JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Secure Communication Android App using AES Algorithm - CrypTalk

Alisha Jamadar
Computer Science & Engineering
PCET's Nutan College of Engineering & Research
Talegaon Dabhade, Pune, India
alishajamadar1500@gmail.com

Shubham Hadgal
Computer Science & Engineering
PCET's Nutan College of Engineering & Research
Talegaon Dabhade, Pune, India
shubhamhadgal45@gmail.com

Saurabh Shahapure
Computer Science & Engineering
PCET's Nutan College of Engineering & Research
Talegaon Dabhade, Pune, India
saurabhshahapure@gmail.com

Prof. A.V. Sagare
Computer Science & Engineering
PCET's Nutan College of Engineering & Research
Talegaon Dabhade, Pune, India
jaat.sagare@gmail.com

Abstract- As smartphones, computers, and social media applications on them have become an inseparable part of our daily lives, so have the security and privacy of our data. The data that is stored and transmitted over the internet via many social media apps might mostly include the user's personal and confidential information such as legal name, date of birth, address and even credit card details. With increasing cyber threats and attacks each day in this fast-paced world, there is an urgent and demanding need to secure such data at a very high speed, and efficiently. For this purpose, the AES (Advanced Encryption Standard) can be implemented to encrypt sensitive data to prevent the risk of any form of cyber-attack. Using a unique secret key between a pair of sender and receiver ensures their data is securely transmitted without being accessed or decrypted by any other entity.

Keywords- AES; cryptography; cyber-security, android application, secure key

I. Introduction

A. AES Algorithm

In the domain of cyber security, cryptography is a term that refers to securing the data regardless of its type, form or volume. It involves the practice of applying security rules and algorithms to data in a human-readable format and converting it into an encrypted text which cannot be easily deciphered by humans. This ensures that the data that exponentially gets generated every day from every other source is well protected at every stage of its existence, whether it is being stored, transmitted or received. One such method to achieve data security is by using the AES algorithm, also known as Rijndael.

The AES algorithm developed by Vincent Rijmen and Joan Daemen Belgian was established by NIST-National Institute of Standard and Technology, US. It is a symmetric key algorithm that takes an input as a block of size 128 bits and uses keys of three different lengths – 128, 192 and 256 bits to convert it into a cypher text. AES is included in the ISO/IEC 18033-3 standard, and is the only publicly accessible cypher approved by the U.S. National Security Agency (NSA) for top secret information.

This paper describes the use and implementation of the AES algorithm within a social media and communication android application to achieve the highest possible security of users' data.

B. CrypTalk – A Secure Communication App

CrypTalk is a secure communication android application that provides all the features of a social media platform incorporated with the ultimate security of the AES algorithm. The application is developed in Dart Flutter, and tested most compatible on Android platforms ver. 9 (Pie) and above. The user interface of the app is designed to resemble the most widely used and popular applications in the current date to provide ease of usage and operation to the user. The application is made free to download and install.

The technical aspects of the application include the following details –

- Size: 25 mb
- Version: 1.0
- Platform: Android v9+
- Database: Google Firebase
- Language: Dart - Flutter
- Algorithm: AES algorithm

II. System Architecture

The system architecture of CrypTalk includes 3 main components – the sender, the receiver, and the database. When an authorized user sends a message through the application, they become the sender of the message. Similarly, when another authorized user receives that message, they become the receiver of the message. Before being able to send any message, the sender has to set a secret key for the intended receiver from inside the chat screen. The key can be of a length of 16, 24 or 32 characters, including alphanumeric. This key has to be shared with the intended receiver via other media. Following this, the receiver will set the received key on his/her device inside the chat screen of the respective sender. In this sequence, when the sender sends the message, it is first encrypted using the AES algorithm and the single key is saved on both the parties' devices and stored in the application's server database. When the receiver is available to receive the message, it is retrieved from the database and sent on to the receiver side and decrypted so that it is in a readable format. This process ensures the security of the data when it is in transit from the sender side to the database, while it is saved in the database, and also when it is being sent to the receiver side from the database. It prevents any kind of interception or man-in-the-middle attacks.

The following diagram (Fig. 1) depicts the system architecture of the CrypTalk application.

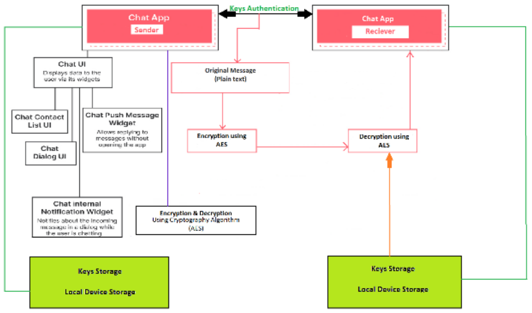

Chat App
Sender
Keys Authentication
Chat App
Reciever
Chat UI
Displays data to the user via its widgets
Original Message (Plain text)
Chat Contact List UI
Chat Push Message Widget
Allows replying to messages without opening the app
Chat Dialog UI
Encryption using AES
Decryption using AES
Chat internal Notification Widget
Notifies about the incoming message in a dialog while the user is chatting
Encryption & Decryption Using Cryptography Algorithm (AES)
Keys Storage
Local Device Storage
Keys Storage
Local Device Storage

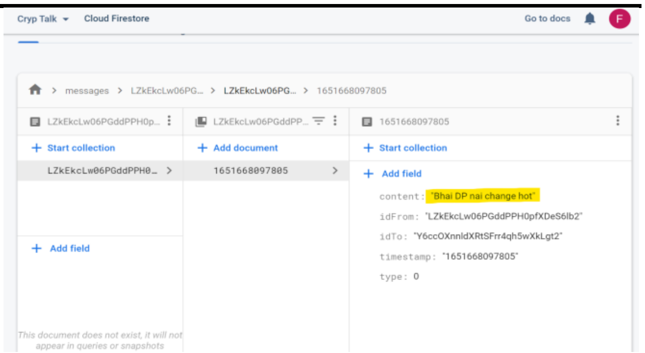
Fig. 1. CrypTalk – System Architecture

III. TOOLS & CODE IMPLEMENTATION

CryptTalk is developed using the Flutter framework of Dart language for the front-end as well as the back-end. Flutter is chosen for its simplicity and reduced code development time. Flutter also provides the potential to go beyond a single platform, which can be very helpful in extending the application from an Android app to a cross-platform app including the iOS and desktop.

The code for the application is first divided into modules such as the user interface, sending, receiving, storing, and finally adding the AES encryption. These code modules are then individually developed and tested in the same fashion.

The code snippets below show the contents of the text message stored in the database before (Fig. 2) and after (Fig. 3) the application of AES encryption. Before applying AES encryption, the message is stored in a simple plain-text format which can be easily read and vulnerable to interception and easy access. However, after the utilization of the encryption technique, the contents of the text message cannot be deciphered even if accidentally exposed to unauthorized entities.



messages > LZkEkcLw06PG... > LZkEkcLw06PG... > 1651668097805

LZkEkcLw06PGddPPH0p... | LZkEkcLw06PGddPP... | 1651668097805

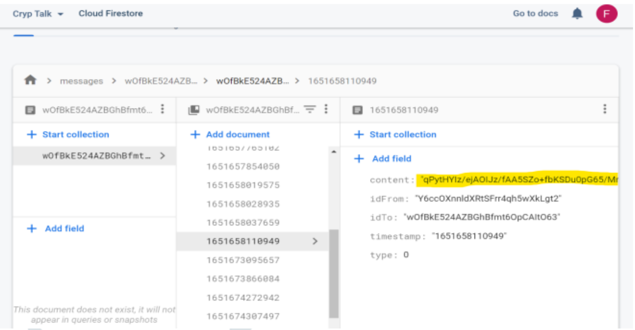
LZkEkcLw06PGddPPH0...

1651668097805

content: "Bhai DP nai change hot"
idFrom: "LZkEkcLw06PGddPPH0pfXDeS6lb2"
idTo: "Y6ccOXnnldXRtSFrr4qh5wXkLgt2"
timestamp: "1651668097805"
type: 0

This document does not exist, it will not appear in queries or snapshots

Fig. 2. Message from chat before encryption



messages > wOfBkE524AZB... > wOfBkE524AZB... > 1651658110949

wOfBkE524AZBGhBfmt6... | wOfBkE524AZBGhBf... | 1651658110949

wOfBkE524AZBGhBfmt...

1651657765102
1651657854050
1651658019575
1651658028935
1651658037659
1651658110949
1651673095657
1651673866084
1651674272942
1651674307497

content: "qPytHYlz/ejAOlJz/fAA5SZo+fbKSDu0pG65/Mr
idFrom: "Y6ccOXnnldXRtSFrr4qh5wXkLgt2"
idTo: "wOfBkE524AZBGhBfmt6OpCAltO63"
timestamp: "1651658110949"
type: 0

This document does not exist, it will not appear in queries or snapshots

Fig. 2. Message from chat after encryption

IV. USER INTERFACE

The user interface is designed and developed with accessibility and ease of use in mind. It has options for light and dark themes which can be changed based on the user's preference. The user interface includes mainly 5 screens –

- Login
- Settings
- Landing screen/Conversation list
- Chat history/chat screen
- Change key

The login screen provides the user with the option to log in or sign in to the app using any of the Google Accounts logged into the device. By default, the app uses the profile picture and the user's name associated with the Google account that is used to sign in. The Settings page allows the user to change or upload their profile picture, name, and their info in brief in the About me section. The landing screen is the page that opens first when you are logged in and open the application. It consists of a list of conversations with people sorted into the latest to earliest format. It also includes a search bar at the top that allows the user to search for a conversation with another person. The chat screen or chat history is the following screen that slides in and shows the chat history with a person when the user clicks on that particular person's conversation. The chat history may include simple text, images, videos, documents, and audio. The chat history will only be visible if the unique key between the sender and receiver is the same, otherwise the contents on the screen will be blocked. Setting the key requires both users to set the same key for each other in their devices.

The sequence of logging in and the usage of the app after the download & installation is as given in the user activity diagram below in Fig. 3.

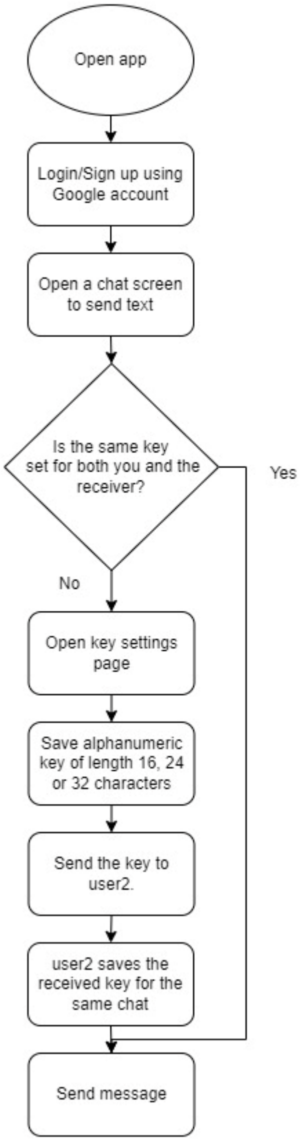

Open app
Login/Sign up using Google account
Open a chat screen to send text
Is the same key set for both you and the receiver?
Yes
No
Open key settings page
Save alphanumeric key of length 16, 24 or 32 characters
Send the key to user2.
user2 saves the received key for the same chat
Send message

Fig. 3. Cryptalk – User Activity Diagram

V. Conclusion & Future Scope

Having compared to existing most popular and widely used social media apps, it is fair to conclude that the CrypTalk application is integrated with the security and privacy provided by the AES algorithm which is mostly not present in most of them. Integrated with the option to add unique security keys to separate individual contacts in the application enforces its security. This ensure that the data sent or received via the application cannot be decoded or decrypted without the key, also getting access to the key without the data itself would prove to be of no use to an interception or hacking activity. Thus comparatively, CrypTalk is far more secure than the variety of currently used applications for communicating and texting purposes.

Being developed in Dart Flutter, it leaves vast scope for further development and integration into other platforms and to also introduce new features within the application exclusive to the CrypTalk. The added feature of posting stories and feeds will be integrated, where using Machine Learning automatic captions and hashtags, automatic face detection and tagging the identified person, can be achieved and implemented. Thus, further enhancing and elevating the entire usage of the application.

References

[1] D. Budiyanto, P. A. W. Putro, “Comparison of Implementation Tiny Encryption Algorithm (TEA) and Advanced Encryption Standard (AES) Algorithm on Android Based Open Source Cryptomator Library,” 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2018.

[2] M. H. Azaim, D. W. Sudiharto, E. Musthofa Jadied, “Design and implementation of encrypted SMS on Android smartphone combining ECDSA - ECDH and AES”, Bali, Indonesia: IEEE, 2016.

[3] A. Aminuddin, “Android Assets Protection Using RSA and AES Cryptography to Prevent App Piracy,” Yogyakarta, Indonesia: IEEE, 2020.

[4] F. B. Setiawan, Magfirawaty, “Securing Data Communication Through MQTT Protocol with AES-256 Encryption Algorithm CBC Mode on ESP32-Based Smart Homes,” Banda Aceh, Indonesia: IEEE, 2021.

[5] M. Gaur, R. Gupta, A. Singh, “Use of AES Algorithm in development of SMS Application on Android Platform,” Noida, India: IEEE, 2021.

[6] T. Adiono, S. Harimurti, B. A. Manangkalangi, W. Adijarto, “Design of smart home mobile application with high security and automatic features,” Yilan, Taiwan: IEEE, 2018.

[7] A. Joshy, K. X. Amitha Baby, S. Padma, K. A. Fasila, “Text to image encryption technique using RGB substitution and AES,” Coimbatore, India: IEEE, 2017.

[8] S. Ariffi, R. Mahmod, R. Rahmat, N. A. Idris, “SMS Encryption Using 3D-AES Block Cipher on Android Message Application,” Kuching, Malaysia: IEEE, 2013.

[9] N. A. Fauziah, E. H. Rachmawanto, D. R. I. M. Setiadi, C. A. Sari, “Design and Implementation of AES and SHA-256 Cryptography for Securing Multimedia File over Android Chat Application,” Yogyakarta, Indonesia: IEEE, 2018.

[10] R. V. Chandrashekhar, J. Visumathi, A. P. Anandaraj, “Advanced Lightweight Encryption Algorithm for Android (IoT) Devices,” Chennai, India: IEEE, 2022.

[11] Abhishek Vichare;Tania Jose;Jagruti Tiwari;Uma Yadav, “Data security using authenticated encryption and decryption algorithm for Android phones,” Greater Noida, India: IEEE, 2017.

[12] M. B. Segoro, P. A. W. Putro, “Implementation of Two Factor Authentication (2FA) and Hybrid Encryption to Reduce the Impact of Account Theft on Android-Based Instant Messaging (IM) Applications,” Depok, Indonesia: IEEE, 2020.

[13] S. Verma, S. K. Pal, S. K. Muttoo, “A new tool for lightweight encryption on android,” Gurgaon, India: IEEE, 2014.

[14] C. A. Lara-Niño, M. Morales-Sandoval, A. Díaz-Pérez, “An evaluation of AES and present ciphers for lightweight cryptography on smartphones,” Cholula, Mexico: IEEE, 2016

[15] R. Talreja, D. Motwani, “SecTrans: Enhacing user privacy on Android Platform,” Vashi, India: IEEE, 2017