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

KEYs Password Management application

Dr. Anil V Turukmane* [1]
Professor,SCOPE
Aman Dhakar[2] , Bheri Saiteja[3] ,Abbaraju Sai Sathwik[4]

School of Computer Science and Engineering (SCOPE), VIT-AP University, India.

Abstract. "KEYs" is an Android-based password management app that prioritizes security through a myriad of features, all while offering a user-centered design via Jetpack Compose and MVVM architecture. The app generates robust passwords from a complex character set and encrypts user data using AES with a 256-byte key stored in Android's Keystore. Firebase serves as the real-time database with stringent security rules tied to unique user IDs. Advanced security measures include variable Initialization Vectors for encryption, end-to-end encrypted password sharing, and biometric authentication for password-less access. Firebase security rules, device monitoring, and code obfuscation further fortify the app against unauthorized access and reverse engineering.

Keywords: Password Management, Jetpack Compose, MVVM Architecture, AES Encryption, Biometric Authentication, Security Measures, User Authorization, Real-time Database, Unique User ID, Secure Data Transfer, Shared Preferences.

1. Introduction

1.1. Background

In the digital age, password management has become an essential yet cumbersome task for individuals and organizations alike. The proliferation of online services requires users to remember multiple complex passwords which can lead to vulnerabilities and inefficiencies. The need for a secure and convenient solution to manage these passwords has led to the development of KEYs, an Android-based application specifically designed to address these challenges.

1.2. Objective and Scope

The primary objective of KEYs is to provide a user-friendly platform that enables secure password storage, encrypted note-taking, document saving, chatting, and auto-filling functionalities within the same mobile device. The scope of this paper is to explore the architecture, technologies, features, and development process of KEYs, highlighting its significance in enhancing both security and convenience for users.

1.3. Technologies and Architecture

KEYs leverages cutting-edge technologies such as Jetpack Compose for UI development, AES encryption for security, Firebase as a live database, and Dagger Hilt for dependency injection. The application follows the Model-View-ViewModel (MVVM) architecture, fostering maintainability, scalability, and a clean separation of concerns. These technologies work in harmony to create a robust and efficient frontend.

1.4. Unique Features and Innovation

What sets KEYs apart from other password management solutions are its specialized features that go beyond mere password storage. The inclusion of encrypted chatting, document saving, and auto-filling within the mobile device adds layers of functionality, making it a comprehensive tool for personal information management. These innovations not only improve user experience but also contribute to the overall security of the application.

1.5. Structure of the Paper

The remainder of this paper is organized as follows: Section 2 details the methodology, including the development environment, architecture, security features, and additional functionalities; Section 3 presents the evaluation and testing of KEYs, highlighting its performance and reliability; and Section 4 concludes the paper, summarizing the key findings and suggesting future directions for research and development.

2. Literature Survey

Three-Factor Authentication Protocols
Le et al. discuss the implementation of a three-factor authentication protocol optimized for 6G-aided intelligent healthcare systems in their 2022 paper[11]. This work is pertinent to "KEYs" as it delves into multi-layered authentication techniques, something our application aims to achieve through unique IDs and biometric authentication.

Online Password Guessing Attacks
Guan and Chen's work on a novel verification scheme to resist online password guessing attacks provides valuable insights into creating a more secure authentication process[2]. Their methodologies can be used to augment the already robust AES-256 encryption and initialization vectors in "KEYs."

Cyber Risk Analysis
The probabilistic analysis of cyber risks by Paté-Cornell and Kuypers offers a comprehensive understanding of the vulnerabilities that might exist in cyber systems[3]. This knowledge is crucial for our application, which focuses intensely on secure data storage and transfer.

Decentralized Authentication
Moreno et al.'s paper on a trusted approach for decentralized and privacy-preserving identity management touches upon the importance of user-specific customization in identity verification[4]. This is directly related to the unique user IDs and specific Firebase security rules in our app.

IoT Security
Rahmani et al. introduce an advanced mobile authentication protocol specifically for GLOMONET[6]. Given that "KEYs" aims for scalability and integration with other smart devices, the work can serve as a roadmap for future IoT compatibility.

Blockchain in Authentication
Ma Zhaofeng's work on a blockchain-based decentralized authentication modeling scheme brings to light the future possibilities of integrating blockchain for enhanced security[7].

Code Security & Obfuscation
Rahman and Parnin's upcoming paper on detecting and characterizing the propagation of security weaknesses in infrastructure management can serve as a guideline for further enhancing the source code security in "KEYs," complementing the already implemented code obfuscation techniques[13].

Modern Authentication Schemes
Ahvanooey et al. provide an empirical survey of modern authentication schemes in smartphones and IoT devices[14]. The paper can serve as a comprehensive guide to evaluate the biometric authentication feature in “KEYs."

Pal et al. offer valuable data on IoT-specific cyber-risks, which could be vital as your app scales[5]. Rana et al. discuss computational efficiency in authenticated digital content distribution, aligning well with your app's focus on speed and secure data handling[8]. Yu's upcoming work on counteracting side channels offers strategies that could be applied to your data encryption process[^9^]. The works of Li, Wei et al. and Li, Yan et al. delve deep into secure authentication and leakage-resilient password systems, which would add weight to your app's biometric and unique ID features[10][1]. Ma et al. highlight common authentication flaws in Android apps, providing a preventive perspective for your project[15]. Tiwari and Hubballi offer methodologies for detecting bruteforce attacks, complementing your project's emphasis on security[16]. Wu et al. and Ryu et al. discuss unified access management and mobile authentication schemes, potentially enriching your app's existing systems[17][18]. Lastly, Li, Yue et al. analyze real-world account recovery issues, helping you consider practical challenges in your app's design[19].

3. Proposed Work

3.1. Enhancing Security Measures

As technology advances, so do the threats that challenge the security of applications like KEYs. While the current implementation has robust security features such as AES encryption, there is always room for further enhancement. Proposed future work includes integrating multi-factor authentication (MFA) and biometric security features. These added layers of security will fortify the application against potential vulnerabilities, ensuring that user data remains inaccessible to unauthorized parties.

3.2. Expanding Functionalities and Cross-Platform Compatibility

KEYs has demonstrated significant potential as a comprehensive password management tool. Future development could explore expanding its functionalities to include features like shared access, backup recovery options, or integration with wearables. Additionally,

developing a cross-platform version of KEYs would allow users to access their data across various devices, not just Android, enhancing usability and reach. A consistent experience across platforms can be achieved by employing frameworks that allow for cross-platform development while maintaining the unique features and security measures that set KEYs apart

4. Methodology

4.1. Development Environment and Tools

The development of KEYs was performed using Android Studio, a popular Integrated Development Environment (IDE) for Android application development. Programming languages such as Kotlin, and XML were employed for building the user interface, demonstrating a modern approach to Android development. Dagger Hilt, a dependency injection library, was incorporated to manage dependencies within the application, improving maintainability and scalability.

4.2. Password Generation Function

The password generation feature utilized a pool of 58 characters, consisting of 26 alphabets (both lower-case and upper-case), 10 numbers, and 22 special characters. A randomizing algorithm was implemented to generate complex passwords by selecting a balanced mix of these characters, to achieve high entropy.

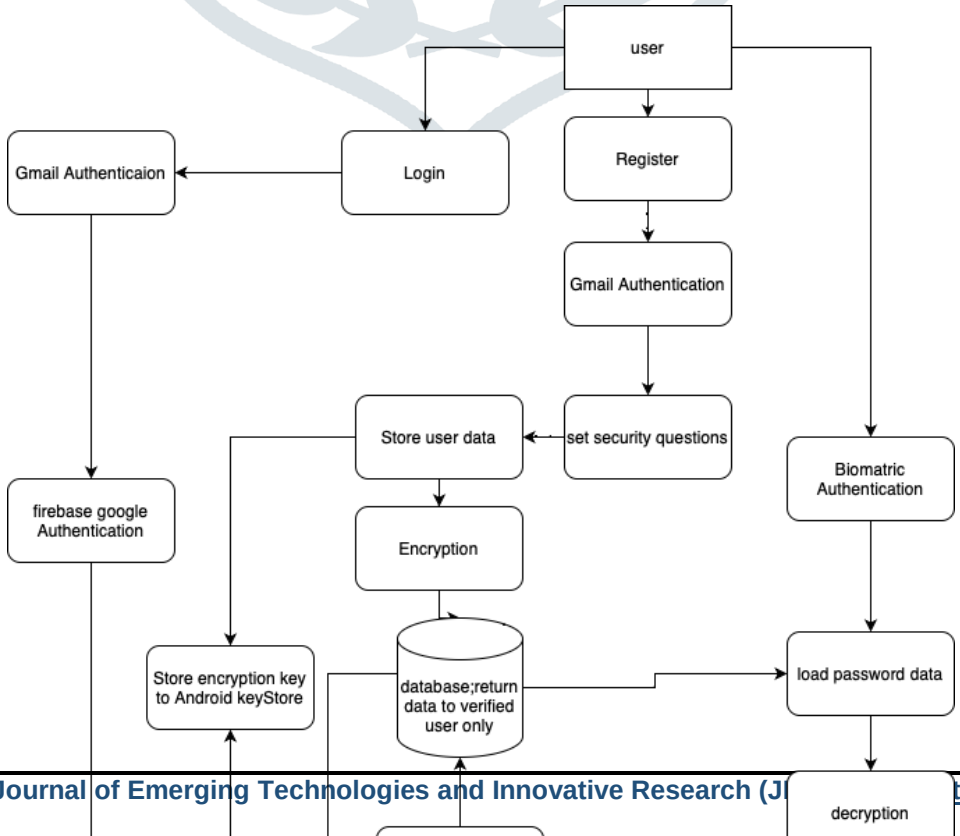
4.3. Architecture: MVVM Pattern

KEYs adheres to the Model-View-ViewModel (MVVM) architectural pattern, a proven structure that promotes a clean separation of concerns. The Model represents the data and business logic, the View handles the user interface, and the ViewModel acts as a bridge between the Model and View. This architecture ensures that the UI code is decoupled from the business logic, allowing for more manageable code and efficient testing.

4.4. Security Measures: AES Encryption

Security is paramount in KEYs, and AES encryption plays a crucial role in safeguarding the sensitive information stored within the application. AES, or Advanced Encryption Standard, encrypts the data using symmetric key algorithms, ensuring that notes, documents, and other personal information are stored securely. This encryption method offers robust protection against potential breaches and unauthorized access.AES encryption with a 256-byte long key was used for encrypting user data. Each user's encryption key was securely stored in the Android Keystore system. For added complexity, different Initialization Vectors (IVs) were employed for each encryption process. Firebase security rules were customized for each user based on a unique ID generated during account creation, ensuring granular access control.

Fig.1: Working model of "KEYs"



4.5. User Authorization and Biometrics

User authentication was implemented without the need for a password. Instead, biometric authentication was em0070loyed, leveraging Android's native biometric API. Each device had to be authorized through this method, and all devices linked to a specific user account were monitored to prevent unauthorized access.

4.6. Real-Time Database: Firebase Integration

Firebase serves as a live database for KEYs, offering real-time data synchronization across devices. This cloud-based NoSQL database enables seamless data management and retrieval, enhancing the user experience. Firebase's integration ensures that user data remains accessible and synchronized, providing a consistent experience across various devices. All user-specific data were stored in Firebase. For less sensitive data, Android's SharedPreferences were used. The Firebase database was configured to deny all unsigned requests, adding another layer of security.

4.7. Code Security

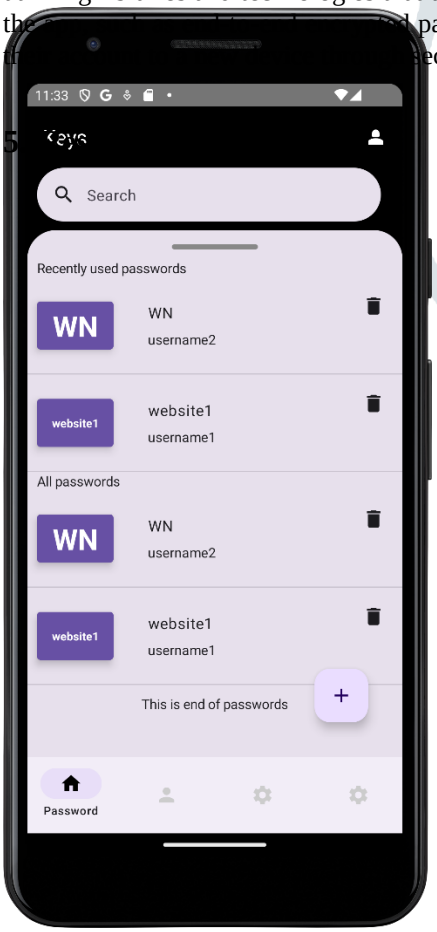
The source code was obfuscated using ProGuard, making it difficult for attackers to reverse-engineer the application. All network communications for data transfer were secured using HTTPS, ensuring encrypted data flow between the client and server.

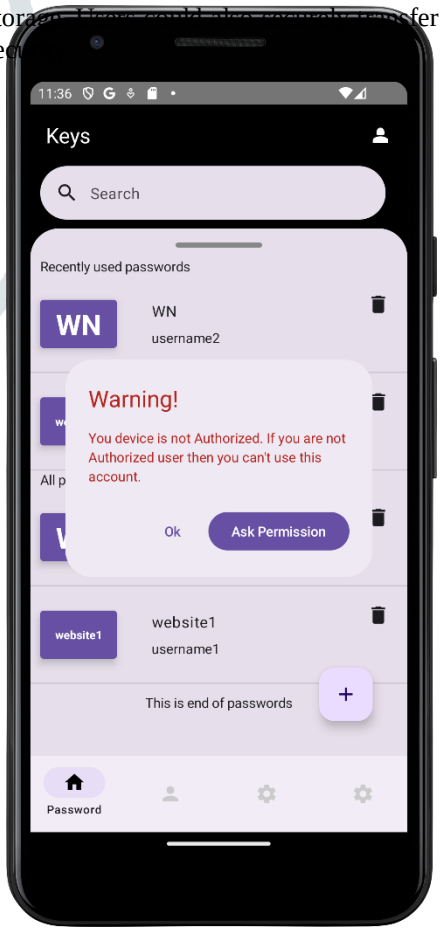
4.8. Testing and Monitoring

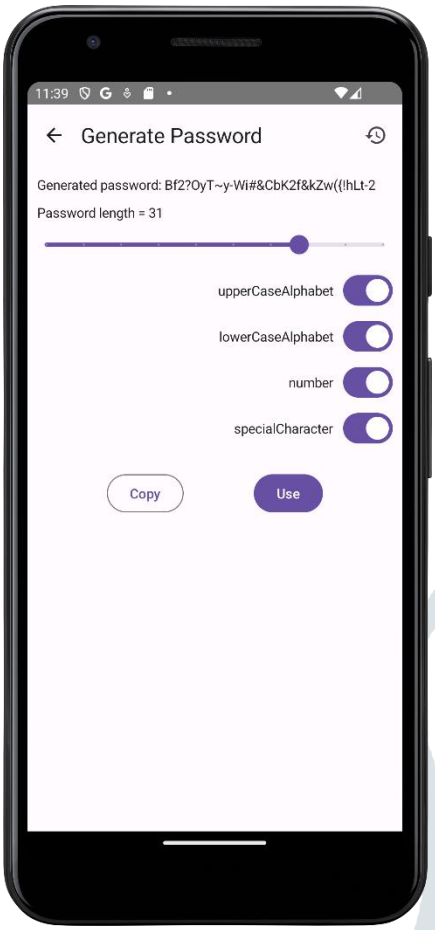
Unit and Integration tests were conducted using JUnit and Espresso. Firebase Analytics was also used to monitor real-time usage and to identify any potential vulnerabilities.

4.9. Additional Features and Functionalities

Beyond the core functionalities of password management, KEYs incorporates unique features such as encrypted chatting, auto-filling within the same mobile device, and more. These additional functionalities are designed to provide users with a comprehensive suite of features that goes beyond mere password storage. The development of these features required careful planning and implementation, utilizing libraries and technologies that align with the overall goals of the application. Several additional features were integrated into th password sharing, secure note-taking, and document stora fer
ecurity questions, which were randomized for added sec







6. Comparative analysis

When compared to existing password management solutions like LastPass, Dash Lane, or native Android password managers, "KEYs" distinguishes itself through its heavy emphasis on user-specific customization and security. Most commercially available password managers prioritize simplicity and user-friendliness but often at the cost of advanced security features. "KEYs," however, integrates advanced AES encryption with a 256-byte long key, unique Initialization Vectors (IVs), and a real-time Firebase database with user-specific security rules, setting a new benchmark in the sector.

Furthermore, the app's UI, developed using Jetpack Compose, Kotlin, and XML, offers an intuitive and streamlined experience without sacrificing security. While many existing solutions use either overly simplistic interfaces that lack advanced options or complicated dashboards that can overwhelm average users, "KEYs" strikes a balance. Its use of biometric authentication for password-less sign-ins aligns it with user convenience, a feature that is often reserved for premium versions in other apps. Perhaps most notably, "KEYs" adds additional layers of security not commonly found in other password management apps. For instance, end-to-end encrypted password sharing and secure device transfer via security questions set it apart. The inclusion of special features like making encrypted notes and storing important documents adds multi-functionality to the app. Additionally, its backend, built on the MVVM architecture, makes it scalable and maintainable, suggesting longevity and reliability in comparison to more straightforward, less modular solutions.

7. Conclusion

"KEYs" stands as a robust, user-centric Android password management application that sets a new standard for mobile security and user experience. Developed using a mix of modern technologies like Jetpack Compose, Kotlin, and Firebase, the app leverages advanced encryption algorithms and meticulous data access controls to safeguard user data. Its unique blend of high-end security features, like AES encryption with 256-byte keys and end-to-end encrypted password sharing, is not just a proof of concept but a live demonstration of how meticulous engineering can elevate user privacy and data protection. The user-friendly interface, built with a combination of Kotlin, and XML, ensures that even the less tech-savvy users can easily navigate through the app, marking it as a comprehensive solution in the mobile security landscape.

8. Further scope

The future scope of KEYs is vast and brimming with possibilities. As the demand for secure and efficient password management continues to grow, KEYs has the potential to evolve into an all-encompassing personal information management system. Beyond its cur-

rent features, future iterations could explore integration with Internet of Things (IoT) devices, voice command capabilities, and even artificial intelligence-driven security enhancements. By keeping pace with technological advancements and prioritizing user-centric design, KEYs can become a pioneering solution in the realm of privacy and personal data management, setting new standards for security, functionality, and user experience in the ever-expanding digital landscape.

References

1. Li, Yan, et al. "Designing leakage-resilient password entry on head-mounted smart wearable glass devices." IEEE Transactions on Information Forensics and security 16 (2020): 307-321.
2. Guan, Albert, and Chia-Mei Chen. "A Novel Verification Scheme to Resist Online Password Guessing Attacks." IEEE Transactions on Dependable and Secure Computing 19.6 (2022): 4285-4293.
3. Paté-Cornell, M-Elisabeth, and Marshall A. Kuypers. "A probabilistic analysis of cyber risks." IEEE Transactions on Engineering Management 70.1 (2021): 3-13.
4. Moreno, Rafael Torres, et al. "A trusted approach for decentralised and privacy-preserving identity management." IEEE Access 9 (2021): 105788-105804.
5. Pal, Ranjan, et al. "Aggregate cyber-risk management in the IoT age: Cautionary statistics for (re) insurers and likes." IEEE Internet of Things Journal 8.9 (2020): 7360-7371.
6. Rahmani, Amir Masoud, et al. "AMAPG: Advanced mobile authentication protocol for GLOMONET." IEEE Access 9 (2021): 88256-88271.
7. Zhaofeng, Ma, et al. "Blockchain-based decentralized authentication modeling scheme in edge and IoT environment." IEEE Internet of Things Journal 8.4 (2020): 2116-2123.
8. Rana, Saurabh, et al. "Computational efficient authenticated digital content distribution frameworks for drm systems: Review and outlook." IEEE Systems Journal 15.2 (2020): 1586-1593.
9. Yu, Chia-Mu. "Counteracting Side Channels in Cross-user Client-side Deduplicated Cloud Storage." IEEE Internet of Things Journal (2023).
10. Li, Wei, et al. "Design of secure authenticated key management protocol for cloud computing environments." IEEE Transactions on Dependable and Secure Computing 18.3 (2019): 1276-1290.
11. Le, Tuan-Vinh, et al. "A novel three-factor authentication protocol for multiple service providers in 6G-aided intelligent healthcare systems." IEEE Access 10 (2022): 28975-28990.
12. Li, Yan, et al. "Designing leakage-resilient password entry on head-mounted smart wearable glass devices." IEEE Transactions on Information Forensics and security 16 (2020): 307-321.
13. Rahman, Akond, and Chris Parnin. "Detecting and Characterizing Propagation of Security Weaknesses in Puppet-based Infrastructure Management." IEEE Transactions on Software Engineering (2023).
14. Ahvanooey, Milad Taleby, et al. "Modern authentication schemes in smartphones and IoT devices: An empirical survey." IEEE Internet of Things Journal 9.10 (2021): 7639-7663.
15. Ma, Siqi, et al. "Orchestration or automation: authentication flaw detection in android apps." IEEE Transactions on Dependable and Secure Computing 19.4 (2021): 2165-2178.
16. Tiwari, Namrata, and Neminath Hubballi. "Secure Socket Shell Bruteforce Attack Detection With Petri Net Modeling." IEEE Transactions on Network and Service Management 20.1 (2022): 697-710.
17. Wu, Liangshun, H. J. Cai, and Han Li. "SGX-UAM: A secure unified access management scheme with one time passwords via intel SGX." IEEE Access 9 (2021): 38029-38042.
18. Ryu, Jihyeon, et al. "SMASG: secure mobile authentication scheme for global mobility network." IEEE Access 10 (2022): 26907-26919.
19. Li, Yue, et al. "Understanding account recovery in the wild and its security implications." IEEE Transactions on Dependable and Secure Computing 19.1 (2020): 620-634.
20. Turukmane, Anil V. "Forecasting the IoT-based cyber threats using the hybrid forage dependent ensemble classifier." *Concurrency and Computation: Practice and Experience* 35.2 (2023): e7460.
21. Turukmane, Anil V., et al. "Multispectral image analysis for monitoring by IoT based wireless communication using secure locations protocol and classification by deep learning techniques." *Optik* 271 (2022): 170122.
22. Turukmane, Anil V., et al. "Smart farming using cloud-based Iot data nalytics." *Measurement: Sensors* 27 (2023): 100806.
23. Chaudhari, K. P., and Anil V. Turukmane. "Dynamic probabilistic packet marking." *International Conference on Advances in Information Technology and Mobile Communication*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.
24. Roopa, Y. M., SatheshKumar, T., Mohammed, T. K., Turukmane, A. V., Krishna, M. S. R., & Krishnaiah, N. (2022). Power allocation model for residential homes using AI-based IoT. *Measurement: Sensors*, *24*, 100461.