VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi, Karnataka – 590018

A

Mini Project Report

On

"ENCRYPTION AND DECRYPTION USING VARIOUS ALGORITHMS "

Submitted By

BHARATH B (1KT20IS002)

Under the Guidance

of

Ms. SINDHU G

Assistant Professor

Dept. of ISE, SKIT

SRI KRISHNA INSTITUTE OF TECHNOLOGY

Department of Information Science and Engineering

No.29, Hesaraghatta Main Road, Chimney hills, Chikkabanavara P.O., Bengaluru – 560090

2022-2023

SRI KRISHNA INSTITUTE OF TECHNOLOGY

No.29, Hesaraghatta Main Road, Chimney hills, Chikkabanavara P.O., Bengaluru –560090

Department of Information Science and Engineering

CERTIFICATE

Certified that the mini project work prescribed in **18CSMP68** entitled **"ENCRYPTION AND DECRYPTION USING VARIOUS ALGORITHMS"** carried out by **BHARATH B (1KT20IS002) ,** bonafide student of **Sri Krishna Institute of Technology**, Bengaluru in partial fulfilment for the award of **Bachelor of Engineering** in **Information Science and Engineering** of the **Visvesvaraya Technological University** Belagavi during the year 2022-23. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements with respect to mini project work prescribed for the said Degree.

Signature of the Guide	**Signature of the HOD**
Ms. Sindhu G	**Dr. Hemalatha K.L**
Assistant Professor	Professor and HOD
Dept of ISE, SKIT	Dept of ISE, SKIT

EXTERNAL VIVA

Name of the Examiners	**Signature with date**
1.	
2.	

ABSTRACT

An Android-based system for data Encryption and Decryption using various algorithms. The objective is to provide users with a secure and versatile encryption framework. The system allows users to choose from a range of algorithms, including AES, DES, Playfair, and Hashing, enabling customization based on security requirements and device capabilities. The implementation leverages the Android Security Library to ensure compatibility with the Android platform. Performance and security evaluations are conducted to assess efficiency and effectiveness against common cryptographic attacks. The findings contribute to mobile data security and offer a comprehensive solution for safeguarding sensitive information on Android devices.

ACKNOWLEDGEMENT

The completion of mini project work brings with a sense of satisfaction, but it is never complete without thanking the persons responsible for its successful completion.

At the outset, I express my most sincere grateful acknowledgment to the holy sanctum **"Sri Krishna Institute of Technology",** the temple of learning, for giving me an opportunity to pursue the degree course in Information Science and Engineering and thus helping me in shaping my career.

I extend my deep sense of sincere gratitude to **Principal, Dr. Mahesha K,** Sri Krishna Instituteof Technology**,** Bengaluru, for providing me an opportunity to continue my higher studies.

I express my heartfelt sincere gratitude to **Dr. Hemalatha K.L**, **Professor and HOD, Department of Information Science and Engineering**, Sri Krishna Institute of Technology**,** Bengaluru,for her valuable suggestions and support.

I extend my special in-depth, sincere gratitude to my guide **Ms. Sindhu G, Assistant Professor, Department of Information Science and Engineering**, Sri Krishna Institute of Technology**,** Bengaluru, for her constant support and valuable guidance for completion of the mini project work.

I would like to thank all the teaching and non-teaching staff members in our **Department of Information Science and Engineering**, Sri Krishna Institute of Technology**,** Bengaluru, for their support.

Finally, I would like to thank all our friends and family members for their constant support, guidance and encouragement.

BHARATH B (1KT20IS002)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction to Android

Android is an operating system and programming platform developed by Google for smartphones and other mobile devices (such as tablets). It can run on many different devices from many different manufacturers. Android includes a software development kit for writing original code and assembling software modules to create apps for Android users. It also provides a marketplace to distribute apps. Altogether, Android represents an ecosystem for mobile apps. It provides a touch-screen user interface (UI) for interacting with apps. Android's user interface is mainly based on direct manipulation, using touch gestures such as swiping, tapping and pinching to manipulate on-screen objects.

In addition to the keyboard, there's a customizable virtual keyboard for text input. Android can also support game controllers and full-size physical keyboards connected by Bluetooth or USB. It is designed to provide immediate response to user input. Besides a fluid touch interface, the vibration capabilities of an Android device can provide haptic feedback. Internal hardware such asaccelerometers, gyroscopes and proximity sensors, are used by many apps to respond to additional user actions. These sensors can detect rotation of the screen from portrait to landscape for a wider view or it can allow the user to steer a virtual vehicle in a racing game by rotating the device as if it were a steering wheel.

1.2 Android Studio

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environmentfor software, and incorporates its code editing and developer tools. To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates and GitHub integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

1.3 Features of Android Studio

- Flexible Gradle-based build system and it has a fast and feature-rich emulator for app testing.
- Android Studio has a consolidated environment where we can develop for all Android devices.
- Apply changes to the resource code of our running app without restarting the app.
- Android Studio provides extensive testing tools and frameworks.
- It supports C++ and NDK.

1.4 Android Studio Project Structure

The Android Studio project contains one or more modules with resource files and source code files. These include different types of modules:

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

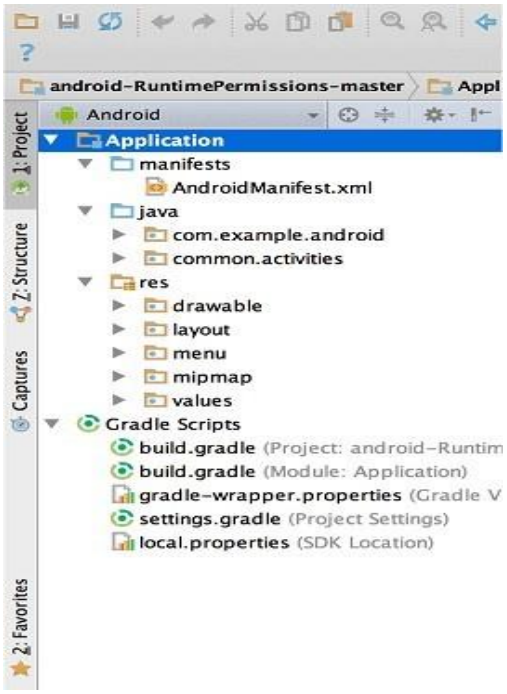

Fig.1.1: Android Project Layout

Fig.1.1 shows that by default, Android Studio displays our project files in the Android project view. This view is formed by modules to provide quick access to our project's key source files.

1.5 Android Studio User Interface

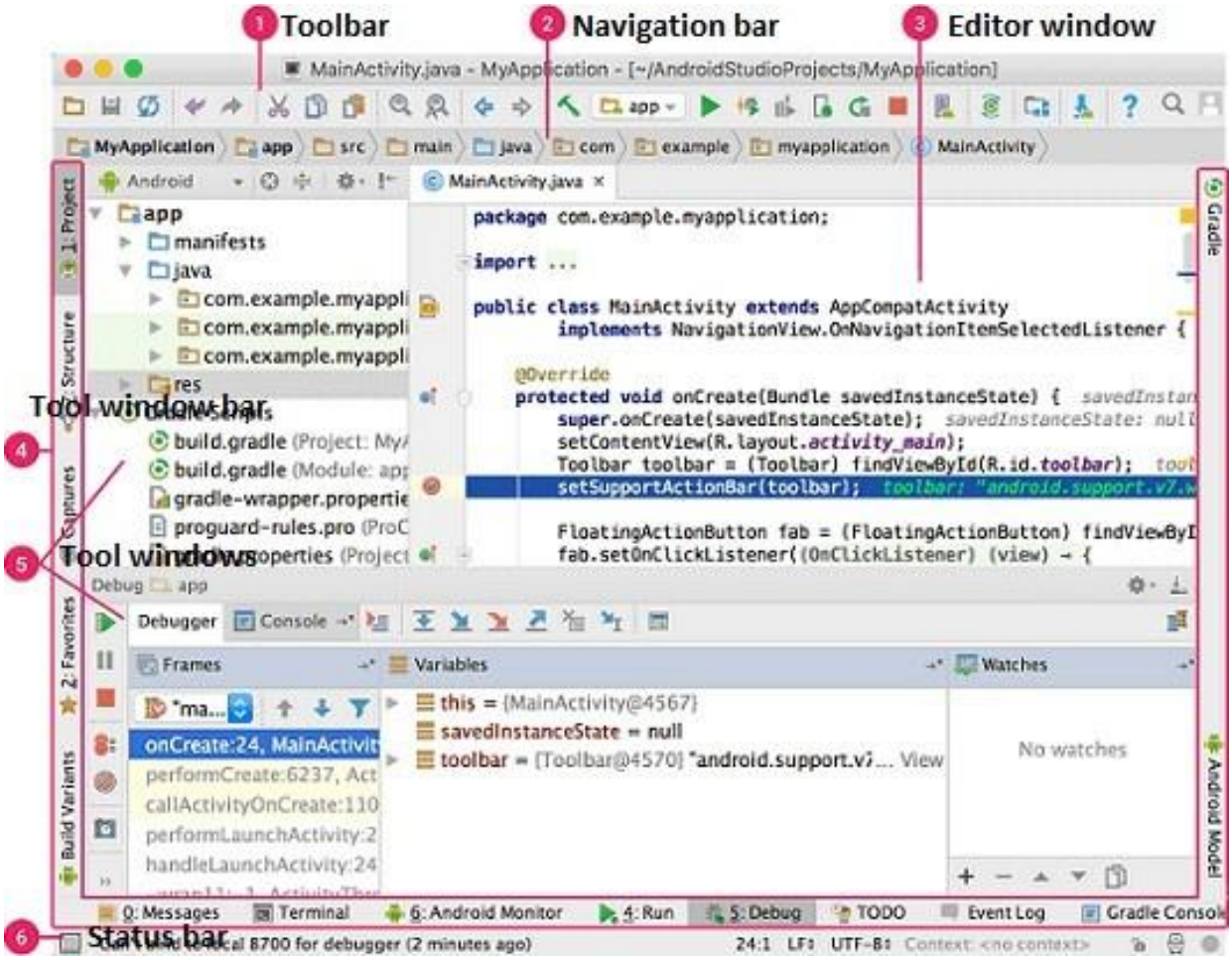

Fig.1.2: Android Studio UI

Fig.1.2 shows that the Android Studio main window contains the several logical areas.

1. Toolbar provides us a wide range of actions, which includes running apps and launching Android tools.
2. The navigation bar helps in navigating our project and open files for editing. It gives a compact view of s structure visible in the Project window.
3. The editor window is a space where we can create and modify our code. On the basis of the current file type, the editor can change. While viewing a layout file, the editor displays the Layout Editor.
4. The tool window bar runs around the outside the IDE window and contains buttons that allow as to expand and collapse individual tool windows.

5. The tool windows provide us access specific tasks like search, project management, version control, and more. we can able to expand and collapse them.
6. The status bar displays the status of our project and IDE itself, as well as any messages or warnings.

1.6 Introduction to Encryption and Decryption

In today's interconnected digital landscape, where the exchange and storage of sensitive information have become commonplace, ensuring data security has become a top priority. Encryption and decryption techniques serve as the backbone of safeguarding confidential data from unauthorized access and maintaining its integrity. This report provides a comprehensive introduction to encryption and decryption, exploring their fundamental concepts, underlying principles, and practical applications. By grasping the basics of encryption and decryption, individuals and organizations can make informed decisions to protect their data and preserve privacy in an increasingly interconnected world.

The report covers topics such as encryption algorithms, key management, symmetric and asymmetric encryption, digital signatures, and the applications of encryption and decryption in secure communication, data protection, and online transactions. It also addresses emerging challenges, such as the impact of quantum computing on encryption, and offers insights into future advancements in encryption technologies. By understanding and implementing encryption and decryption best practices, individuals and organizations can fortify their digital infrastructure, shield sensitive information, and mitigate the risks associated with unauthorized access and data breaches.

CHAPTER 2

FEASIBILITY STUDY

The feasibility study was intended to examine the current system and determine whether there was need for a new system to replace it or not. It tended to check whether the current system was viable. Basically, this was meant to analyze the feasibility of a new system through cost-benefit analysis. It included: Technical Feasibility, Economical Feasibility, Legal Feasibility and Social Feasibility.

2.1 Economical Feasibility

2.2 Legal Feasibility

2.3 Technical Feasibility

2.4 Social Feasibility

2.1. Economical Feasibility

This study assesses the economic feasibility of implementing an Android-based encryption and decryption system with multiple algorithms. The analysis focuses on evaluating the costs involved in development, deployment, and maintenance. It also explores potential cost savings and benefits associated with improved data security. Factors such as development costs, licensing fees, hardware requirements, and ongoing maintenance expenses are considered. The study examines potential revenue streams, market demand, and barriers to adoption. The findings provide decision-makers with insights into the financial viability, return on investment, pricing strategies, and cost optimization opportunities for the proposed system.

2.2 Legal Feasibility

This study assesses the legal feasibility of implementing an Android encryption and decryption system. It ensures compliance with data protection and privacy regulations, evaluates restrictions on cryptographic algorithms, and addresses legal risks associated with encryption capabilities. Insights from legal experts and consideration of recent developments help decision-makers ensure compliance and mitigate legal risks during system implementation.

2.3 Technical Feasibility

This study explores the technical feasibility of implementing encryption and decryption on Android devices. It assesses compatibility with various Android devices, efficiency of algorithm implementation, and integration of cryptographic functionalities. Factors such as hardware requirements, performance metrics, and availability of cryptographic resources are considered. Experiments on Android devices provide valuable insights for optimizing system design and decision-making.

2.4 Social Feasibility

This study assesses the social feasibility of implementing an Android encryption and decryption system. It examines user attitudes towards data privacy and security, considering acceptance, trust, and ethical considerations. The impact on user experience and convenience is evaluated, addressing potential barriers to adoption. Ethical considerations surrounding privacy, law enforcement, and social equity are also analyzed. Insights from user surveys and ethical frameworks aid decision-makers in understanding user perceptions, addressing concerns, and aligning the system with societal expectations.

CHAPTER 3

SYSTEM DESIGN

3.1 System Architecture of Encryption and Decryption

This section presents a concise overview of the system architecture for an Android-based encryption and decryption solution. The architecture includes key components such as the user interface, encryption and decryption engine, key management module, Android Security Library, and data storage. The user interface enables users to specify encryption parameters, while the encryption and decryption engine processes the data using selected algorithms. The key management module ensures secure key generation and storage. The Android Security Library provides cryptographic functions, and the data storage component securely stores the encrypted data. Together, these components facilitate a seamless and secure encryption and decryption experience on Android devices.

The system architecture consists of several key elements. Firstly, the user interface allows users to specify encryption parameters, such as the algorithm, key size, and encryption mode. This component ensures a user-friendly experience and facilitates customization.

The Encryption and Decryption Engine forms the core of the system. It encompasses the selected encryption algorithm(s), implementing the actual encryption and decryption processes. This engine interacts with the user interface to receive encryption parameters and deliver the encrypted or decrypted data.

The Key Management module is responsible for securely generating, storing, and retrieving encryption keys. It ensures that keys are properly protected and accessible only to authorized users.

3.2 Function Design of Encryption and Decryption

This section presents a simplified overview of the function design for an Android encryption and decryption system. The design includes key functions essential for the system's operation, ensuring secure data encryption and efficient decryption. These functions encompass encryption, decryption, algorithm selection, key management, user interface, and compatibility, allowing users to securely encrypt and decrypt their data on Android devices.

3.3 Data Flow Diagram of Encryption and Decryption

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an Information System. A data flow diagram can also be used for the visualization of Data Processing. It is common practice for a designer to draw a context- level DFD first which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

This System describes how the system is divided into subsystems, each of which deals with one or more of the data flows to or from an external agent and which together provide all functionality of Encryption and Decryption.

Fig.3.1 shows the flow of data in which the text is encrypted and decrypted in text form with same key.

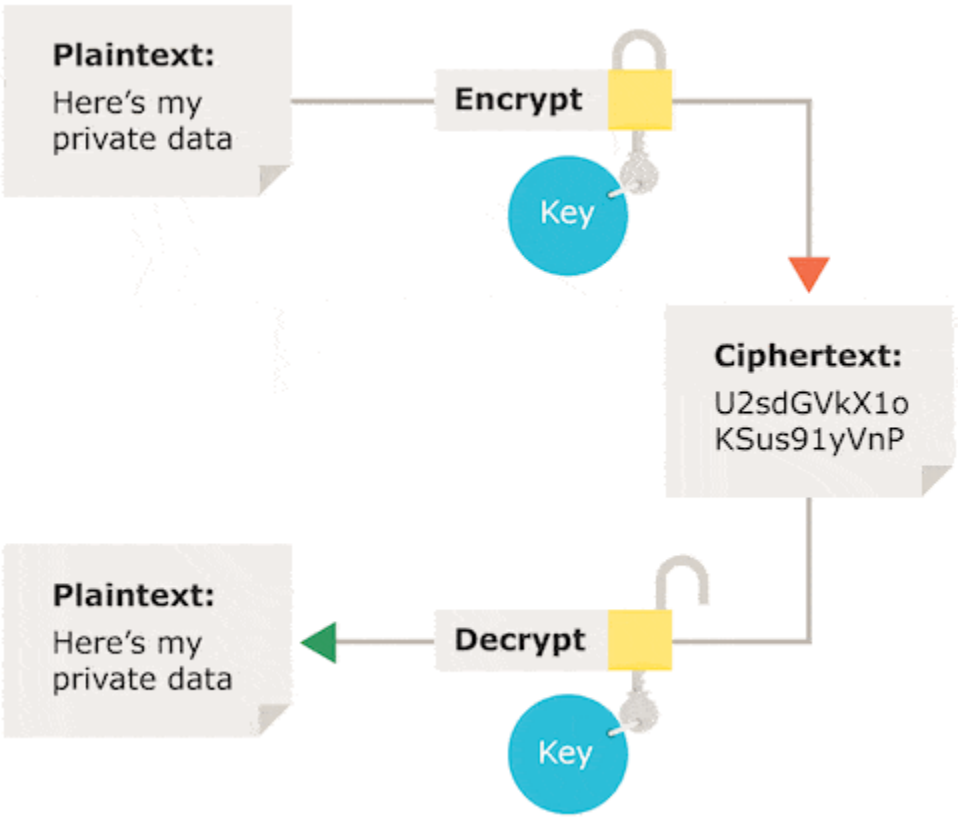

Fig.3.1: Data Flow Diagram for Encryption and Decryption

CHAPTER 4

IMPLEMENTATION

The Software Requirement Specification is a document that describes the external requirement for any system. The requirement analysis has to identify the requirements by taking to the users and understanding their needs. The inputs are to be gathered from different resources, these inputs may be inconsistent. The requirement phase translates the ideas in the minds of the users into a formaldocument, with addition to the admin dashboard. In simple words, it is a set of technologies that are used in developing the user interface of web applications and webpages. With the help of front-end technologies, developers create the design, structure, animation, and everything that you see on the screen while opening up a website, web application or mobile app.

4.1 Software Requirements

Front End : XML

Backend : Java

User OS : Windows

Tools :

- Android SDK (Software Development Kit), ADT (Android Development Tool).
- JDK : Java Runtime Environment virtual machine, Java Development Kit (JDK).

4.2 Hardware Requirements

Ram Capacity : 512 (min)

Hard Disk : 80 GB (min)

Processor : Intel Core i5 (min)

System Type : 64-bit OS

Installation steps of the developing environment:

- Install the Java virtual machine JDK version – 7.
- Install the Android SDK: first download the Android SDK.
- Download address: http://developer-android-com/sdk/index-html.
- Input SDK tools path in SDK location: D: \ android \ software \ android SDK– Windows.
- The Android environment is set up successfully.

4.3 Android Studio

In recent times, Android became the world's most popular operating system for variousreasons. As an Android programmer, I want to share what the Android Studio is? Android Studio is anIDE for Google Android Development launched on 16th May 2013, during Google's I/O 2013 event. Android Studio contains all the Android tools to design, test, debug, and profile your application. TheAndroid Studio uses Gradle to manage your project, a Build Automation Tool.

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on window, macOS and Linux based operating systems or as a subscription-based service in 2020.It is a replacement for Eclipse Android Development Tools asthe primary IDE for native Android application development.

4.4 Front End Technology

4.4.1 XML

XML or Extensible Markup Language is a text language that can be used to describe the behavior of programming languages that process them. XML was developed XML working group in 1996. XML is used when transferring data from the database to the client, and in designing the visual aspect of Android applications. When data is sent from the database, it is sent using XML. This allows the data to be processed by any programming language the same way, since the data is always sent using XML. As mentioned, XML is also used to design the user interface of Android applications. This means that all the visual aspects such as, the layout of the page, the position of all button and text fields, as well as the color of anything on the page is specified using XML. Since XML is human-legible, it makes the process of designing a page in the app.

XML Code of Encryption and Decryption.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/ConstraintLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    tools:context="Main.">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="centerCrop"
        android:src="@drawable/grjml"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/Swtich"
        android:layout_width="300dp"
        android:layout_height="90dp"
        android:background="@drawable/buttonshape"
        android:onClick="encryptionButtonClick"
        android:text="@string/advanced_encryption_standard"
        android:textColor="#000000"
        android:textSize="20sp"
        android:layout_marginTop="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"



        app:layout_constraintTop_toTopOf="parent" />
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_marginTop="20dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/Key">

        <Button
            android:id="@+id/Encrypt_Buuton"
            android:layout_width="111dp"
            android:layout_height="60dp"
            android:background="@drawable/buttonshape"
            android:onClick="encryptionButtonClick"
            android:text="@string/encrypt"
            android:textSize="19sp"
            android:layout_marginRight="10dp"/>

        <Button
            android:id="@+id/Decrypt_Buuton"
            android:layout_width="111dp"
            android:layout_height="60dp"
            android:background="@drawable/buttonshape"
            android:onClick="encryptionButtonClick"
            android:text="@string/decrypt"
            android:textSize="19sp"
            android:layout_marginLeft="10dp"/>

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

4.5 Back End Technology

4.5.1 Java

Java is an object-oriented programming language created by James Gosling, Mike Sheridan, and Patrick Naughton in 1991. In the paper The Java Language Specification Java SE 8 Edition JamesGosling states, “Java programming language is a general purpose, concurrent, class based, object- oriented language. It is designed to be simple enough that many programmers can achieve fluency in the language. The Java programming language is related to C and C++ but is organized rather differently, with a number of aspects of C and C++ omitted and a few ideas from other languages included. It is intended to be a production language, not a research language.

Java is a very flexible programming language which is used to create many different types of applications for many different operating systems. This is possible because Java can be run on any operating system, as long a s the Java Runtime Environment is available. The application created for Android devices must be coded using Java programming language. This allows these apps to work on variety of different devices, no matter the company that has manufactured the device.

Java Class used in this project:

1. Main Activity
2. Encryption
3. Decryption
4. Hashing

Java Code of Encryption and Decryption

```
package Encryption.Algorithms;

import android.util.Base64;
import android.util.Log;

import java.security.MessageDigest;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class AES {
public String AESencrypt ( byte[] key, byte[] clear) throws Exception {
```

```
MessageDigest md = MessageDigest.getInstance("md5");
        byte[] digestOfPassword = md.digest(key);

        SecretKeySpec skeySpec = new SecretKeySpec(digestOfPassword, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        byte[] encrypted = cipher.doFinal(clear);
        return Base64.encodeToString(encrypted, Base64.DEFAULT);

    }
    public String AESdecrypt (String key,byte[] encrypted) throws Exception {
        MessageDigest md = MessageDigest.getInstance("md5");
        byte[] digestOfPassword = md.digest(key.getBytes("UTF-16LE"));

        SecretKeySpec skeySpec = new SecretKeySpec(digestOfPassword, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding");
        cipher.init(Cipher.DECRYPT_MODE, skeySpec);
        byte[] decrypted = cipher.doFinal(encrypted);
        return new String(decrypted, "UTF-16LE");
    }
} package Hash;

import android.content.Context;
import android.os.Bundle;
import android.text.InputType;
import android.util.Base64;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.Fragment;
import com.example.Algorithms.R;
```

```
public void hash(View view) throws Exception {
    if (Textfield_Text.length() == 0) {
        Toast.makeText(view.getContext(), "Enter a message to Hash", Toast.LENGTH_SHORT).show();
        return;
    }
    message = String.valueOf(Textfield_Text.getText());
    salt = String.valueOf(Textfield_salt.getText());
    String Algorithm = String.valueOf(Switch.getText());
    String answer="";
    switch (Algorithm) {
        case "MD5":
            answer=hashText("MD5",salt,message);
            Answer.setText(answer);
            break;
        case "SHA-256":
            answer=hashText("SHA-256",salt,message);
            Answer.setText(answer);
            break;
        case "SHA-512":
            answer=hashText("SHA-512",salt,message);
            Answer.setText(answer);
            break;
    }
}


public void switchAlgho(View view) {
    reset(null);
    String SwitchValue = Switch.getText().toString();
    switch (SwitchValue) {
        case "MD5":
            Switch.setText("SHA-256");
            break;
        case "SHA-256":
            Switch.setText("SHA-512");
            break;
        case "SHA-512":
            Switch.setText("MD5");
            break;
    }
}
```

CHAPTER 5

SYSTEM TESTING

Testing is a process of executing a program with the intent of finding an error. It is the crucial element of software quality assurance and presents ultimate review of specification, design and coding.System testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptancetesting.

5.1 Testing Principles

- All tests should be traceable to end user requirements.
- Tests should be planned long before testing begins and Exhaustive testing is not possible.
- Testing should begin on a small scale and progress towards testing in large.
- To be most effective testing should be conducted by a independent third party.

5.2 Test Cases

Table 5.1:Unit Testing for Encryption and Decryption

Test Case ID	Test Case Description	Expected Output	Actual Output	Result
1	Encrypt with AES algorithm.	0X69D52F8C	0X69D52F8C	Pass
2	Decrypt with AES algorithm.	Hello World!	Hello World!	Pass
3	Encrypt with Triple DES algorithm.	0X3F6B1AA9	0X3F6B1AA9	Pass
4	Decrypt with Triple DES algorithm.	Hello World!	Hello World!	Pass
5	Encrypt with AES algorithm (Invalid Key).	Key does not match.	Key matches.	Fail
6	Decrypt with AES algorithm (Invalid Key).	Key does not match.	Key matches.	Fail

Table.5.1 Shows the test case description briefly explains the purpose of the test case. The actual output column contains the output obtained from executing the encryption or decryption process using the specific algorithm. The expected output column holds the expected output for each test case. Finally, the result column indicates whether the actual output matches the expected output (Pass) or not (Fail).

Table 5.2:Unit Testing for Hashing using Various Techniques

Test Case ID	Test Case Description	Expected Output	Actual Output	Result
1	Hash with MD5 algorithm.	5eb63bbbe01eeed0	5eb63bbbe01eeed0	Pass
2	Hash with SHA-1 algorithm.	a94a8fe5ccb19ba6	a94a8fe5ccb19	Pass
3	Hash with SHA-256 algorithm.	ba7816bf8f01cfea	ba7816bf8f01c	Pass
4	Hash with SHA-512 algorithm.	2ef7bde608ce5404	2ef7bde608ce540	Pass
5	Hash with MD5 algorithm (Invalid Key).	Key does not match.	Key matches.	Fail
6	Hash with SHA-256 algorithm (Invalid Key).	Key does not match.	Key matches.	Fail

Table.5.2 Shows the test case ID, test case description, and result columns have the same meaning as in the previous example. The actual output column contains the hash output obtained from executing the hashing process using the specific algorithm. The expected output column holds the expected hash output for each test case. The result column indicates whether the actual output matches the expected output (Pass) or not (Fail).

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The Android encryption and decryption system using various algorithms is a secure and flexible solution for protecting data on Android devices. Users can choose from multiple encryption algorithms, such as AES, Triple DES, and Hashing, based on their security needs. The system's architecture ensures a smooth and secure encryption process.

Feasibility studies confirm the system's viability. The economic feasibility study shows potential financial viability, while the technical feasibility study confirms compatibility and efficiency. The legal feasibility study ensures compliance with data protection and cryptographic regulations. The social feasibility study addresses user perceptions and ethical considerations.

Overall, the Android encryption and decryption system provides a comprehensive solution, enhancing mobile data security and safeguarding confidential information effectively.

6.2 Future Enhancement

In order to further enhance the Android encryption and decryption system using various algorithms, there are several future enhancements to consider. Firstly, integrating post-quantum cryptography algorithms can ensure long-term data security against emerging quantum computing threats. Secondly, implementing hardware acceleration techniques, such as Trusted Execution Environments (TEEs) or Secure Elements (SEs), can improve system performance and efficiency.

Enhancing key management with secure key exchange protocols and rotation mechanisms can strengthen data security. Additionally, supporting cloud integration would enable seamless and secure data transfer and storage. User-friendly authentication mechanisms, like biometric or multi-factor authentication, can enhance usability without compromising security. Continuous security audits and updates, including the latest patches and cryptographic standards, are essential to address vulnerabilities and emerging threats. These future enhancements will ensure the Android encryption and decryption system remains at the forefront of data security, offering advanced features, improved performance, and enhanced user experience.

BIBLIOGRAPHY

[1] Mihir Bellare and Phillip Rogaway, "Hash Functions: Theory, Attacks, and Applications", PP: 56-58, 2004.

[2] https://www.geeksforgeeks.org/encryption-and-decryption-application-in-android-using-caesar-cipher-algorithm.

[3] Jean-Philippe Aumasson, Luca Henzen and Willi Meier, "The Hash Function BLAKE", PP: 78-81, 2013.

APPENDIX

APPENDIX A: SNAPSHOTS

A.1: ENCRYPTION

A.1 Shows the Encryption of plain text with the key.

A.2: DECRYPTION

A.2 Shows the copied encrypted text in the plain text to decrypted with same key used in the encryption.

A.3: HASHING (MD5)

A.3 Shows the Hashing using the MD5 technique with the 128 bits.

A.4: HASHING (SHA-256)

A.4 Shows the Hashing using the SHA technique with the 256 bits.

A.5: HASHING (SHA-512)

A.5 Shows the Hashing using the SHA technique with the 512 bits.