A
Mini-Project Report on

Android Text Encryption Application

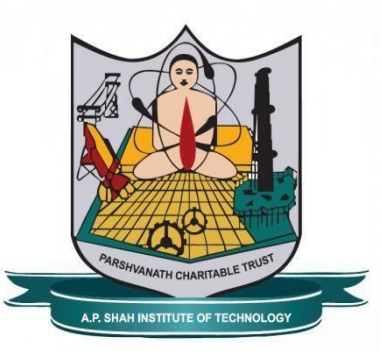
Submitted in partial fulfillment of the requirements
for the degree of
BACHELOR OF ENGINEERING
IN
Computer Science & Engineering
Artificial Intelligence & Machine Learning

By

Ritik Pandey (22106054)
Om Panchal (22106025)
Krishna Mishra (22106131)
Sumedh Gadpayle (22106076)

Under the guidance of

Prof. Monali Korde



Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)
A. P. Shah Institute of Technology
G. B. Road, Kasarvadavali, Thane (W)-400615
University Of Mumbai
2023-2024

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "**Android Text Encryption Application"** is a bonafide work of Ritik Pandey (22106054), Om Panchal (22106025), Krishna Mishra (22106131) , Sumedh Gadpayle (22106076) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

Prof. Monali Korde	Dr. Jaya Gupta
Mini Project Guide	Head of Department

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval

This Mini project report entitled "**Android Text Encryption Application"** by **Ritik Pandey, Om Panchal, Krishna Mishra, Sumedh Gadpayle** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2023-24***.

External Examiner: _____

Internal Examiner: _____

Place: APSIT, Thane
Date:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission hasnot been taken when needed.

Ritik Pandey	**Om Panchal**	**Krishna Mishra**	**Sumedh Gadpayle**
(22106054)	**(22106025)**	**(22106131)**	**(22106076)**

ABSTRACT

The Android Text Encryption App is designed to address the vulnerability of unsecured communication over social media platforms. With messages being transmitted without encryption, they are susceptible to interception by unauthorized individuals. To ensure secure communication, the app employs a selected encryption algorithm and key to encrypt plain text, generating a cipher text that represents the secured message. Upon receipt, the receiver can use the same algorithm to decrypt the message. Implemented using Android Studio and Java, the app is compatible with Android devices, ensuring widespread accessibility. By offering a secure method for communication, the Android Text Encryption App enhances user privacy and data security in digital interactions.

Index

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

In today's digital age, the need for secure communication has become more critical than ever. With the widespread use of social media platforms for communication, the vulnerability of unsecured messages being intercepted by unauthorized individuals has increased. To address this concern, the Android Text Encryption App provides a solution by offering a secure method for encrypting and decrypting messages transmitted over social media platforms. By leveraging encryption algorithms and keys, the app ensures that messages are protected from intruders, enhancing user privacy and data security.

The encryption process of the app involves using a selected algorithm and key to encrypt plain text, generating a cipher text that represents the secured message. This cipher text can only be decrypted by the intended receiver using the same algorithm and key, ensuring that the message remains confidential during transmission. By providing this encryption functionality, the app empowers users to communicate securely over social media platforms, mitigating the risk of unauthorized access to their messages.

Implemented using Android Studio and Java, the app is designed to be user-friendly and compatible with a wide range of Android devices. This compatibility ensures that users can easily access and utilize the app, regardless of their device specifications. Additionally, the app's intuitive interface makes it easy for users to encrypt and decrypt messages with minimal effort, enhancing the overall user experience.

Overall, the Android Text Encryption App offers a practical and effective solution to the security vulnerabilities associated with unsecured communication over social media platforms. By providing a secure method for encrypting and decrypting messages, the app enhances user privacy and data security, ultimately empowering users to communicate confidently in the digital age.

CHAPTER 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2.1-HISTORY

The history of text encryption is as old as the art of communication itself. Early civilizations, including the Egyptians, Greeks, and Romans, used simple encryption techniques to conceal messages. One of the earliest and simplest forms of encryption was the Caesar cipher, named after Julius Caesar, who is said to have used it to protect military messages. The Caesar cipher works by shifting each letter in the plaintext message by a fixed number of positions down the alphabet. While rudimentary, this method laid the foundation for more sophisticated encryption techniques.

The Renaissance period saw the development of more complex encryption methods, including the Vigenère cipher, which used a keyword to encrypt messages. This cipher was considered unbreakable for many centuries until Charles Babbage and Friedrich Kasiski independently developed methods to crack it in the mid-19th century.

One of the most significant advancements in text encryption came during World War II with the development of the Enigma machine by the Germans. The Enigma machine used a series of rotors to encrypt messages, creating a complex and constantly changing cipher. The Allies initially struggled to decrypt Enigma-encrypted messages until the efforts of cryptanalysts such as Alan Turing and the invention of the bombe machine led to the breaking of the Enigma code, a crucial turning point in the war.

In the modern era, the development of digital encryption algorithms has revolutionized the field of text encryption. The Data Encryption Standard (DES), developed in the 1970s, was one of the first widely used encryption standards. However, advances in computing power led to the development of the Advanced Encryption Standard (AES) in the early 2000s, which is now considered the gold standard for encryption.

The rise of the internet and digital communication has highlighted the importance of secure communication methods. With cyberattacks and data breaches becoming increasingly common,

encryption has become essential for protecting sensitive information. Modern encryption techniques use complex mathematical algorithms to ensure the security and privacy of digital communication.

Overall, the history of text encryption is a testament to human ingenuity and the constant evolution of encryption techniques in response to the changing landscape of communication and security threats. From simple ciphers to sophisticated encryption algorithms, the history of text encryption is a fascinating journey through the development of secure communication methods.

2.2-LITERATURE REVIEW

Text encryption plays a crucial role in securing digital communication, especially on mobile devices like Android smartphones. This literature review examines the current research and developments in the field of Android text encryption applications.

1. **Abdullah, Ako Muhamad. "Advanced encryption standard (AES) algorithm to encrypt and decrypt data."** ***Cryptography and Network Security*** **16.1 (2017):**
   Advanced Encryption Standard (AES) algorithm is one on the most common and widely symmetric block cipher algorithm used in worldwide. This algorithm has an own particular structure to encrypt and decrypt sensitive data and is applied in hardware and software all over the world. It is extremely difficult to hackers to get the real data when encrypting by AES algorithm. Till date is not any evidence to crake this algorithm. AES has the ability to deal with three different key sizes such as AES 128, 192 and 256 bit and each of these ciphers has 128-bit block size. This paper will provide an overview of AES algorithm and explain several crucial features of this algorithm in details and demonstration some previous researches that have done on it with comparing to other algorithms such as DES, 3DES, Blowfish etc.

2. **Tayde, Suchita, and Seema Siledar. "File encryption, decryption using AES algorithm in android phone."** ***International Journel of Advanced Research in computer science and software engineering*** **5.5 (2015).**

   Smartphones and tablets offer extensive functionality but face security concerns. Encryption, vital for data security, is often limited by small key sizes or slower processes in algorithms like DES and 3DES. The Advanced Encryption Standard (AES) addresses these issues, providing both security and speed, making it ideal for mobile devices. This Android app allows users to encrypt various file types using AES for enhanced data security.

3. **Deshpande, Ashwini M., Mangesh S. Deshpande, and Devendra N. Kayatanavar. "FPGA implementation of AES encryption and decryption." 2009 international conference on control, automation, communication and energy conservation. IEEE, 2009.**
   This paper explores implementing the AES algorithm on Field Programmable Gate Arrays (FPGAs) using VHDL. It leverages ModelSim SE PLUS 5.7g for simulation and Xilinx ISE 8.2i suite for synthesis and implementation. The approach aims to optimize hardware usage while maintaining high data rates for encryption and decryption. By integrating encrypter and decrypter modules, the architecture achieves low complexity, making it suitable for hardware-critical applications like smart cards and mobile phones.

In conclusion, the research papers reviewed highlight the importance of encryption algorithms, key management, user interface design, performance analysis, and security vulnerabilities in the development of Android text encryption applications. Future research in this field should focus on addressing security vulnerabilities, improving encryption efficiency, and enhancing user experience to promote the widespread adoption of secure text encryption practices on Android devices.

CHAPTER 3

Problem Statement

3. Problem Statement

The Android Text Encryption Application revolves around the inherent vulnerability of text communication on Android devices, especially over social media platforms, where messages are often transmitted without encryption. This lack of encryption makes the messages susceptible to interception by unauthorized individuals, compromising user privacy and data security. The aim of the Android Text Encryption Application is to address this issue by providing a secure method for encrypting and decrypting text messages on Android devices, ensuring that sensitive information remains confidential during transmission.

Key Challenges:

Key Challenges in Developing an Android Text Encryption Application:

- **Algorithm Selection:** Choosing the right encryption algorithm is crucial, as it impacts both security and performance. The algorithm must be strong enough to resist attacks while being efficient enough to run on resource-constrained mobile devices like smartphones.

- **Key Management:** Managing encryption keys securely is a major challenge. Keys must be generated, stored, and exchanged securely to prevent unauthorized access to encrypted data. Key management becomes even more complex in scenarios involving multiple users or devices.

- **User Interface Design:** Designing a user-friendly interface for the encryption application is essential for ensuring user adoption and adherence to encryption practices. The interface should be intuitive and easy to use, even for users with limited technical knowledge.

- **Performance Optimization:** Encryption and decryption operations can be resource-intensive, especially on mobile devices. Optimizing the performance of the encryption application to minimize battery consumption and processing overhead is a significant challenge.

- **Compatibility and Interoperability:** Ensuring that the encryption application is compatible with a wide range of Android devices and can seamlessly integrate with other applications and services is essential for widespread adoption and usability.

- **Security Vulnerabilities:** Identifying and mitigating security vulnerabilities in the encryption application is critical. Common vulnerabilities include implementation flaws, weak key management practices, and susceptibility to attacks such as brute-force attacks or side-channel attacks.

- **Regulatory Compliance:** Adhering to relevant regulations and standards related to data protection and encryption, such as GDPR or HIPAA, adds complexity to the development and deployment of the encryption application.

CHAPTER 4

Experimental Setup

4. Experimental Setup

4.1Software Setup

- Android Studio:
    - Android Studio is the official Integrated Development Environment (IDE) for Android app development. It provides tools for building, testing, and debugging Android applications.
- Java Development Kit (JDK):
    - Android apps are primarily developed using the Java programming language. Install the latest version of JDK compatible with Android Studio.
- Android SDK:
    - Android Software Development Kit (SDK) provides necessary libraries, tools, and APIs for developing Android applications.
    - Ensure to install SDK components required for targeting the desired Android version (e.g., Android 9 or above).

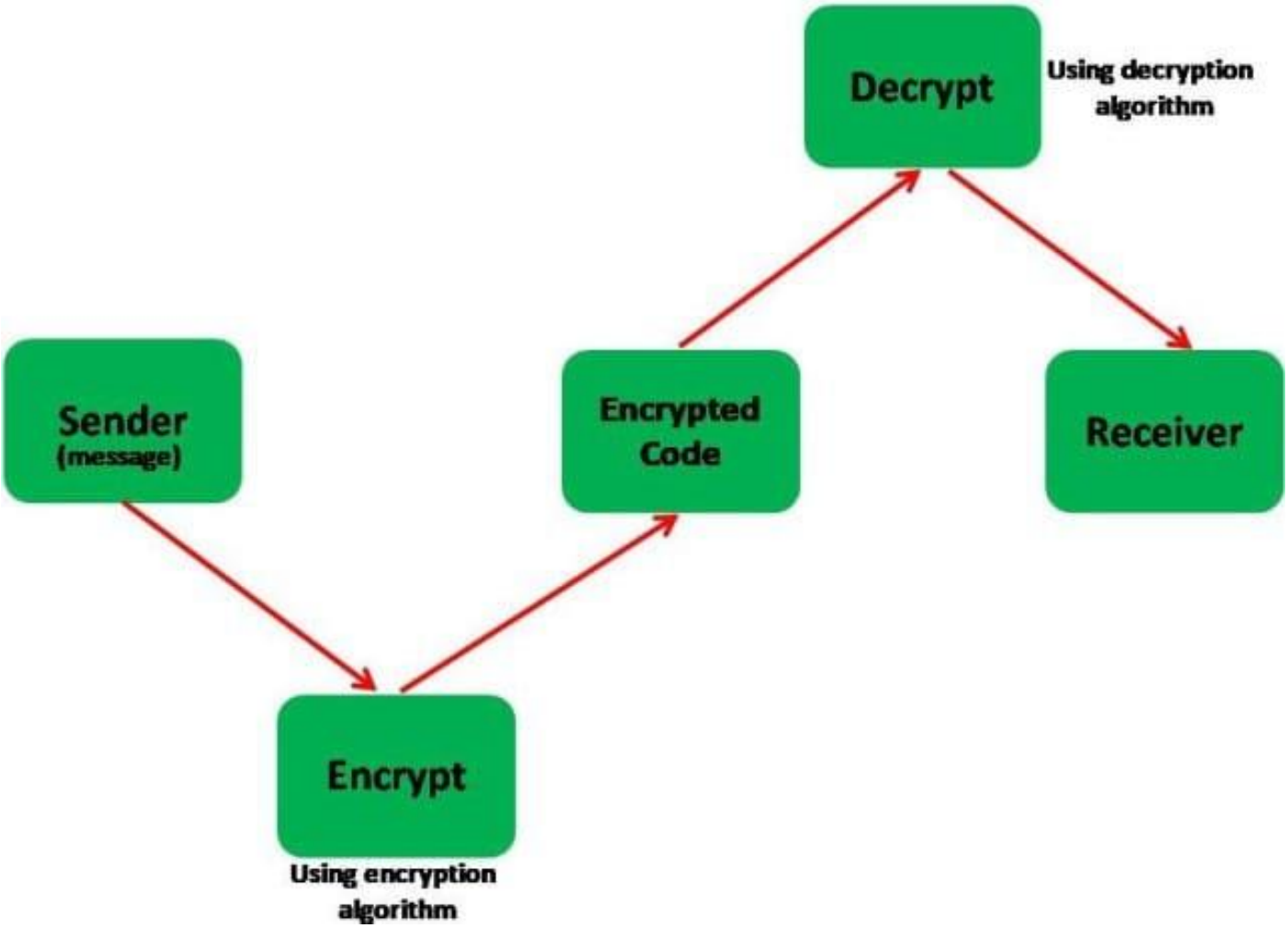
4.2hardware setup:

- Android Device:
    - A compatible Android device running Android OS version 9 or above.
    - This device will serve as the platform for testing and running the Android Text Encryption Application.
- Computer:
    - A modern computer capable of running Android Studio and the Android SDK.
        - Recommended specifications:
            - Processor: Intel Core i5 or higher
            - RAM: 8GB or higher
            - Storage: SSD recommended for faster build times
            - Operating System: Compatible with Windows, macOS, or Linux.

CHAPTER 5

Proposed System & Implementation

5.1 Block diagram of proposed system



5.1 Block Diagram

5.2 Description of block diagram:

- **Sender:**

  - The sender initiates the process by composing a message that they want to send securely.

  - This message needs to be encrypted before transmission to ensure its confidentiality.

- **Encrypt (Using Encryption Algorithm):**

  - The message undergoes encryption using an encryption algorithm.

  - This process scrambles the message into an unreadable format, making it secure from unauthorized

- **Cipher Text or Encrypted Code Is Generated:**

  - After encryption, the message is transformed into ciphertext or encrypted code.

  - Ciphertext appears as random and meaningless characters, ensuring the original message is concealed.

- **Decrypt Text Using Decryption Algorithm:**

  - The encrypted message is transmitted to the receiver.

  - Upon reception, the receiver needs to decrypt the ciphertext to recover the original message.

  - Decryption involves using a decryption algorithm, typically the inverse of the encryption algorithm.

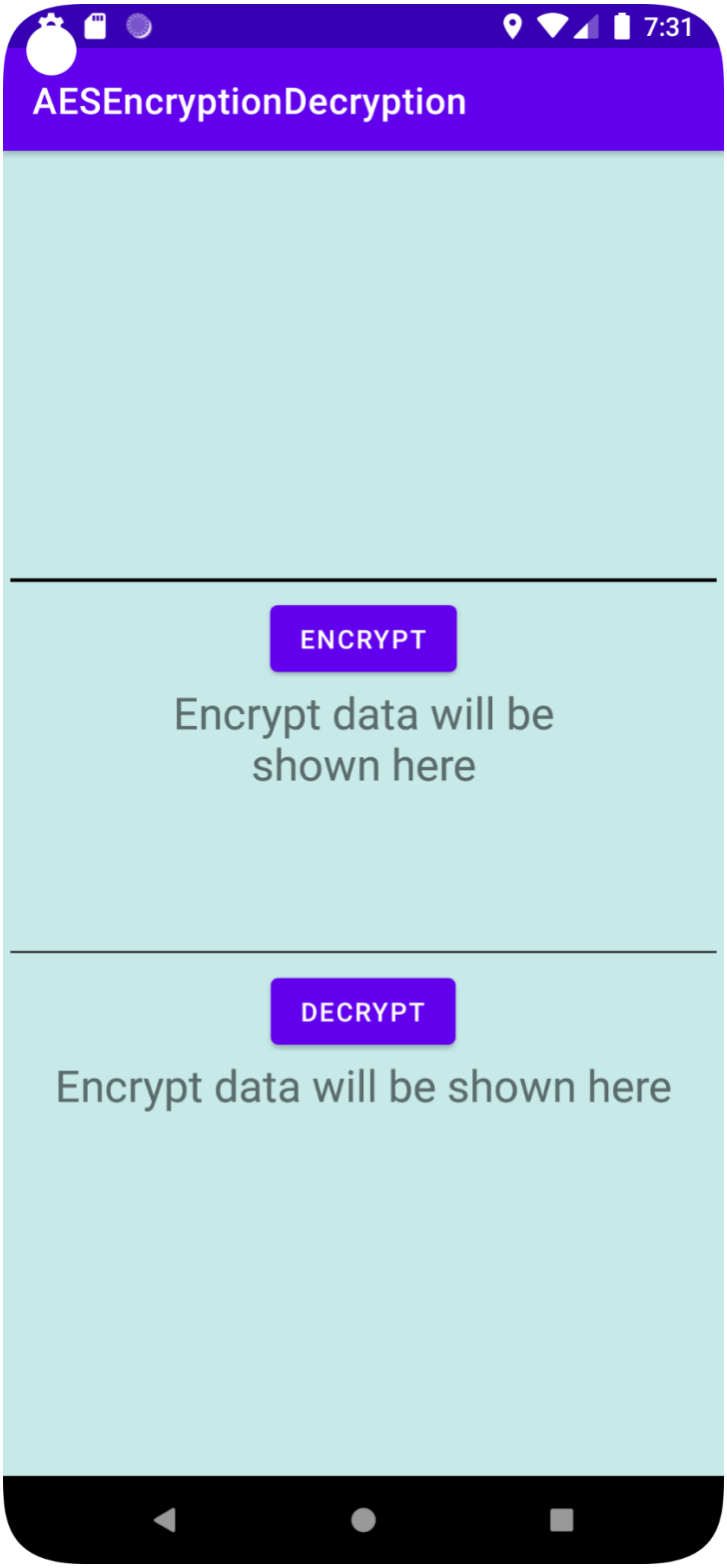
  - The decryption process unscrambles the ciphertext, revealing the original plaintext message.

- **Receiver:**

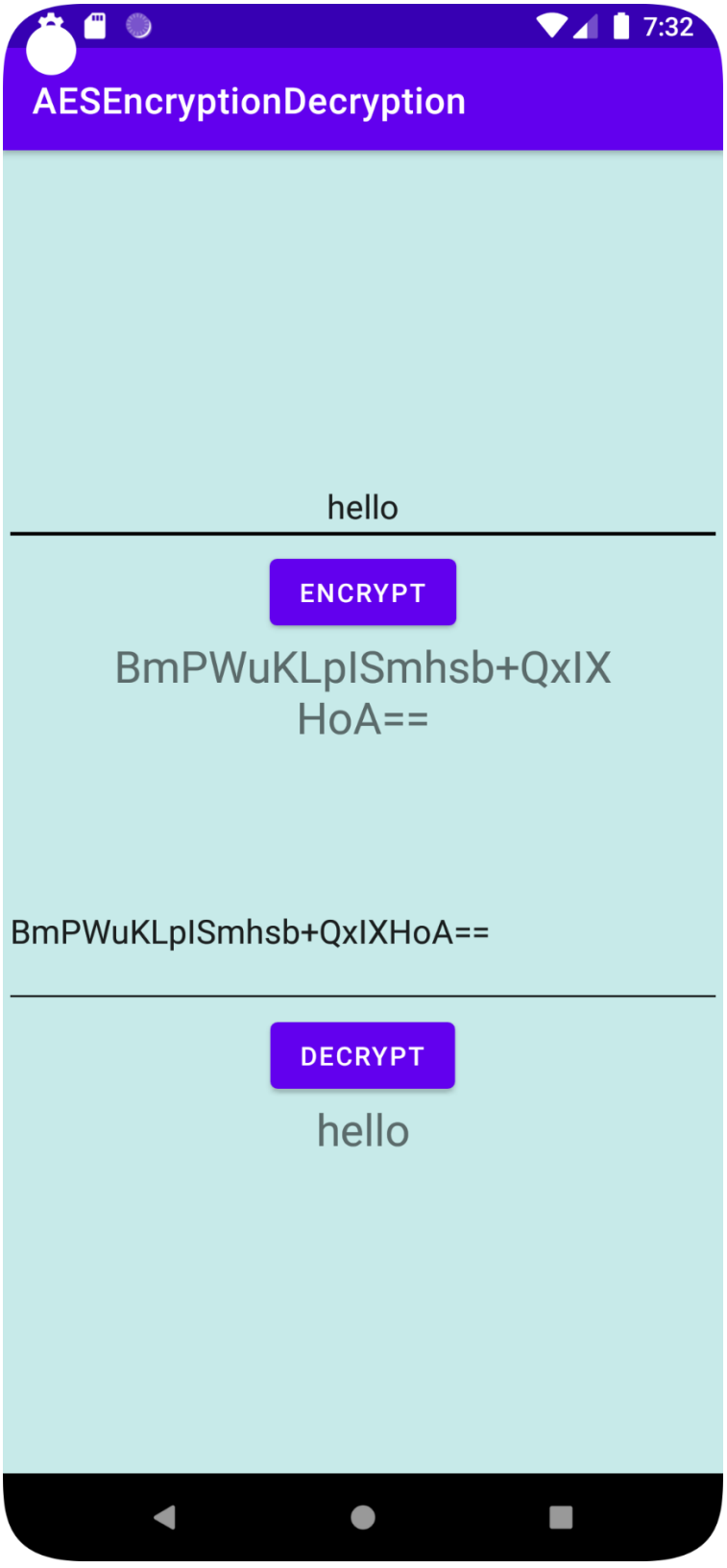
  - The receiver receives the decrypted message.

  - They can now read the message in its original form, ensuring secure communication between the sender and receiver.

5.3 Implementation



5.3 (a) Home Page



5.3(b) Text Encryption

CHAPTER 6

Conclusion

6. Conclusion

The Android Text Encryption application employs the AES (Advanced Encryption Standard) algorithm to secure text data. AES is a symmetric encryption standard renowned for its strength and efficiency. This application integrates AES to encrypt and decrypt text, ensuring that sensitive information remains confidential and secure.

Encryption is a process that converts plaintext into ciphertext, making it unreadable without the correct decryption key. The AES algorithm encrypts data in blocks of 128 bits, using keys of varying lengths (128, 192, or 256 bits). This flexibility allows the application to adjust the level of security based on the chosen key size.

When encrypting text, the application generates a random encryption key of the specified length. It then uses this key, along with the AES algorithm, to encrypt the text. The encrypted text, or ciphertext, is what is transmitted or stored. Decryption reverses this process, using the same key to convert the ciphertext back to plaintext.

The use of AES ensures that the Android Text Encryption application provides a high level of security for text data. The strength of AES lies in its ability to withstand attacks and its widespread adoption in various security-sensitive applications. By utilizing AES, the application offers users a reliable and robust method for encrypting and decrypting text, ensuring the confidentiality and integrity of their data.

References

- Abdullah, Ako Muhamad. "Advanced encryption standard (AES) algorithm to encrypt and decrypt data." *Cryptography and Network Security* 16.1 (2017):
- Tayde, Suchita, and Seema Siledar. "File encryption, decryption using AES algorithm in android phone." *International Journel of Advanced Research in computer science and software engineering* 5.5 (2015).
- https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- "Android Developer Documentation." : https://developer.android.com/docs