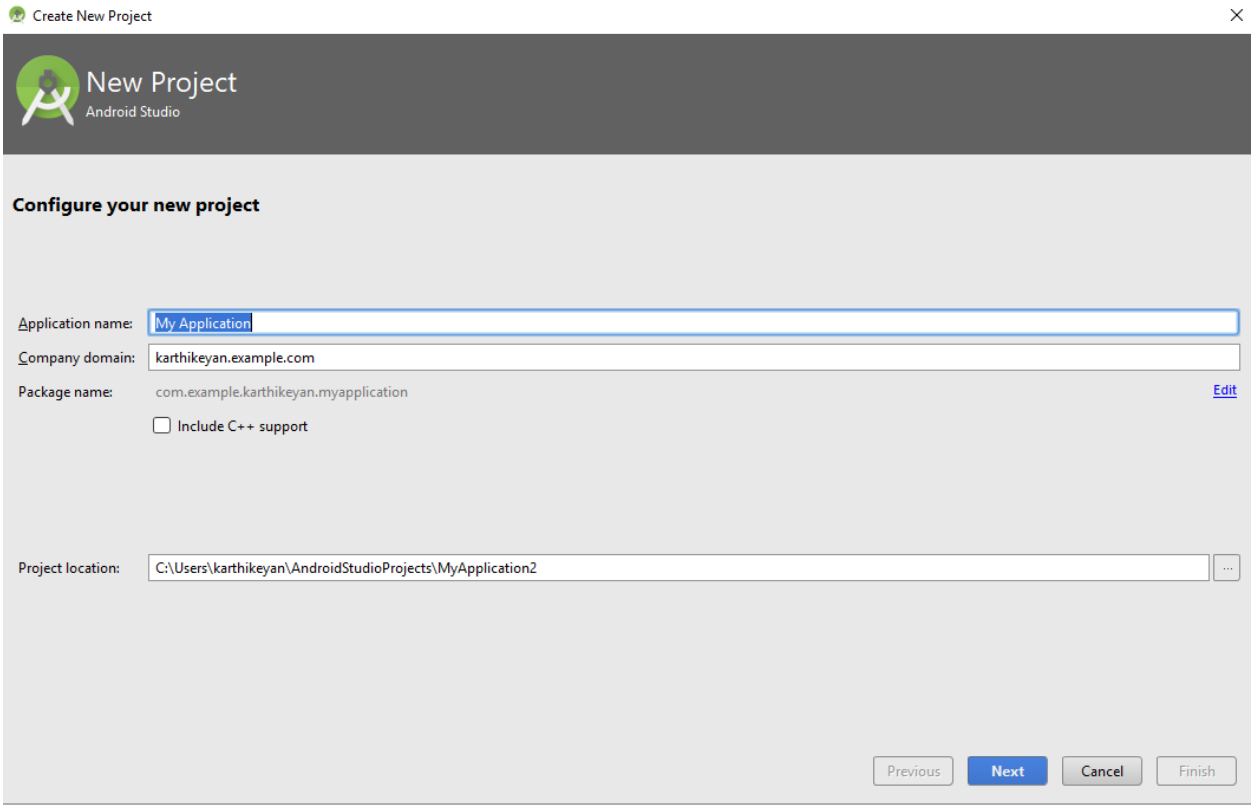


Figure 3. 1 New project creation

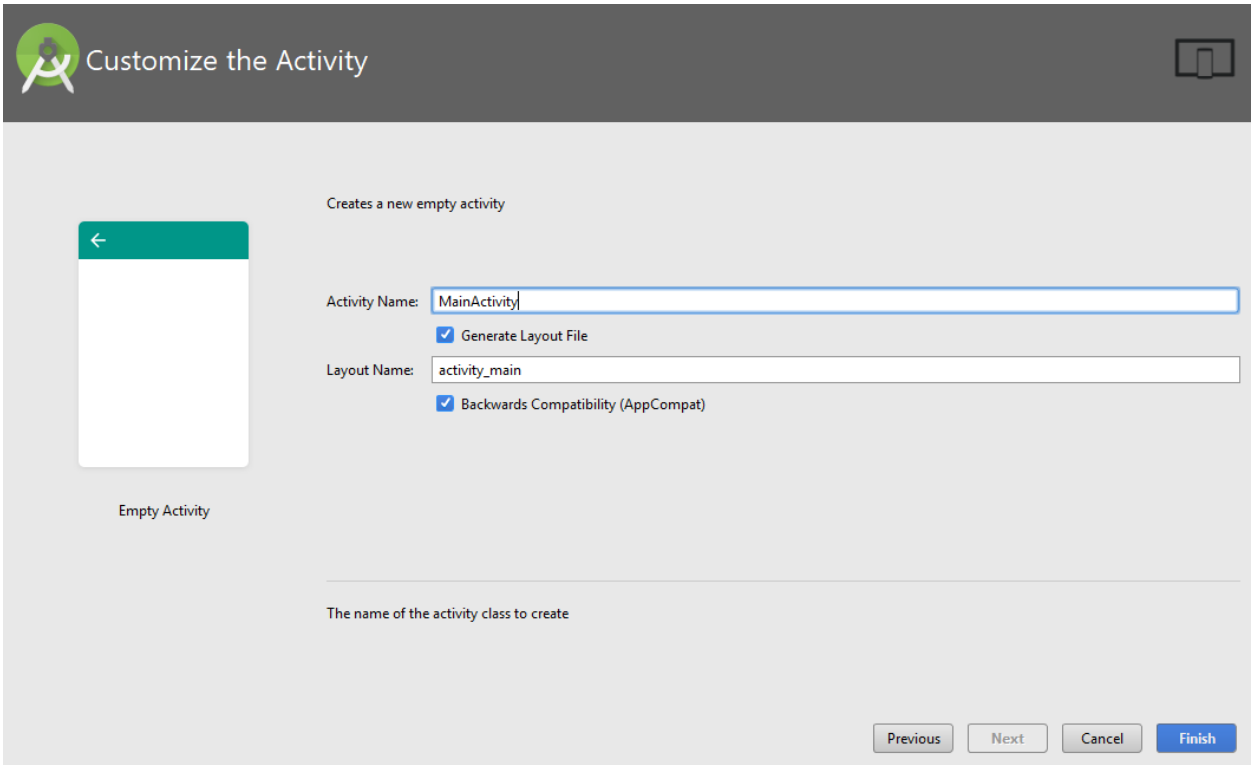


Figure 3. 2 defining master activity name for the created project

Android Studio opens with new project Name and developing and design options. Layout file is an XML format, where you must design the app. either by dragging the elements or writing an XML code.

Android Buttons

It's is the most common elements used to perform user action when the button is Clicked. Over the button, you can name it for user easy understanding. And there is an attribute On-Click " android: onClick="OnSendClick". This attribute helps you define a method OnsendClick, inside the method, you can write a code for action you want to perform with this button.

Android ImageButtons

It is same as the Buttons properties, in addition to that top of the button, you can lay image instead of text for elegant presentation of your app design.

Android Text View.

TextView helps you get user input from the app, later we can process using a coding. MultiText let you allow get more user input more than one line.

Android List View

It is very useful when you want to arrange the app contents (any data) in a List format, CText uses text messages in List view.

Android Floating Button

It's the special type of button not attached to any location, you can drag it around the app and still it performs the actions.

Android Switch

From the name, we understand that either it can be ON or OFF. It lets the user to choose the default sending method as encrypt message or regular message.

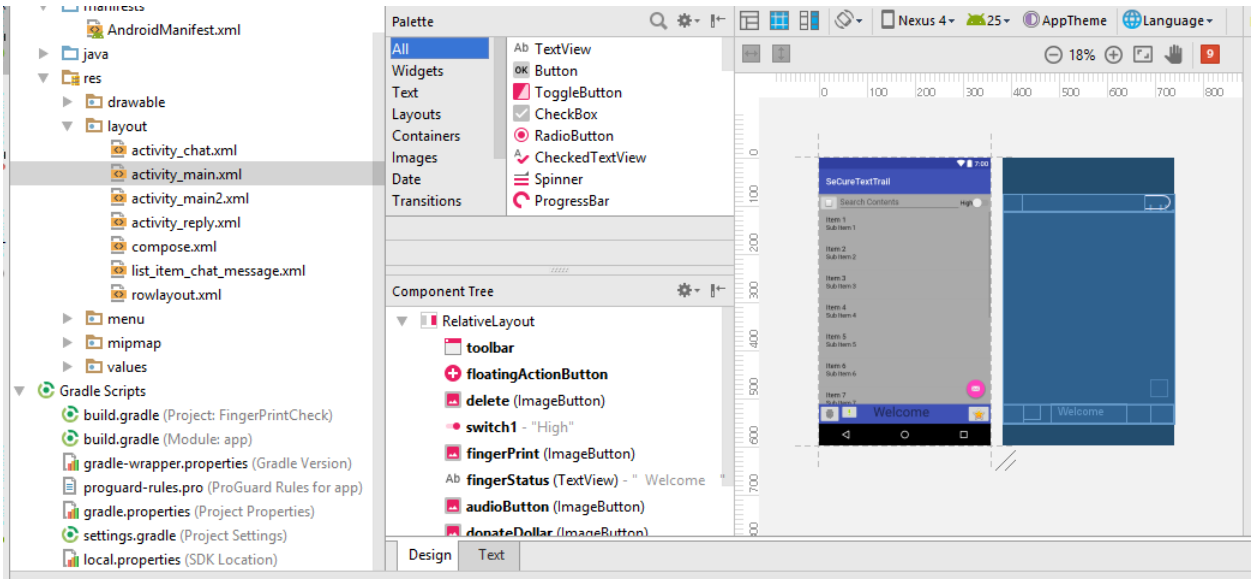


Figure 3. 3 shows the App design layout

Figure 3. 4 *CText App Main page*

Home page loaded with user text messages in an unreadable cipher format.

New Text message page,

On click the floating Button, it takes you to the new message writing section. Design as follows.

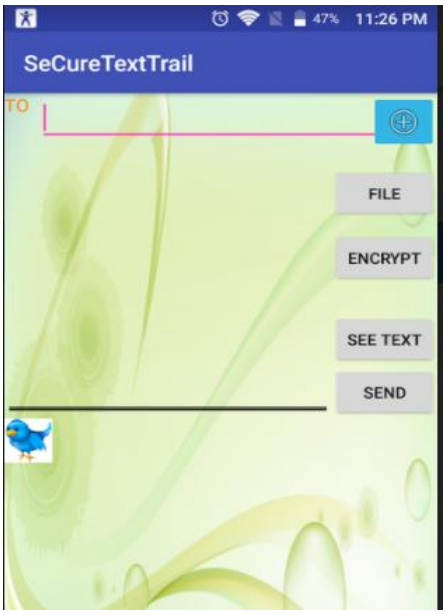


Figure 3. 5 *New Text writing page*

```
RelativeLayout

    <TextView
        android:id="@+id/textViewTo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TO  "
        android:textColor="@android:color/holo_orange_dark"
        android:textSize="16sp"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_above="@+id/msgText" />

    <EditText
        android:id="@+id/numberText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentTop="true"
        android:layout_toEndOf="@id/textViewTo"
        android:inputType="phone"
        android:textColor="@android:color/holo_purple" />

    <EditText
```

Figure 3. 6 *New Text writing page, Xml coding for design.*

In this New Message page, it let you select contacts using an **+ Button**, you can type your message, it has option to send regular message or encrypted message,

Encrypt button, your messages will be encrypted after you click this button and you can attach any file using File button,

Send Button, will send you message to the receiver, see Text lets you see your encrypted text, before sending.

Reply Activity

This page will be loaded, after you read your received text, if you hit reply button. This page will be loaded with all the conversation with that user in the chat format. This view required Android Msg Container to hold all the conversation. (stackoverflow.com, n.d.)

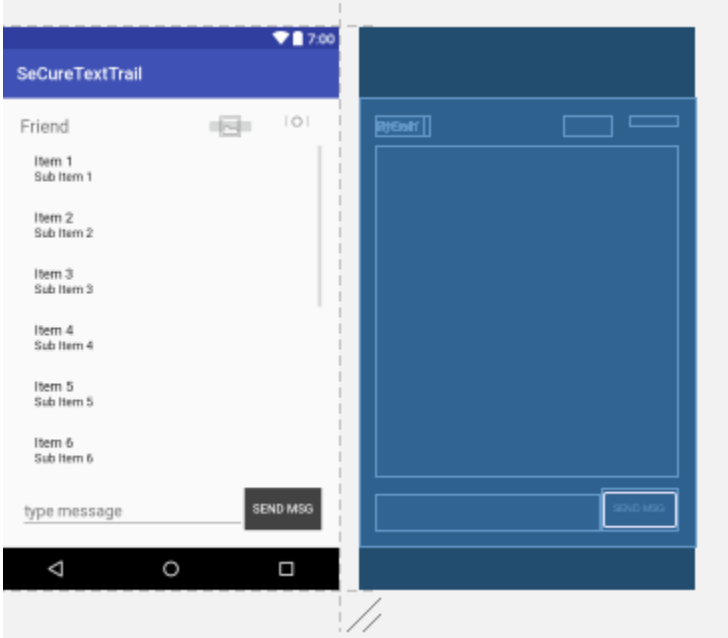


Figure 3. 7 Conversation activity blueprint

Each Text message will be loaded into the view left alignment or right alignment based on sender within the conversation.

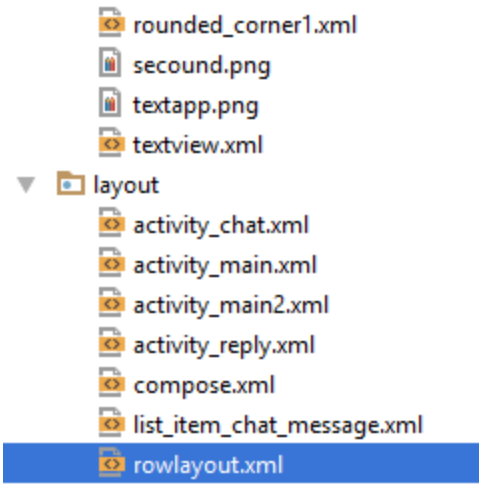


Figure 3. 8 Different XML files for different Activity within the app.

3.3 Develop an App- Coding section

This is most important stage of software lifecycle. Development, in this stage we enable life for the app, each action is written by set of commands. In this App coding is fully written in java with some extension of android libraries.

Home page is loading with user SMS messages in a cipher format, it reads the android built-in content, requires READSMS permission, which needs to define in Android manifest file as uses permission. After Read the content, it has to encrypt with the key and passes to the custom adapter.

```
context=this;
messages = (ListView) findViewById(R.id.listMessages);
send = (FloatingActionButton) findViewById(R.id.floatingActionButton);
checkAndRequestPermissions();
if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_SMS)!= PackageManager.PERMISSION_GRANTED)
{
    getPermissionToReadSMS();
}
else
{
    //refreshSmsInbox();
    smsEncListAdapter =new SMSEncListAdapter(this,cursor);
}


    ArrayList<Sms> sms = getAllSms();
    ArrayAdapter encAdapter = new EncCustomAdapter(this, R.layout.rowlayout, sms);
    messages.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
    messages.setAdapter(encAdapter);
```

Before loading an SMS content, program code checks the user permission is granted, if not it will ask user permission in a dialog box.

```
public boolean checkAndRequestPermissions()
{
    int permissionSMS = ContextCompat.checkSelfPermission(this,Manifest.permission.READ_SMS);
    int fingerprintPermission = ContextCompat.checkSelfPermission(this,Manifest.permission.USE_FINGERPRINT);
    int phonestatePermission = ContextCompat.checkSelfPermission(this,Manifest.permission.READ_PHONE_STATE);

    List<String> listPermissionsNeeded = new ArrayList<>();
    if (fingerprintPermission != PackageManager.PERMISSION_GRANTED) {
        listPermissionsNeeded.add(Manifest.permission.USE_FINGERPRINT);
    }
    if (permissionSMS != PackageManager.PERMISSION_GRANTED) {
        listPermissionsNeeded.add(Manifest.permission.READ_SMS);
    }
    if (phonestatePermission != PackageManager.PERMISSION_GRANTED) {
        listPermissionsNeeded.add(Manifest.permission.READ_PHONE_STATE);
    }
    if (!listPermissionsNeeded.isEmpty())
    {
        ActivityCompat.requestPermissions(this,
                listPermissionsNeeded.toArray(new String[listPermissionsNeeded.size()]), MY_PERMISSIONS_REQUEST_ACCOUNTS);
        return false;
    }

    return true;
}
```

SMS contents are read and form an SMS object in an Array List. Java ArrayList is a collection of objects of same data types or class, Sms is a user defined object in different java class, which we used here

ArrayList<Sms> lstSms = new ArrayList<Sms> ();
Sms objSms = new Sms ();

URI parsing is an android built in library for reading system stored data for user manipulation, SMS conversations are read from the table “MMS-SMS/Conversations”. Shown below in codes

```
Uri uriSMSURI = Uri.parse("content://mms-sms/conversations/");
Cursor c = getContentResolver().query(uriSMSURI, null, null, null, "date desc");
int totalSMS = c.getCount();

if (c.moveToFirst()) {
    for (int i = 0; i < totalSMS; i++) {

        objSms = new Sms();
        objSms.setId(c.getString(c.getColumnIndexOrThrow("thread_id")));
        objSms.setMsg(c.getString(c.getColumnIndexOrThrow("body")));
        String address =c.getString(c.getColumnIndexOrThrow("address"));
        objSms.setReadState(c.getString(c.getColumnIndex("read")));
        Date smsDayTime = new Date(Long.valueOf(c.getString(c.getColumnIndexOrThrow("date"))));
        objSms.setTime(smsDayTime.toString());
        if (c.getString(c.getColumnIndexOrThrow("type")).contains("1")) {
            objSms.setFolderName("inbox");
        } else {
            objSms.setFolderName("sent");
        }
```

Custom Adapter.

It is the extension of Array Adaptor class, which we used to load our SMS Messages in List, in this class Sms Array list takes as input and perform processing and produce the required data alone in an elegant way, and finally load it to the List in main Activity. (vogella.com, n.d.)

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View row = convertView;
    HeaderHolder holder = null;

    if(row == null)
    {
        LayoutInflater inflater = ((Activity)context).getLayoutInflater();
        row = inflater.inflate(layoutResourceId, parent, false);

        holder = new HeaderHolder();
        holder.contactName = (TextView) row.findViewById(R.id.contactName);
        holder.number = (TextView) row.findViewById(R.id.textViewSMSSender);
        holder.msg = (TextView)row.findViewById(R.id.textViewMessageBody);
        holder.time = (TextView)row.findViewById(R.id.receivedTime);
        holder.Thread_id = (TextView)row.findViewById(R.id.ThreadId);
        holder.image= (ImageView) row.findViewById(R.id.list_image);
        row.setTag(holder);
    }
    else
```

LayoutInflater takes layout Xml files as input for presenting a view.

Figure 3. 9 design of each row in an Msg List (Row Layout)

RowLayout.xml Design. It produces the output as follows in List Adapter

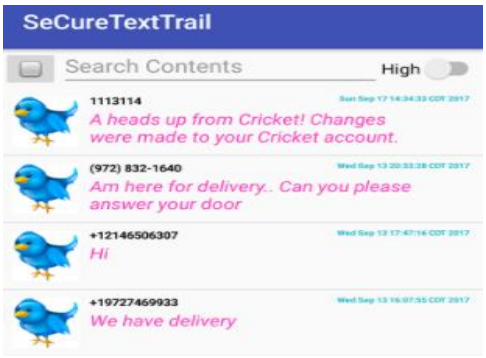


Figure 3. 10 Row layout output from the custom adaptor

Encryption & Decryption

This is the main moto of this app, where it provides strong security against privacy. For Encryption and Decryption requires a key, but the user does not know about the key. CText app itself generate a key from variables in the phone. And encryption is done all the time, Decryption happens when the user is authenticated using a fingerprint.

```
btn.setOnClickListener((view) → {
        message.setText("Use FingerPrint");
        fph.doAuth(fingerprintManager, cryptoObject);
        message.setText("click again");

        if (value == 1) {
            DecryptedInbox();
            authenticated=true;
            value=0;
        }
```

This app doesn't require separate fingerprint registration, it takes the default system registered fingerprint for verification. If there is no fingerprint saved in the system, it will tell the user to enable fingerprint, CText will not decode the user content until the user is enable fingerprint in the phone security system and validate using fingerprint.

```
fingerprintManager = (FingerprintManager) getSystemService(FINGERPRINT_SERVICE);

try {
    // Check if the fingerprint sensor is present
    if (!fingerprintManager.isHardwareDetected()) {
        message.setText("Fingerprint authentication not supported");
        return false;
    }

    if (!fingerprintManager.hasEnrolledFingerprints()) {
        message.setText("No fingerprint configured.");
        return false;
    }

    if (!keyguardManager.isKeyguardSecure()) {
        message.setText("Secure lock screen not enabled");
        return false;
    }
```

You cannot match the specific word to the cipher text, let's assume I sent hello to two different people, first hello ciphered text and second hello ciphered text will be different.

The key is different from each user, whenever the key changes, it produces different text. Encrypt method takes key and plain text as arguments. And return the cipher text.

```
public static String encrypt(String strToEncrypt, String secret)
{
    try
    {
        setKey(secret);
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKey);
        byte[] encodedBytes = cipher.doFinal(strToEncrypt.getBytes("UTF-8"));
        return Base64.encodeToString(encodedBytes,0);
    }
    catch (Exception e)
    {
```

Decryption method takes key and cipher text as arguments and returns plain text.

```
    try
    {
        setKey(secret);
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        return new String(cipher.doFinal(Base64.decode(strToDecrypt,0)));
    }
    catch (Exception e)
    {
```

Send Message method.

This method let you send the user message to the intended contact. It calls the SMS manager class in Android.

```
public void OnSendClick1(View v)
{

    keypart = msgText.getText().toString();

    if ( keypart .length() >= 1 && numberText.getText().toString().length() >= 9 ) {

        smsManager.sendTextMessage(numberText.getText().toString(),null,msgText.getText().toString(),null,null);


    }else
    {
        AlertDialog dialog = new AlertDialog.Builder(context).create();
        dialog.setTitle("Error");
        dialog.setIcon(android.R.drawable.ic_dialog_info);
        dialog.setMessage("you cant send empty text and also verify phone number");
        dialog.setButton(DialogInterface.BUTTON_POSITIVE, "ok",
```

File Attach method.

This Feature is available in both Reply Activity and New Text message writing section. Image Button is used, Gallery pic is imbedded in the button. It invokes the system file explorer app to attach the image file.

```
galleryOpen.setOnClickListener((v) → {

        OnChooseFile();

});

public void OnChooseFile()
{
    Intent photoPickerIntent = new Intent(Intent.ACTION_PICK);
    photoPickerIntent.setType("image/*");
    startActivityForResult(photoPickerIntent, SELECT_PHOTO);
}
```

Photo attach using camera.

It is same as file attach method, but this button, invokes the phone camera enables you to live capture photos or videos. Code as follows.

```
camphoto.setOnClickListener((v) → {

            //
            // openCamera();

            Intent cameraIntent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
            File dir = Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DCIM);
            String currentDateTimeString = DateFormat.getDateTimeInstance().format(new Date());
            File output = new File(dir,currentDateTimeString+".png");

            cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(output));
            String path =output.getAbsolutePath();
            startActivityForResult(cameraIntent, TAKE_PHOTO);


});
```

Both the above function requires system permission to access camera and read gallery and store images. Depends on the app given permission by the user, it invokes right permission before proceeding with the action, if the user dismisses the permission, File attach/camera action will be dismissed, app will refresh the current page.

```
alertDialog.setButton(AlertDialog.BUTTON_POSITIVE, "ALLOW",
        (OnClickListener) (dialog, which) → {
                dialog.dismiss();
                ActivityCompat.requestPermissions(ReplyActivity.this,
                        new String[]{Manifest.permission.CAMERA},
                        MY_PERMISSIONS_REQUEST_CAMERA);
        });
alertDialog.show();
```

Alert Notification

Whenever App received a message. It will notify the user with defined sound in the system, it will pop up in the notification panel at the top. Without revealing contact name or phone number. It just notifies that you got text message.

```
public void addNotification()
{
    NotificationCompat.Builder builder =
            new NotificationCompat.Builder(this)
                    .setSmallIcon(R.drawable.image)
                    .setContentTitle("Msg received")
                    .setContentText("To View the Text content , use APP.");

    Intent notificationIntent = new Intent(this, MainActivity.class);
    PendingIntent contentIntent = PendingIntent.getActivity(this, 0, notificationIntent,
            PendingIntent.FLAG_UPDATE_CURRENT);
    builder.setContentIntent(contentIntent);

    // Add as notification
    NotificationManager manager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
    manager.notify(0, builder.build());
}
```

Setting CText as default App.

In the Home page, Right-lower corner, there is an image button to set as default app, when you click that button, it lets you choose the default SMS app. After the Android KitKat version. Only the default app can able to delete the text message or thread.

```
donate.setOnClickListener((view) → {
        Log.i("MainActivity", "Button Pushed");
        Intent intent = new Intent(Telephony.Sms.Intents.ACTION_CHANGE_DEFAULT);
        startActivity(intent);
});
```

Broadcast Receiver.

To receive text message in the app, needs to define the broadcast activity class. It will monitor the incoming text messages, once it's received, the app will notify, app data will be refreshed to get updated. It extends android Broadcast Receiver Class.

```
public class SMSBroadcastActivity extends BroadcastReceiver {
    public static final String SMS_BUNDLE = "pdus";
    public void onReceive(Context context, Intent intent)
    {
        Bundle intentExtras = intent.getExtras();

        if (intentExtras != null) {
            Object[] sms = (Object[]) intentExtras.get(SMS_BUNDLE);
            String smsMessageStr = "";
            for (int i = 0; i < sms.length; ++i) {
                String format = intentExtras.getString("format");
                SmsMessage smsMessage = SmsMessage.createFromPdu((byte[]) sms[i], format);
                String smsBody = smsMessage.getMessageBody();
                String address = smsMessage.getOriginatingAddress();
                smsMessageStr += "SMS From: " + address + "\n";
                smsMessageStr += smsBody + "\n";
            }

            MainActivity inst = MainActivity.instance();
            inst.DecryptedInbox();
            inst.addNotification();
        }
    }
}
```

This app is designed, and coding done to ensure the user privacy 100% all the time. In future, plan to enable voice lock before going to sleep, when you are sleep, fingerprint will not unlock your message. This Feature is not yet implemented. Noted for future work.

3.4 Testing

This is the final stage before release the software to the market in agile methodology. This stage is crucial as development, since it going to test in some multiple scenarios and make sure this APP running fine without any bugs. In this section, we can able to find if there any logical flaws of the app.

This App is developed in Android studio, and designed to support all phones after android version 6.0 Marshmallow.

Requirements to install this app.

- Android Phone
- Version 6.0 or better
- Fingerprint scanner built-in phone

Internet connection is required to download from play store.

Initial Testing using Android Emulator

Android studio provides the basic functionality testing for design and other testing without use of network can be achieved using Emulator. When you run the application, it shows the emulator option dialog, where you can choose the deployment target phone.

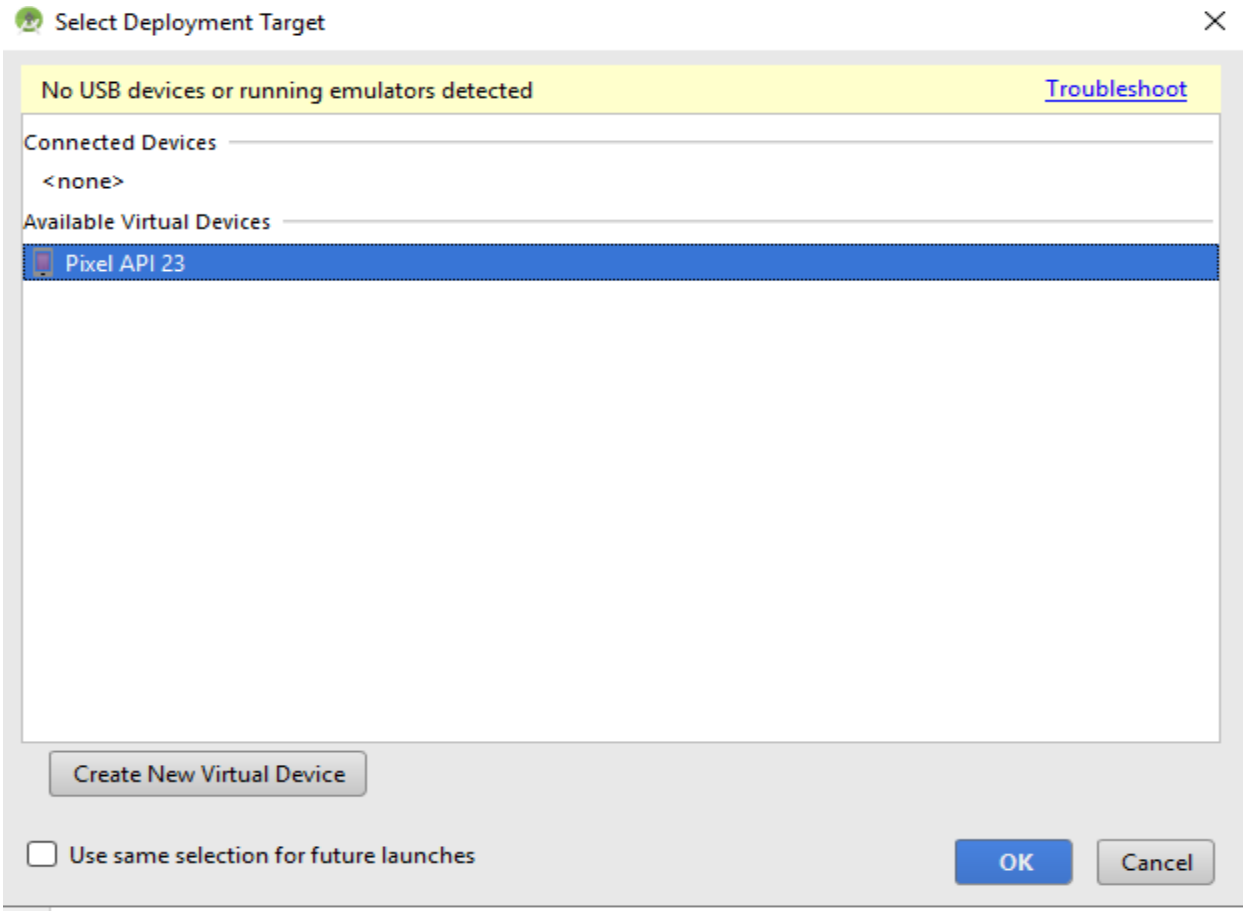


Figure 3. 11 Android Emulator, device selection

Here Android pixel is chosen.

In case the phone is physically connected via data cable, then then phone model will show in the above dialog box. Phone should be connected via USB debugging mode.

Enable debugging mode steps. Later android versions after 4.2, debugging mode is hidden. To enable to debugging options.

1. Settings → About phone → locate a Build Number.
    - Hit the build Number 7 times continuously

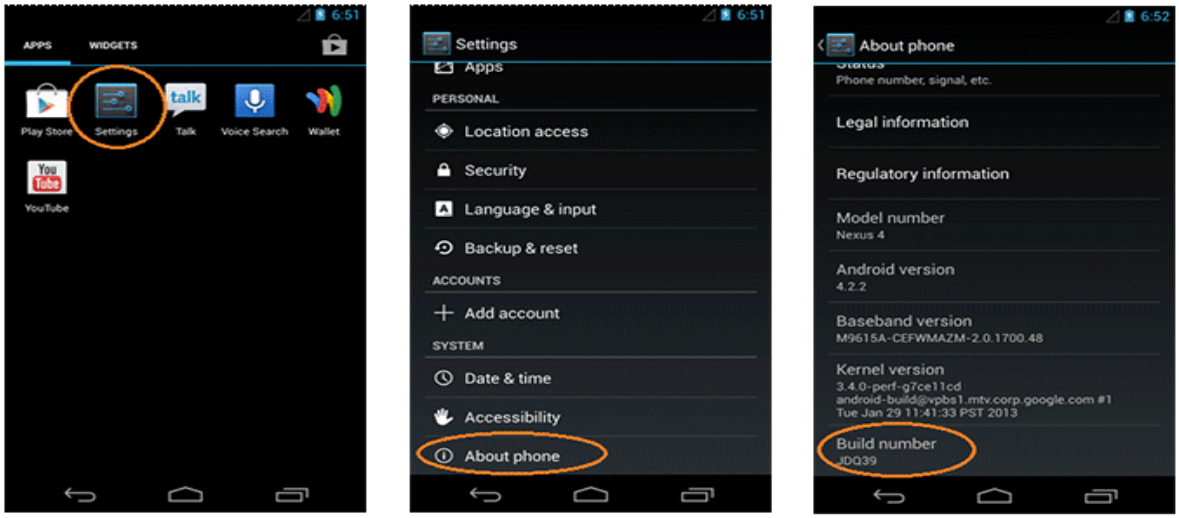


Figure 3. 12 steps for enable debugging option (kingoapp, n.d.)

When you are done, pop will appear that "Now you are a developer"

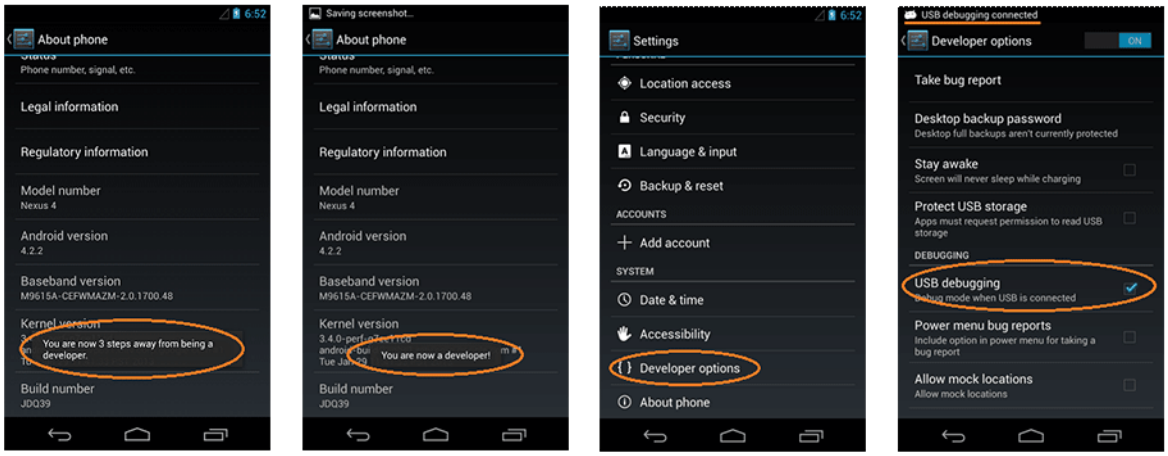


Figure 3. 13 steps for to enable USB debugging to view android studio. (kingoapp, n.d.)

Now you can able to see the developer options in settings. Settings→Developer Options→USB Debugging →click the USB debugging Checkbox.

Once you have selected deployment target in Android Emulator, it takes couple of minutes of run the emulator and load the app based on system configuration. New Emulator window opens

Figure 3. 14 CText App installed in the phone. Named SecureTextTrail.

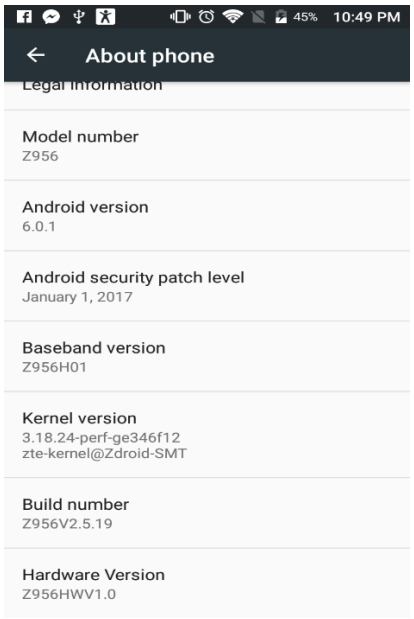
In This pic above, you can see the secure text app is installed. Despite the fact app installed, CText App cannot be test using an Emulator, Reasons are CText app work on SMS services and Fingerprint scanning, both requires physical device.

Test cases using an Android phone.

Test objectives can test all the defined functionalities in this app. Test cases are.

1. Able to request user permissions for reading SMS, Reading Phone contacts.
2. Able to Read existing SMS in cipher mode
3. Able to view the original SMS after the fingerprint validation success.
4. View whole SMS Thread conversation
5. No fingerprint configured mobile, App alert the user.
6. Able to send normal SMS messages and ciphered Messages.
7. Receive Messages in another phone
8. Able to decrypt the message by another user fingerprint without key.
9. Notification alert
10. Default App set up permission asking the user.

Secure Text App testing using the device ZTE phone.



Model: ZTE Z956. Android Version: 6.0.1

Test case 1. Request User Permission.

Install CText app, in android device. And open in android device, it should request user permission for Reading SMS, Reading phone, and contacts.

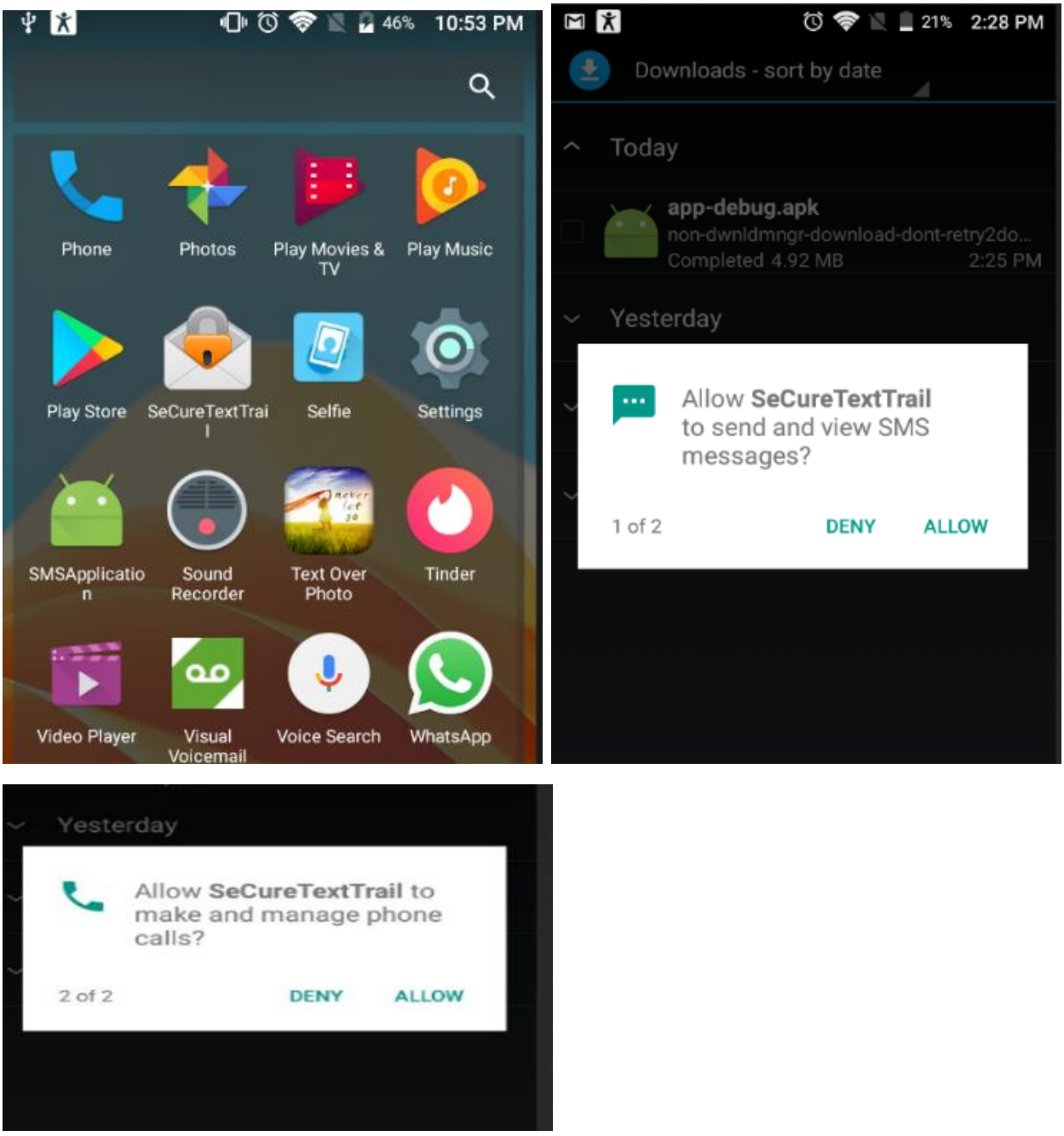


Figure 3. 15 Test case 1

Test case 1 success, result is positive as expected.

Test case 2. Read Existing SMS in Cipher mode

App should run without any issues and open Existing phone SMS in a cipher mode, All Messages should be encrypted to unreadable format.

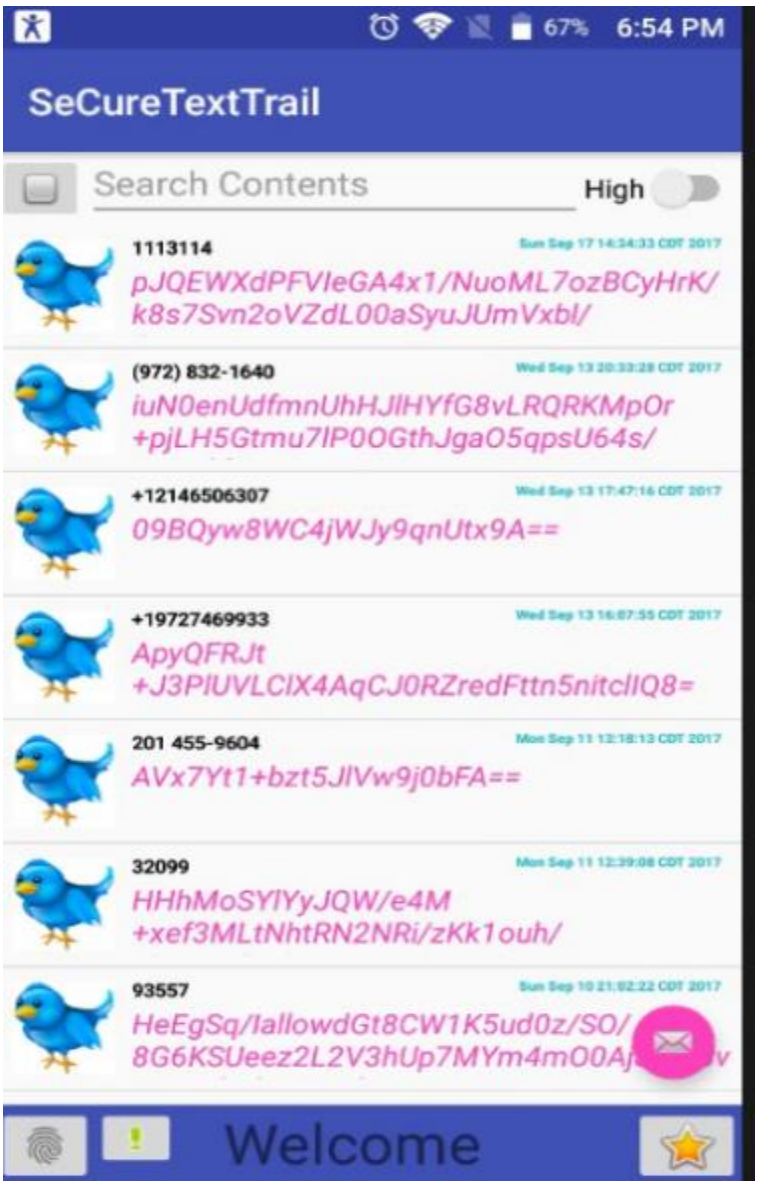


Figure 3. 16 Test case 2

App reads all the existing message, from the date format we can identify and now the contents are unreadable cipher format

Test case 2 success, result is positive as expected.

Test case 3. Read Existing SMS in Text mode after the user validation

App should run without any issues and open Existing phone SMS Messages should be able to read after the Fingerprint validation.

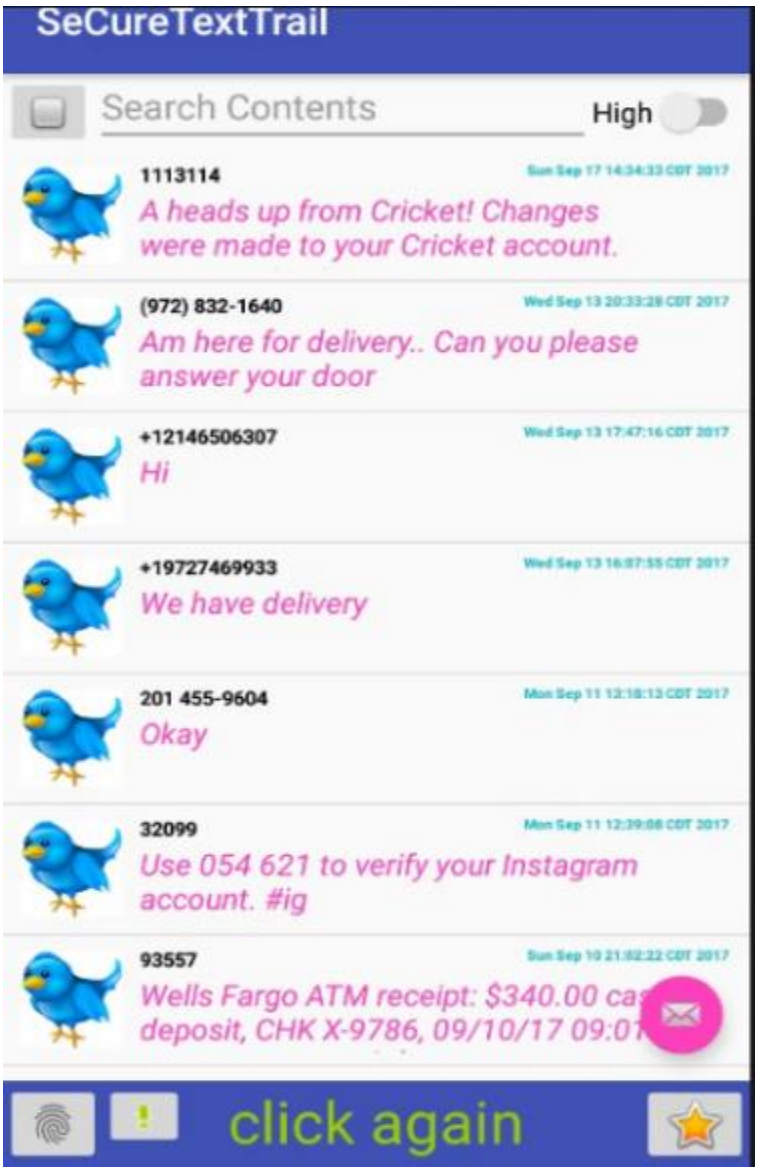


Figure 3. 17 Test case3 output

App reads all the existing message, from the date format we can identify and now the contents are readable text format

Test case 3 success, result is positive as expected.

Test case 4. Read SMS conversation in chat view

If the user selects any message in the list, the SMS conversation should open in new activity with chat view with the option to send messages to the user.

Figure 3. 18 Test case 4 output

In this conversation view, received messages are on left side, send messages are on right side.in this chat view user can reply instant to the message and attach the pic.

Test case 4 success, result is positive as expected.

Test case 5. Alert the user about fingerprint configuration

If there is no fingerprint configured in the phone, it will notify the user that No fingerprint configured in the status bar. Messages unlock button will be disabled.

Figure 3. 19 Test case 5 output

In this view, it is notified to the user, until user configure the fingerprint, App wont the user to see the original message.

Test case 5 success, result is positive as expected.

Test case 6. Send Normal SMS and Cipher SMS using CText app.

In the app main page, there is a floating Image button with mail image on it. When the user clicks that button, it should take to the compose activity.

Figure 3. 20 Test case 6 output

Figure 3. 21 Test case 6 output

In this compose page, send button do the same function, it will let the user to send text message. But before there is an option to convert regular message or cipher message using encrypt button,

Test case 6 success, result is positive as expected.

3.5 Evaluation

This is the final stage of software development life cycle defined by agile methodologies. This chapter evaluates the solution of this project defined during planning stage, it analyses the requirement and meets the demand. And any other considerations during the development of this App.

The technologies we used is java, it's a well-known programming language which is run in most of the platform, and it provides the output of platform independent software. Java coding is used for ciphering operation defined in this app. Which requires complex mathematical analysis and perform quick steps for encryption is using AES key length of 128 bit. Java library "javax crypto. Cipher. * " . This coding takes bit longer time for compile first time and when it is running large set of data for encryption/decryption operations, this app takes few more seconds to display the ciphered SMS page. Especially when you are switching the apps faster. Still the app not speed is not compromising, it provides consistent output all the time of the defined functions.

Another Technology is used is Android, it's mainly used for app design, App design is simple and lightweight, so It does not lack anywhere in app different activities. App is in initial phase, so few important activities only enabled for the defined objective to work. Later plan to add extra features. During the initial stage of design, I face lot of issues for orientation change, I come up with relative layout fix most of the issues, and this design invokes most of the system function to perform some activities such as selecting file from a gallery or taking a pic from camera. If the user has default app for these operations, App speed will be moderately higher than who don't have default app. whenever you perform that operation, it must open the external app to perform this action.

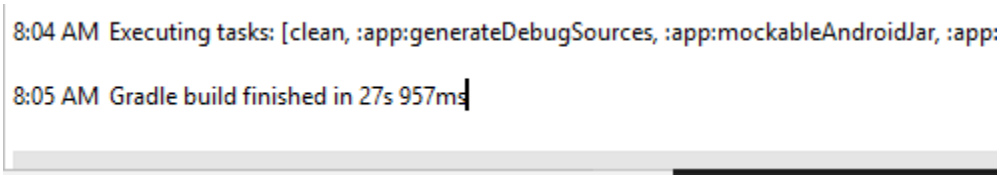
Another important factor is human interface. Fingerprint is required to open this app; fingerprint scanning is fast authentication method for most of the phones. Still it purely depends how the user handles that. When user perform this operation, the finger is free from oil or water or some dust, it makes the scanner provides wrong result, despite the user is genuine, scanner may it read as false matching. It's not the CText app issue not performing the decryption, user must provide valid verification using their fingers right way.

Execution Time of CText

This app running time is android platform is consistent, Initial build process taking 68 secs. And running time after the build is 55s to 8s. The application installs and launch activity in android phone is 1 or 2 seconds.

Android studio uses gradle service to build the app, when you build an app. The following process will be executed

8:04 AM Executing tasks: [clean, :app:generateDebugSources, :app:mockableAndroidJar, :app:prepareDebugUnitTestDependencies, :app:generateDebugAndroidTestSources, :app:compileDebugSources, :app:compileDebugUnitTestSources, :app:compileDebugAndroidTestSources]

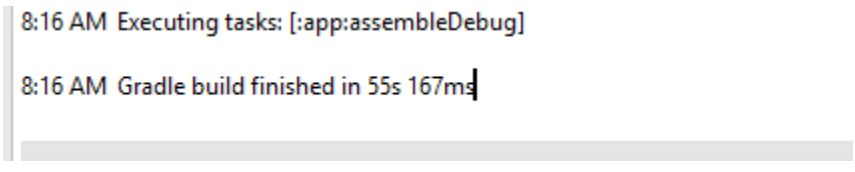


This all process takes 27s 955milliseconds.

Android Monitor logs from android studio.

```
01-17 16:16:25.051 19440-19440/com.example.karthikeyan.fingerprintcheck V/BoostFramework: BoostFramework() : mPerf = com.qualcomm.qti.Performance@d98415c
01-17 16:16:25.051 19440-19440/com.example.karthikeyan.fingerprintcheck V/BoostFramework: BoostFramework() : mPerf = com.qualcomm.qti.Performance@6e60c65
01-17 16:16:25.671 19440-19533/com.example.karthikeyan.fingerprintcheck D/OpenGLRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVED: true
01-17 16:16:25.731 19440-19533/com.example.karthikeyan.fingerprintcheck I/Adreno-EGL: <qeglDrvAPI_eglInitialize:379>: EGL 1.4 QUALCOMM build:  (I2fa6926f94)
                                                                                    OpenGL ES Shader Compiler Version: XE031.08.00.00
                                                                                    Build Date: 11/13/16 Sun
                                                                                    Local Branch:
                                                                                    Remote Branch:
                                                                                    Local Patches:
                                                                                    Reconstruct Branch:
01-17 16:16:25.731 19440-19533/com.example.karthikeyan.fingerprintcheck I/OpenGLRenderer: Initialized EGL, version 1.4
01-17 16:16:25.931 19440-19440/com.example.karthikeyan.fingerprintcheck W/art: Before Android 4.1, method int
android.support.v7.widget.ListViewCompat.lookForSelectablePosition(int, boolean) would have incorrectly overridden the package-private method in
android.widget.ListView
```

When the user installs CText app in phone via USB Time taken is 55s 167ms



4 Conclusion

CText/SecureText App ensures user privacy in all the time, data in phone is secured irrespective the user knowledge of advanced concepts. It is developed and tested in live environment for traditional texting and found that security feature was consistent and gave positive result overall, proved that the project is success, this project vision is completed through master thesis which is defined during the planning phase. Using biometric fingerprint and encryption this app provides user privacy in this phase.

At initial phase, I faced lot of issues for choosing an encryption algorithm and more challenges, how to implement in mobile devices where there is limited resources and computation power to support heavy weight javax cipher library. This class has some built-in methods for AES key definition and encryption methods. And it's customized to support this project.

I want to publish this app in Android store for users free of charge, based on customers usage and support, plan to add more attractive features such as smart reading using face recognition and voice support to perform basic functions. In this stage, encryption supports in text level only, in future planning to support multimedia content as well. From this we have concluded that, using biometric we can able to achieve more privacy and security in mobile or portable devices.

Apart from the future work, App development thesis to "Ensure user privacy and security in mobile devices using Biometrics "vision is success and completed within the stipulated time as part of master thesis.

References

Anon., 2017. *wiki/Android_(operating_system).* [Online]
Available at: https://en.wikipedia.org/wiki/Android_(operating_system)
[Accessed 18 November 2017].

Anon., 2017. *wiki/Biometrics.* [Online]
Available at: https://en.wikipedia.org/wiki/Biometrics
[Accessed 25 November 2017].

Anon., October 2016. *https://techdifferences.com/difference-between-des-and-aes.html.* [Online]
Available at: https://techdifferences.com/difference-between-des-and-aes.html
[Accessed 18 November 2017].

kingoapp, n.d. *how-to-enable-usb-debugging-mode-on-android.htm.* [Online]
Available at: https://www.kingoapp.com/root-tutorials/how-to-enable-usb-debugging-mode-on-android.htm
[Accessed 5 January 2018].

Larson, M., 2104. *what-are-the-differences-between-des-and-aes-encryption.* [Online]
Available at: https://info.townsendsecurity.com/bid/72450/what-are-the-differences-between-des-and-aes-encryption
[Accessed 20 November 2017].

Leith, G., n.d. *https://seelio.com.* [Online]
Available at: https://seelio.com/w/1wd0/aes-encryption-and-decryption
[Accessed 2 November 2017].

National Institute of Standards and Technology, 2017. *csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf.* [Online]
Available at: https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf
[Accessed 18 November 2017].

Rouse, M., n.d. *agile-software-development.* [Online]
Available at: http://searchsoftwarequality.techtarget.com/definition/agile-software-development
[Accessed 2 Decemeber 2017].

stackoverflow.com, n.d. *21091611/how-to-read-sms-conversation-in-android.* [Online]
Available at: https://stackoverflow.com/questions/21091611/how-to-read-sms-conversation-in-android
[Accessed 25 Dec 2017].

Tumbleson, M., 2016. *19-text-messaging-stats-that-will-blow-your-mind/.* [Online]
Available at: https://teckst.com/19-text-messaging-stats-that-will-blow-your-mind/
[Accessed 26 August 2017].

vogella.com, n.d. *AndroidListView.* [Online]
Available at: http://www.vogella.com/tutorials/AndroidListView/article.html
[Accessed 28 August 2016].