India Language Forum (ILF)

A Project Report Submitted

to

MANIPAL ACADEMY OF HIGHER EDUCATION

BACHELOR OF TECHNOLOGY

in

Information Technology

Submitted by

Kabir Bajaj 225811338

For

The course “Information Security”

MANIPAL INSTITUTE OF TECHNOLOGY

BENGALURU

(A constituent unit of MAHE, Manipal)

DEPARTMENT OF INFORMATION TECHNOLOGY

October 2024

DECLARATION

We hereby declare that the project work entitled **Cryptchat** is original and has been carried out at for the course “Information Security”, under the guidance of **Dr. Abhijeet Das**. We further declare that the work reported in this document has not been submitted either in part or full to any other Institute/University for the award of any other degree.

Place: Bengaluru
Date: 2 – 11 - 2024

ACKNOWLEDGMENTS

We would like to extend my sincere gratitude to everyone who has supported and guided me throughout the development of this project.

First and foremost, we are deeply grateful to my project supervisor, Dr. Abhijeet Das, for their insightful guidance, continuous support, and encouragement throughout this project. Their expertise and constructive feedback were invaluable in helping me shape and refine my ideas, and their unwavering patience and encouragement were instrumental in completing this project successfully.

We would also like to express my appreciation the Department of Information Technology at Manipal Institute of Technology for providing the resources and a conducive environment for research and development. The facilities and support provided by the institution enabled me to approach and execute this project effectively.

Our heartfelt thanks go out to my family and friends, whose understanding, encouragement, and belief in me provided the strength and motivation needed to persevere. We are also thankful to my colleagues for their suggestions, support, and teamwork, which added value to this work.

Lastly, we would like to acknowledge the invaluable contributions of all the authors and researchers whose work served as a foundation for this project. Their research and insights were instrumental in guiding me through the literature review and theoretical framework. Thank you to everyone who played a part in the completion of this project.

2. Abstract

This project report presents *CryptChat*, an Android-based secure chat application designed for end-to-end encrypted messaging between a client and server. CryptChat employs the Diffie-Hellman key exchange to securely generate a common secret key, used to encrypt and decrypt chat messages using AES encryption. The system architecture, including both client and server, is designed to establish secure communication channels that ensure message confidentiality and integrity. BASE64 encoding is used for message transmission, ensuring data compatibility over the network socket. This report explores the system design, implementation, testing, and security analysis, showcasing *CryptChat* as a scalable solution for secure communication on Android.

3. Table of Contents

4. Introduction

13. 4.1 Overview

With the growing demand for secure communication, end-to-end encrypted chat applications have become essential to ensure data privacy. CryptChat is a secure chat application that utilizes the Diffie-Hellman key exchange for secure key generation and AES encryption for message confidentiality. This approach prevents unauthorized access to chat data by establishing a robust encryption system between the Android client and a server.

14. 4.2 Background and Motivation

End-to-end encryption has become standard in secure messaging applications. This project aims to explore and implement key cryptographic techniques to provide users with a secure chat experience, particularly focusing on confidentiality and key security using Diffie-Hellman and AES algorithms.

5. Objectives

- To design and implement a secure chat application on Android with client-server architecture.
- To utilize Diffie-Hellman key exchange for secure key generation without directly transmitting the key.
- To use AES encryption for encrypting messages with a shared secret key.
- To encode messages with BASE64 for network transmission, ensuring data integrity.

6. System Architecture

The *CryptChat* architecture is designed in a client-server model with the following key components:

1. **Client-Side (Android App):** Initiates the connection with the server, performs key exchange, and sends/receives encrypted messages.
2. **Server-Side:** Listens for client connections, participates in the Diffie-Hellman key exchange, and handles encrypted message transactions.

The client and server independently generate key pairs, exchange public keys, compute the shared secret key, and use it for symmetric encryption (AES) of messages.

7. Methodology

15. 7.1 Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange allows two parties to securely generate a common secret key over a public network:

- **Public Key Exchange:** Both client and server generate public/private key pairs and exchange public keys.
- **Secret Key Generation:** Each party uses their private key and the other's public key to derive a common secret key.
- **Security Benefit:** The common secret key is never transmitted, reducing interception risk.

16. 7.2 AES Encryption

AES (Advanced Encryption Standard) is a symmetric encryption algorithm used in *CryptChat* to secure message content:

- **Encryption/Decryption Process:** Messages are encrypted on the sender's side with the shared secret key and decrypted on the receiver's side using the same key.
- **Symmetric Key:** The same key derived from Diffie-Hellman is used for both encryption and decryption, ensuring data privacy.

17. 7.3 BASE64 Encoding

BASE64 encoding converts binary data into ASCII text, making it compatible for transmission over network sockets:

- **Encoding:** Messages are encoded before sending, ensuring reliable transport.
- **Decoding:** Encoded messages are decoded on the receiving end, preparing them for decryption.

8. Implementation

18. 8.1 System Requirements

- **Android Studio (for client development)**
- **Server Environment:** Node.js or Python for server-side scripting
- **Java Cryptography Extension (JCE) Library:** For cryptographic algorithms

19. 8.2 Client-Server Communication Flow

1. **Initialization:** Client and server establish a connection.
2. **Key Exchange:** Both generate a key pair and exchange public keys.
3. **Secret Key Calculation:** Both compute the shared secret key using Diffie-Hellman.
4. **Message Exchange:** Messages are encrypted, BASE64 encoded, and sent to the server.

20. 8.3 Key Generation and Exchange

- **Client and Server Key Pair Generation:** Both use a key generator to create public and private keys.
- **Public Key Exchange:** The public keys are exchanged over the network.
- **Shared Secret Key Derivation:** Both derive the same secret key independently.

21. 8.4 Message Encryption and Decryption

1. **Encryption:** Messages are encrypted with AES using the shared secret key.
2. **BASE64 Encoding:** Encrypted messages are encoded in BASE64 for network transmission.
3. **Decryption:** The server decodes and decrypts the message, and vice versa.

9. Security Analysis

- **Confidentiality:** Messages are encrypted end-to-end, preventing unauthorized access.
- **Data Integrity:** BASE64 encoding ensures message consistency over the network.
- **Key Security:** Diffie-Hellman generates a shared key without direct transmission, securing key exchange.

10. Testing and Evaluation

CryptChat was rigorously tested to evaluate the following:

- **Encryption/Decryption Functionality:** Verified that only intended recipients could decrypt messages.
- **Key Exchange Verification:** Ensured Diffie-Hellman successfully generated matching keys on both sides.
- **Transmission Accuracy:** BASE64 encoding maintained message integrity over network transmission.

11. Challenges and Solutions

- **Challenge:** Managing encryption compatibility across different devices.
    - **Solution:** Ensured both client and server adhered to the same AES encryption standards.
- **Challenge:** Securely exchanging public keys in a network environment.
    - **Solution:** Used an SSL layer to protect public key exchanges.

12. Future Enhancements

- **Multi-User Chat Support:** Expanding *CryptChat* to support secure group messaging.
- **User Authentication:** Adding user authentication for additional security.

- **Push Notifications:** Implementing encrypted notifications to alert users of new messages.

13. Conclusion

CryptChat successfully demonstrates a secure chat system using Diffie-Hellman and AES encryption, protecting messages from interception and unauthorized access. Through robust encryption and secure key exchange, *CryptChat* delivers a reliable solution for confidential messaging on Android.