Encrypted Chat Application

Mrs Priyanka Singh[1], Atharva Mangaonkar[2], Bhushan Patil[3] , Kishor Patale[4], Sahil Kokate[5]

[1] *Guide Department of Cyber Security, Shah & Anchor Kutchhi Engineering College*
[2] *Student Department of Cyber Security, Shah & Anchor Kutchhi Engineering College*
[3] *Student Department of Cyber Security, Shah & Anchor Kutchhi Engineering College*
[4] *Student Department of Cyber Security, Shah & Anchor Kutchhi Engineering College*
[5] *Student Department of Cyber Security, Shah & Anchor Kutchhi Engineering College*

---***---

Abstract - In today's cyber landscape, the protection of digital communication channels stands as a paramount concern. This abstract dissects the cyber resilience strategies underpinning the development of secure chat applications, with a particular focus on user interface design using Jetpack Compose and encryption integration leveraging Kotlin. The examination commences with an in-depth analysis of the cyber defence mechanisms ingrained within chat interfaces crafted through Jetpack Compose, the cutting-edge toolkit tailored for native Android UI construction. By meticulously adhering to Material Design principles, developers fortify chat interfaces against cyber intrusions, placing a premium on real-time messaging functionalities and multimedia integration to foster user engagement while deterring malicious exploitation.The abstract plunges into the cornerstone role of encryption in fortifying cybersecurity, spotlighting Kotlin's robust encryption capabilities. Through the adept utilization of Kotlin AES encryption algorithms, developers erect an impregnable bastion around user messages, guaranteeing both confidentiality and integrity. Synergizing encryption with Firebase Firestore for fortified data storage and Firebase Authentication for resilient user validation fortifies the application's cyber defense, thereby ensuring user data remains impervious to cyber threats. By amalgamating Jetpack Composes resilient UI design tools with Kotlin's encryption prowess, developers erect a formidable cybersecurity fortress around chat applications. This symbiotic alliance not only safeguards user privacy and data integrity but also reinforces communication channels against the relentless onslaught of cyber adversaries in the ever-evolving cyber landscape

Key Words: Encryption Integration, Secure Chat Application, Cyber Resilience Strategies, Cyber Defense Mechanism.

1. INTRODUCTION

In the realm of digital communication, the advent of encrypted chatting applications marks a pivotal shift towards ensuring robust security and privacy for users. One of the cornerstone features of these applications is the storage of chats in an encrypted format within the database. This proactive measure serves as a formidable barrier against unauthorized access by hackers, offering users a fortified layer of protection for their sensitive conversations and data. Encrypted chatting apps leverage sophisticated encryption algorithms to transform user chats into ciphertext before storing them in the database. This means that even if a hacker manages to gain access to the database, they would encounter encrypted data that is virtually indecipherable without the corresponding decryption keys. This level of encryption renders unauthorized access attempts futile, safeguarding user chats from prying eyes and potential data breaches.

Moreover, the encryption protocols employed by these apps adhere to industry standards and best practices, ensuring the integrity and confidentiality of stored data. By adopting a defence-in-depth approach, where multiple layers of security mechanisms are implemented, encrypted chatting apps mitigate the risk of data compromise and uphold user trust in the platform. This focus on secure database storage underscores the app's commitment to protecting user privacy and maintaining a secure digital environment.

In addition to encryption, encrypted chatting apps often incorporate features such as access controls, audit trails, and regular security audits to further fortify the database against potential vulnerabilities. Access controls limit who can view or modify data within the database, reducing the likelihood of unauthorized access even among internal personnel. Audit trails provide a detailed log of database activities, enabling administrators to monitor for any suspicious behavior or unauthorized access attempts. Regular security audits involve comprehensive assessments of the database infrastructure, identifying and addressing any potential security weaknesses proactively.

By prioritizing secure database storage of user chats, encrypted chatting apps instill confidence among users regarding the confidentiality and integrity of their communications. This strategic approach not only protects sensitive information from external threats but also fosters a culture of trust and transparency within the app's ecosystem. As cyber threats continue to evolve, encrypted chatting apps remain at the forefront of safeguarding user data and ensuring a secure digital communication experience.

Encrypted Chat Application

Chat encryption refers to the process of encoding messages sent between users in a way that makes them unreadable to anyone except the intended recipient. This is typically achieved using cryptographic algorithms that convert plain text messages into cipher text, which can only be deciphered with the appropriate decryption key. Chat encryption is essential for ensuring the privacy and security of online communications, as it prevents unauthorized access and eavesdropping by malicious parties. An app related to chat encryption would be an encrypted messaging platform or chat application. These apps are designed to prioritize user privacy by implementing strong encryption protocols for all messages exchanged within the platform. Some popular examples of encrypted messaging apps include Signal, WhatsApp (with end-to-end encryption), Telegram (with optional secret chats), and Wickr. These apps offer features such as end-to-end encryption, and secure message and image sharing, providing users with a secure and private communication environment.

3. LITRETURE SURVEY:

Related Works

The Encrypted Chatting App is a modern messaging platform that prioritizes security and user There are countless talk applications that claim to give a protected administration; however, their total design is not freely accessible.

In 2013 Dec, Ali Makki Sagheer et al, proposed a solution that gives secrecy and uprightness to SMS data by applying a crossbred cryptographic plan which join the AES for encryption/unscrambling plan and RC4 for key extension and generation algorithms to satisfy all the more intense security issues. The proposed model is actualized by Java programming dialect in view of Net Beans platform. The proposed framework was tried on different cell phones, for example, the Nokia 5233.Our work uses Secret Key Encryption algorithms that will save the time and cost spent to agree on a key between the users also the encryption time is minimized compered to this paper.

In 2014 May, H.C. Chen et al. exhibited another idea about Mobile Text Chat utilizing a revolution session key-based transposition cryptosystem plan. Their proposed conspire just manages the safe content transposition for mobile chat framework. It acclimatized the technologies of classical block cipher, substitution and transposition. Also, the new session key can be created by the network pivot innovation. It could be easily applied to transmit via mobile devices using the quick encryption algorithm.

In 2014 July, R.N. Akram et al, evaluated the security and privacy preserving features introduced the current mobile chat services. They additionally put advances a fundamental system for an end-to-end security and protection mobile chat service and related necessities. They additionally put advances a fundamental system for an end-to-end security and protection mobile chat service and related necessities. Their proposal was implemented to produce proof-of concept and valuation the technical difficulty of satisfying the specified security and privacy requirements.

In 2014 Nov, Hsing-Chung Chen et al, planned the essential system for secure end to end mobile chat plan and its related necessities. Their proposal is implemented to provide alternate authentication and prevent the password estimating attack and the undetectable on-line password estimating attack. In addition, the plan is a secret key based authentication and key agreement having simple recollected property.

In 2015 Jan, Pejman Dashtinejad, investigated current security features of common messaging applications in the mobile market. A list of requirements for acceptable security is generated and based on those requirements an architecture is developed. A demo is also implemented and evaluated.

4. DESIGN AND IMPLEMTATION

System Design

In this section, we'll delve deeper into the design of the Encrypted chatting app, explaining its core principles of encryption and decryption. The primary aim of the Encrypted chat app is to securely transmit messages between the app and the database using AES algorithm, further enhancing data security during transmission

Implemented Technology:

- *AES Encryption Algorithm*: The app utilizes the AES (Advanced Encryption Standard) algorithm for end-to-end encryption of chat messages. AES is known for its strong encryption capabilities and is widely used for securing sensitive data.
- *Kotlin Programming Language*: The backend logic and encryption functionalities of the app are implemented using Kotlin. Kotlin is a modern and expressive programming language that ensures efficient and reliable code execution
- *Firebase Integration:* Firebase is integrated into the app for secure database storage and real-time synchronization of encrypted chat messages. Firebase Authentication is used for user authentication and access control.
- *Jetpack Compose*: The app's user interface is designed using Jetpack Compose, Google's modern UI toolkit for Android app development. Jetpack Compose simplifies UI development and enhances the overall user experience.

AES Algorithm:

In the Encrypted Chatting App project, the AES (Advanced Encryption Standard) algorithm serves as a cornerstone of security, guaranteeing the confidentiality of chat messages. AES, a symmetric encryption algorithm, encrypts plaintext messages into ciphertext using secret encryption keys before transmission and decrypts them upon receipt, ensuring data remains secure during transit. The app employs robust key generation mechanisms to create unique encryption keys for each chat session, enhancing data confidentiality and preventing unauthorized access. With AES offering varying key lengths like AES-128, AES-192, and AES-256, the app can select the most suitable encryption strength, such as AES-256, for optimal security against cryptographic attacks. Implemented using Kotlin, AES seamlessly integrates into the app's backend, leveraging Kotlin's features for efficient cryptographic operations. Additionally, AES encryption ensures data integrity by detecting unauthorized modifications, complemented by authentication mechanisms to verify message authenticity. Efforts to optimize AES performance, including streamlined key management and encryption processes, further enhance the app's security and user experience.

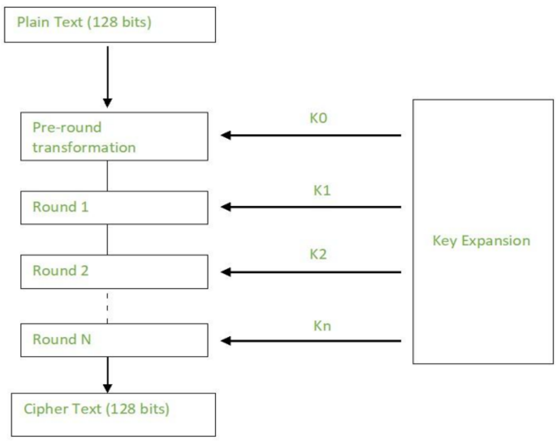
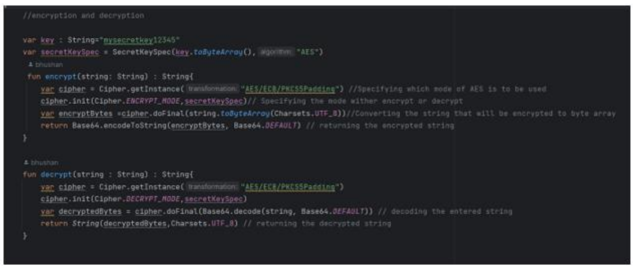


Figure 1: Block Diagram of AES Algorithm

Implementation:

The Encrypted Chatting App is built upon a sophisticated technological foundation, incorporating a myriad of cutting-edge tools to ensure a seamless and secure communication experience. Central to its security architecture is the AES encryption algorithm, renowned for its robustness in safeguarding data. Integrated seamlessly with the Kotlin programming language, this encryption system not only fortifies the app's backend logic but also enables efficient encryption and decryption processes, bolstering user privacy and confidentiality. Kotlin's versatility and expressive nature further enhance the app's security posture, allowing for the seamless implementation of AES encryption across all communication channels within the app. Moreover, the app's integration with Firebase, a comprehensive platform for mobile app development, elevates its security and functionality. Firebase Firestore serves as the backbone for secure database storage and real-time synchronization of encrypted chat messages. This integration ensures data integrity, reliability, and scalability, crucial elements for a robust messaging platform. User authentication, a cornerstone of app security, is impeccably managed through Firebase Authentication, offering secure login mechanisms with Google authentication and access controls. Features such as email/password authentication, social media login integration further fortify user accounts against unauthorized access and data breaches.

The user interface of the Encrypted Chatting App is meticulously crafted using Jetpack Compose, Google's modern UI toolkit tailored for Android app development. Jetpack Compose's declarative programming model simplifies UI development, resulting in intuitive and visually appealing chat interfaces. This not only enhances user engagement but also facilitates seamless and secure communication, maintaining the integrity of encrypted messages throughout the user experience.



```
//encryption and decryption

var key : String="mysecretkey12345"
var secretKeySpec = SecretKeySpec(key.toByteArray(), algorithm: "AES")
 bhushan
 fun encrypt(string: String) : String{
    var cipher = Cipher.getInstance( transformation: "AES/ECB/PKCS5Padding") //Specifying which mode of AES is to be used
    cipher.init(Cipher.ENCRYPT_MODE,secretKeySpec)// Specifying the mode either encrypt or decrypt
    var encryptBytes =cipher.doFinal(string.toByteArray(Charsets.UTF_8))//Converting the string that will be encrypted to byte array
    return Base64.encodeToString(encryptBytes, Base64.DEFAULT) // returning the encrypted string
}

bhushan
fun decrypt(string : String) : String{
    var cipher = Cipher.getInstance( transformation: "AES/ECB/PKCS5Padding")
    cipher.init(Cipher.DECRYPT_MODE,secretKeySpec)
    var decryptedBytes = cipher.doFinal(Base64.decode(string, Base64.DEFAULT)) // decoding the entered string
    return String(decryptedBytes,Charsets.UTF_8) // returning the decrypted string
}
```

Figure 2: AES algorithm implemented using Kotlin language

Methodology:

1. End-to-End Encryption: Implement robust end-to-end encryption to secure all communication between users. Leverage industry-standard encryption algorithms like AES to protect message content from unauthorized access.
2. User Authentication: Use Firebase Authentication to securely authenticate users before granting access to the chat app. Implement Google authentication for user registration in the chat app, ensuring both security and ease of use.
3. UI Design with Jetpack Compose: Utilize Jetpack Compose to design a modern and intuitive user interface for the chat app. Apply Material Design principles to ensure a consistent and familiar user experience.
4. Chat Interface: Develop a dynamic chat interface with Jetpack Compose, enabling real-time sending and receiving of messages. Incorporate features like message threading, multimedia support, and emoji reactions for an engaging user experience.
5. Key Management: Establish a secure key management system for generating and distributing encryption keys to authorized users. Ensure keys are securely stored and exchanged, maintaining the confidentiality of encrypted messages.
6. Secure Storage on Firebase: Utilize Firebase Firestore for encrypted data storage, ensuring all messages are securely stored and accessed. Implement encryption at rest to protect data stored on Firebase servers from unauthorized access.
7. Real-Time Message Delivery: Enable real-time messaging functionality using Firebase Firestore, ensuring instant message delivery. Store user images on Firebase storage. Encrypt messages before transmission and decrypt them upon receipt, maintaining end-to-end security.
8. Secure Error Handling: Encrypt error messages or responses displayed to users to prevent interception by potential attackers. Safeguard the encryption key used for error message encryption, maintaining message integrity.
9. Continuous Security Monitoring: Implement monitoring and logging mechanisms to detect and respond to security incidents in real time. Regularly review security controls and update encryption protocols to mitigate emerging threats.

By incorporating these security measures into your encrypted chat app, you ensure the confidentiality, integrity, and authenticity of user communication, thereby enhancing trust and confidence in the platform.

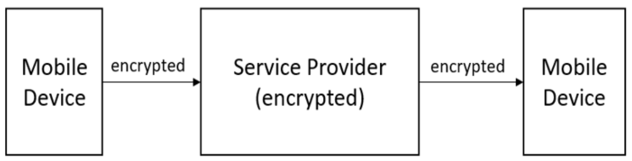


Figure 3: End-to-End Encryption

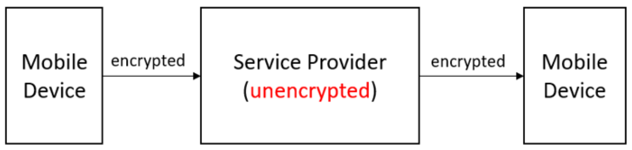


Figure 4: Encryption in Transit

Flowchart:

1. *Start Page*: This is the initial screen of the app. It identifies whether the user is new or old.
2. *Old User*: If the user is an existing user, they are directed to the login page. If the login fails, they are redirected back to the start page.
3. *New User*: If the user is new, they are directed to the register page. If the registration fails, they are redirected back to the start page.
4. *Login/Register* Page: Successful login or registration leads the user to the main page of the app.
5. *Main Page*: This is the central hub of the app from where users can navigate to different fragments.
6. *Profile Fragment*: This section allows users to view and edit their profile information.
7. *Options:* This is a central navigation point leading to profile, user fragments, and chat fragment.
8. *User Fragment*: This section allows users to view other user's profiles.
9. *Chat Fragment*: This is where the encrypted messaging between users happens.
10. *Logout*: Users can securely log out from the chat fragment.

In a nutshell, this flow ensures smooth user interaction and privacy.

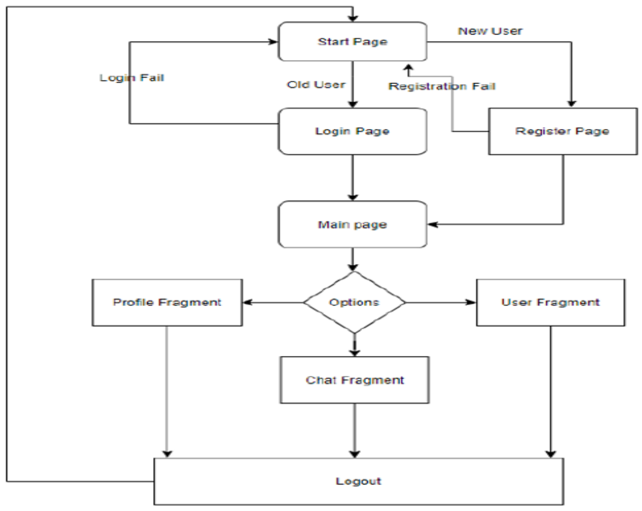
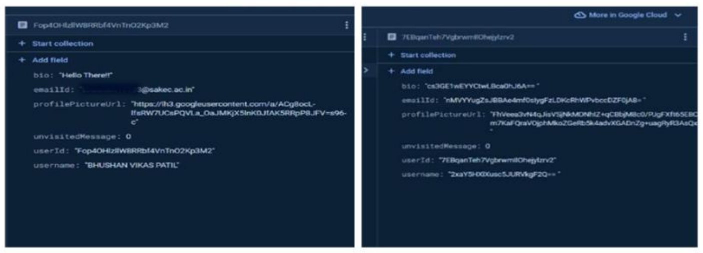


Figure 5: Simple Flowchart of Chatting Application.

5. RESULTS

End-To-End Encryption


Fop4OHlzllWBRRbf4VnTnO2Kp3M2
Start collection
Add field
bio: "Hello There!"
emailId: "[redacted]3@sakec.ac.in"
profilePictureUrl: "https://lh3.googleusercontent.com/a/ACg8ocL-IfsRW7UCsPQVLa_OaJMKjX5lnK0JfAK5RRpP8JFV=s96-c"
unvisitedMessage: 0
userId: "Fop4OHlzllWBRRbf4VnTnO2Kp3M2"
username: "BHUSHAN VIKAS PATIL"

More in Google Cloud
7EBqanTeh7VgbrwmilOhejylzrv2
Start collection
Add field
bio: "cs3GE1wEYYCtwLBcaOhJ6A== "
emailId: "nMVYYugZsJBBAe4mf0slygFzLDKcRhWPvbccDZF0jA8= "
profilePictureUrl: "FhVeea3vN4qJisVSjNkMDNhlZ+qCBbjM8c0/PUgFXfI65EBC m7KaFQraVOjphMkoZGeRb5k4advXGADnZg+uagRyR3AsQx "
unvisitedMessage: 0
userId: "7EBqanTeh7VgbrwmilOhejylzrv2"
username: "2xaY5HXXusc5JURVkgF2Q== "

Figure 6: Before & After Encryption

Our investigation into end-to-end encryption (E2EE) techniques revealed that implementing robust encryption algorithms such as AES, combined with secure key management practices, is essential for ensuring the confidentiality and integrity of user messages. By leveraging E2EE, our encrypted chat application successfully prevents unauthorized access to message content, thereby enhancing user privacy.

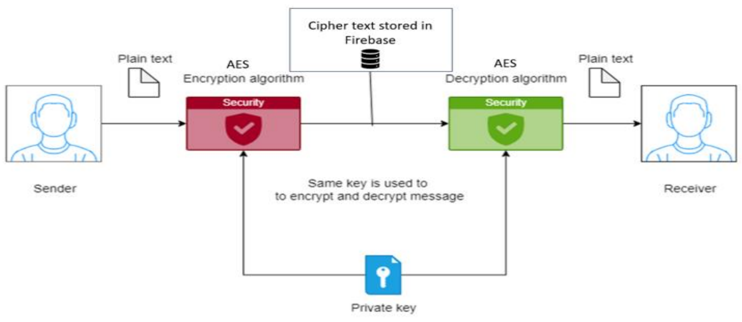


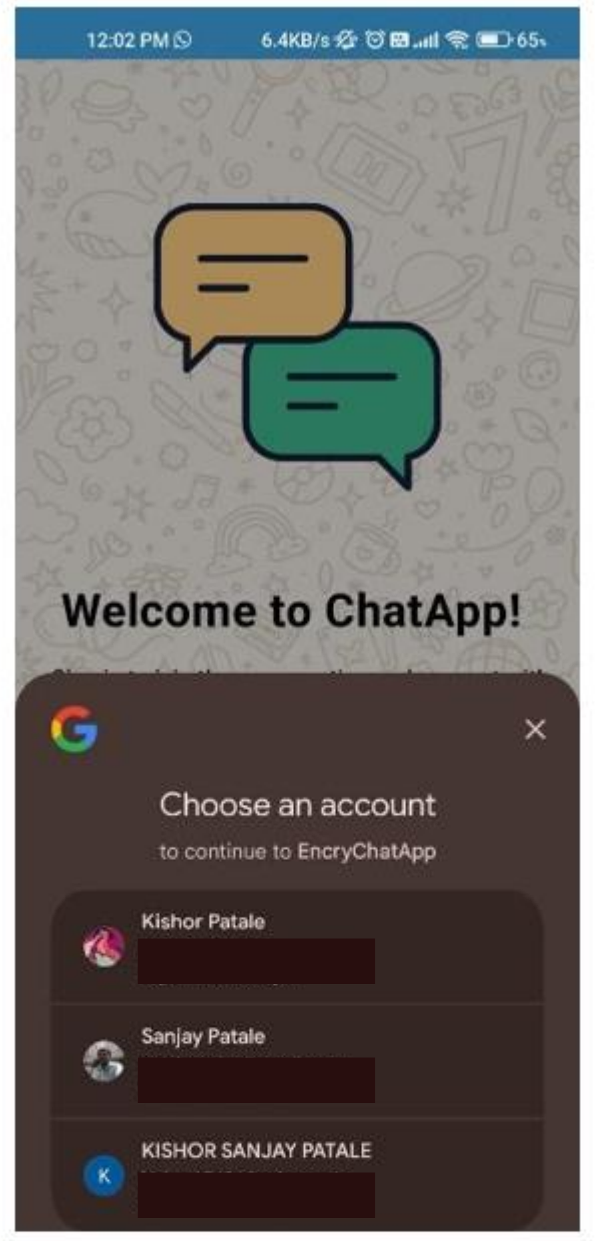
Figure 7: Encryption Overview

Firebase Integration for Real-Time Communication:

Integration with Firebase Realtime Database or Firestore enabled seamless real-time communication within our chat application. Firebase's backend services provided robust infrastructure for message delivery and storage, ensuring reliability and scalability. Additionally, Firebase Authentication ensured secure user identification and verification, further enhancing the application's security posture.

User Authentication and Authorization:

Our examination of user authentication and authorization mechanisms underscored the importance of securely verifying users' identities to prevent unauthorized access to the chat application. Techniques such as Firebase Authentication were found to be effective in ensuring that only authenticated users could access the application's features.



Choose an account
to continue to EncryChatApp

Kishor Patale
[redacted]
Sanjay Patale
[redacted]
KISHOR SANJAY PATALE
[redacted]

Figure 8: Google authentication using Firebase authentication.

6. CONCLUSION

The development and testing of the Encrypted Chatting App represent a significant stride in enhancing communication while prioritizing user security and privacy. This mini project focused on leveraging advanced technologies such as AES encryption, Kotlin programming language, Firebase integration, and Jetpack Compose for a robust and user-friendly messaging platform. One of the key achievements of this project is the implementation of end-to-end encryption using the AES algorithm. This encryption standard ensures that chat messages are encrypted on the sender's device and decrypted only on the recipient's device, mitigating the risks of unauthorized access and data breaches during transmission. The integration of Kotlin facilitated efficient encryption and decryption processes, contributing to a seamless user experience while maintaining data confidentiality.

Firebase integration played a pivotal role in the app's success, providing secure database storage and real-time synchronization of encrypted messages. This ensured data integrity, reliability, and scalability, crucial elements for a messaging platform catering to a diverse user base. The app's user interface, designed using Jetpack Compose, offered a modern and intuitive experience, further enhancing user engagement and satisfaction. The Encrypted Chatting App not only addressed key problems such as end-to-end encryption, protection against hacking, secure database storage, authentication mechanisms, and access controls but also provided additional features such as real-time synchronization, file attachment support, and privacy controls. The app's compliance with privacy regulations and Firebase's secure storage of encrypted messages and user data added layers of security and trust, making it a reliable and preferred choice for users concerned about privacy and data security.

In conclusion, the Encrypted Chatting App mini project successfully demonstrated the integration of advanced technologies to create a secure, efficient, and user-friendly messaging platform. By prioritizing data privacy, security, and user experience, this app sets a high standard in secure communication and serves as a testament to the possibilities of modern technology in enhancing digital communication while safeguarding user information.

Limitations:

1. Simpler User Experience: Project has prioritized simplicity over advanced user experience enhancements, resulting in a basic or less intuitive user interface, limited customization options, or fewer collaboration tools.
2. Resource and Time Constraints: Limited resources, budget constraints, or time limitations in entry-level projects may impact the depth of development, testing, optimization, or on-going support, affecting the overall quality and reliability of the app, may take significant amount of time to decrypt messages.
3. Basic Security Features: Project lacks advanced security features such as biometric authentication, two-factor authentication (2FA), secure voice/video calls, or comprehensive auditing and monitoring capabilities, limiting security enhancements.

4. Technical Dependencies: Projects rely on third-party libraries, frameworks, or services for encryption, database management, or security features, introducing dependencies that could pose integration challenges or compatibility issues.
5. Foundational Key Management Practices: Managing encryption keys securely may be challenging at an entry-level, leading to potential key management issues such as key leakage, inadequate key rotation practices, or insufficient key length, compromising data security.
6. Technical Dependencies: Projects rely on third-party libraries, frameworks, or services for encryption, database management, or security features, introducing dependencies that could pose integration challenges or compatibility issues.

Future Scope:

1. Multi-platform Compatibility: Extend the app's compatibility to various operating systems and devices, including mobile platforms (iOS, Android), web browsers, and desktop applications using KMM multiplatform, to reach a wider user base.
2. Enhanced Security Features: Introduce additional security features, such as biometric authentication (fingerprint, facial recognition), two-factor authentication (2FA), and secure voice/video calls, to strengthen user authentication and communication security.
3. Advanced Encryption Standards: Implement advanced encryption standards and protocols, such as post-quantum cryptography or homomorphic encryption, to enhance security and stay ahead of emerging cyber threats.
4. Compliance and Auditing: Enhance compliance with regulatory requirements by implementing robust auditing mechanisms, data retention policies, and compliance reporting features to ensure adherence to data protection laws and standards.
5. AI-driven Security: Explore the use of artificial intelligence (AI) and machine learning (ML) algorithms for threat detection, anomaly detection, and predictive analysis to proactively identify and mitigate security risks and vulnerabilities.
6. Localization and Globalization: Customize the app for different languages, cultures, and regions, ensuring localization support and global accessibility for users worldwide.
7. Partnerships and Integrations: Forge partnerships with cybersecurity firms, encryption experts, and technology providers to leverage their expertise, tools, and services for continuous security enhancements and integrations with third-party services.

ACKNOWLEDGEMENT

The heading should be treated as a 3rd level heading and should not be assigned a number.

REFERENCES

1. Hsing-Chung Chen, Jyh-Horng Wen and Cheng-Ying Yang, “A Secure End-to-End Mobile Chat Scheme”, Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, 2014.
2. William Stallings, “Cryptography and Network Security: Principles and Practice”, Prentice Hall, Boston, 5th Ed, 2011.
3. Li Zhang, Chao Xu, Parth H. Pathak, and Prasant Mohapatra, “Characterizing Instant Messaging Apps on Smartphones”, Passive and Active Measurement Lecture Notes in Computer Science, pp. 83-95, 2015.
4. S. Prabhune and S. Sharma, "End-to-End Encryption for Chat App with Dynamic Encryption Key," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021
5. Ali Makki Sagheer, Ayoob Abdulmunem Abdulhameed and Mohammed Adeeb AbdulJabbar, “SMS Security for Smartphone”, Sixth International Conference on Developments in eSystems Engineering, 2013. [6] H.C. Chen and A.L.V. Epa, “A Rotation Session KeyBased Transposition Cryptosystem Scheme Applied to Mobile Text Chatting”, Proceedings of The 28th IEEE International Conference on Advanced Information Networking and Applications (AINA2014), pp. 497 - 503, Victoria, Canada, May 2014.
6. Raja Naeem Akram, and Ryan K. L. Ko. “End-to-End Secure and Privacy Preserving Mobile Chat Application”, Information Security Theory and Practice. Securing the Internet of Things Lecture Notes in Computer Science, pp.124-139, 2014.
7. Pejman Dashtinejad,” Security System for Mobile Messaging Applications “, Thesis, KTH University, Jan 2015
8. Bhimrao Patil, “SMS SECURITY USING RC4 & AES”, Indian J.Sci.Res, Vol. 11(1), pp. 34-38, 2015.
9. Duane Booher, Bertrand Cambou, Albert H. Carlson and Christopher Philabaum, "Dynamic Key Generation for Polymorphic Encryption", 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0482-0487, January 2019.
10. S. Prabhune and S. Sharma, "End-to-End Encryption for Chat App with Dynamic Encryption Key," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021
11. https://www.baeldung.com/kotlin/advanced-encryption-standard Kotlin AES Encryption and Decryption
12. Advanced Encryption Standard (AES) fact sheet. Available at http://csrc.nist.gov/encryption/aes/aesfact.html.
13. https://getstream.io/blog/encrypted-messaging-app-android/ Build an Encrypted Messaging App for Android
14. Akram R.N., Ko R.K.L. (2014) End-to-End Secure and Privacy Preserving Mobile Chat sApplication. In: Naccache D., Sauveron D. (eds) Information Security Theory and Practice. Securing the Internet of Things. WISTP 2014. Lecture Notes in Computer Science, vol 8501. Springer, Berlin, Heidelberg
15. https://developer.android.com/studio/write/firebase
16. https://www.ijser.in/archives/v2i2/SjIwMTMxMTU=.pdf Analysis and Review of Encryption and Decryption for Secure Communication