Iraqi Journal for Computer Science and Mathematics

Volume 5 | Issue 4 Article 20

2024

Develop Secure Software specifications for Android App concealing the Information and Safeguarding Data

Huda Abdulaali Abdulbaqi
Computer Science Department, College of Science, Mustansiriyah University, Baghdad, Iraq, huda.it@uomustansiriyah.edu.iq

Ahmmad Mohamad Ghandour
Computer and Communication Engineering Islamic University of Lebanon, Lebanon, ahmad.ghandour@iul.edu.lb

Thekrayat Abbas Jawad
Computer and communications Engineering, Baghdad, Iraq, lady_th@yahoo.com

Recommended Citation

Abdulbaqi, Huda Abdulaali; Ghandour, Ahmmad Mohamad; and Jawad, Thekrayat Abbas (2024) "Develop Secure Software specifications for Android App concealing the Information and Safeguarding Data," *Iraqi Journal for Computer Science and Mathematics*: Vol. 5: Iss. 4, Article 20.
DOI: https://doi.org/10.52866/2788-7421.1215
Available at: https://ijcsm.researchcommons.org/ijcsm/vol5/iss4/20

IRAQI JOURNAL FOR COMPUTER SCIENCE AND MATHEMATICS 2024;5:243–257

RESEARCH ARTICLE

Develop Secure Software specifications for Android App concealing the Information and Safeguarding Data

Huda Abdulaali Abdulbaqi [a,*], **Ahmmad Mohamad Ghandour** [b], **Thekrayat Abbas Jawad** [b]

[a] Computer Science Department, College of Science, Mustansiriyah University, Baghdad, Iraq
[b] Computer and Communication Engineering Islamic University of Lebanon, Lebanon

ABSTRACT

In the current landscape of technological advancement, data holds a pivotal role, shaping societal interactions and daily routines. The rapid escalation in digital data volume, driven by technological strides, has underscored the critical necessity for robust protective measures to safeguard its sensitive nature. This study aims to develop a secure software specification for Android application ensuring effective data protection through a specialized Android application tailored explicitly for data concealment, assuring utmost confidentiality and secure transmission. In this paper we revolve around the integration of multifaceted security and privacy protocols, employing advanced information concealment techniques, encryption mechanisms, secure key management, and meticulous access control by MAC address. Termed 'Secure Data Applications,' these interfaces are meticulously crafted using PHP interlinked with MySQL databases, and developed using Flutter and Dart languages. The primary goal is to make a substantial contribution to the domain of mobile data protection, enabling secure and streamlined data exchanges while placing paramount importance on an intuitive user experience The 'Secure Data App' application, emerged successfully through a battery of rigorous testing methodologies. Throughout the testing phases, the application seamlessly navigated without encountering any errors, affirming its robustness and reliability. Moreover, the tested value (6.8%) of maximum CPU usage signifies that the application utilizes the CPU at a relatively low rate. Also, the encryption ratio of 1.35 is relatively high which guarantees a higher level of security.

Keywords: Information security, Software specification, Functional requirement, Cryptography, Access control

1. Introduction

Because of the speed at which information technology is developing, the amount and significance of digital data is expanding tremendously in the modern era. The widespread use of smart devices, especially Android smartphones, has become essential to our day-to-day lives. A wide range of uses, including the safeguarding of sensitive data, communication, entertainment, and business administration [1].

However, even with all of the benefits that come with our increasing reliance on technology, we must admit that there are concerns and doubts about the matter of protecting data integrity and protection.

As a result, protecting private and sensitive information has become increasingly important, which emphasizes the necessity to investigate the creation of an Android application especially meant for concealment matter of protecting data integrity [2].

Received 15 February 2024; accepted 21 February 2022.
Available online 6 December 2024

* Corresponding author.
E-mail addresses: huda.it@uomustansiriyah.edu.iq (H. A. Abdulbaqi), ahmad.ghandour@iul.edu.lb (A. M. Ghandour), lady_th@yahoo.com (T. A. Jawad).

https://doi.org/10.52866/2788-7421.1215

Programmers who have access to highly configurable and feature-rich tools can make a big difference in data security. This idea applies to both private citizens and businesses. With the use of these technologies, software developers may incorporate strong security measures into their apps and protect sensitive data from tampering or unauthorised access. However, it is also critical to recognize that user awareness and education must be given equal priority with technology solutions. This is essential for encouraging responsible data usage practices and lowering the possibility of human error. Through focusing on user awareness and education, we can improve the overall security architecture of applications. This proactive approach comprises informing customers about the risks associated with their data and providing them with appropriate information to effectively mitigate these risks. Through intuitive user interfaces, educational alerts, and comprehensive tutorials, users can gain a deeper understanding of how to protect their information and securely utilize the service. Furthermore, incorporating user education into the application experience itself helps users adopt security best practices into their daily lives, which lowers the possibility of unintentional data disclosure or misuse. [3].

A comprehensive strategy that takes into account several elements, including encryption techniques, secure storage mechanisms, access control protocols, and secure communication channels, is necessary due to the varying security characteristics of Android applications. This study attempts to provide developers with useful information and suggestions on how to smoothly integrate robust security features into their Android applications by carefully examining each of these components. By means of a thorough analysis, realistic implementation strategies, and empirical evaluations, this study seeks to significantly influence the field of mobile application security, thereby encouraging a security-conscious design mindset and fortifying the core foundation of trust in the Android application ecosystem [4].

Our main goal is to strengthen the protective measures by combining technology and privacy regulations. Techniques like access control, steganography, cryptography, and secure key management are the essential building blocks used in this particular field with the main goal of guaranteeing the highest level of protection for the confidentiality, integrity, and verifiable source of the inherent sensitive information. Within this domain, the aforementioned techniques are widely utilized to create a strong framework that prevents unauthorized access to sensitive information, guards against data alteration or tampering, and guarantees the integrity and verification of the data.

1.1. Android overview

The development of Android can be credited to a calculated alliance by several companies that formed the Open Handset Alliance (OHA). Notable members of the OHA include Samsung, Sony, Intel, and a number of other notable companies. The OHA aims to provide a plethora of services while simultaneously ensuring the expansion of Android-powered handsets.

Android has seen numerous updates since its first appearance, each of them giving codename this code was derived from a delectable treat, such as Apple Pie, Gingerbread.

The operating system of Android mobile is notable for being based on an open-source foundation, which makes anyone use its source code without any limitations. In the core is, the Linux kernel—which serves as the base for Android is important for guaranteeing user security since it default to access to system for that its imposable access to the data.

1.2. Security issues of android

Security standards have been very important in order to ensure the truest and of important data included in Android applications, it is necessary to apply a group of these apps. These Apps

Data Encryption:
Strong encryption solutions are necessary to protect personal data while it is being kept on devices and sent over networks.

Secure Key Management:
Use of the Android Key Store System for the safekeeping of cryptographic keys is advised in order to improve encryption process security and protect keys from unwanted access.

Data Steganography:
Steganography techniques are recommended for the discrete integration of private data into different files or media. Since many decades ago, steganography has been used to protect communication confidentiality.

Concealed or veiled, and “graphein,” which denotes writing.

The Greek word “steganos,” which meaning to be extracted at the desired location and includes text, photos, videos, and music, is where the name first appeared. It involves putting data into digital files so that they can later.

Table 1. Summary of previous studies.

Thesis reference	Technology used	Application field
[6]	Data encryption.	Hidden applications on Android.
[7]	Blockchain Technology With Data Encryption and Decryption Technology.	Industrial Internet of Things (IIoT).
[8]	Blockchain Technology and Interplanetary File System (IPFS).	Electronic Health Records.
[9]	Blockchain Technology With Federated Learning	Industrial Internet of Things (IIoT).
[10]	Encryption Techniques and Access Control Measures	Industrial Internet of Things (IIoT).
[11]	outline major Blockchain technology that based as solutions for IOT security.	IOT security
[12]	AES (Advanced Encryption Standard) Algorithm and LSB (Least Significant Bit) Steganography Process.	High-Efficiency Video Coding (HEVC) Streams.
[13]	Blockchain Technology and Artificial Intelligence (AI).	Sharing CT Images in Medical Facilities.
[14]	Blockchain With Attribute-Based Encryption (ABE).	5G Aerial Drones.
[15]	Blockchain and The A3C Learning Approach.	Mobile-edge computing (MEC).
[16]	Blockchain Technology and Smart Contracts.	IoT Devices.
[17]	Attribute-Based Encryption (ABE) With Blockchain Technologies.	Heath Care.
[18]	Blockchain Technology and an Asynchronous Advantage Actor-Critic (A3C).	Mobile-Edge Computing.
[19]	Encryption Techniques, Authentication Mechanisms, and Access Control Protocols.	Medical Cyber-Physical Systems (CPS).
[20]	Blockchain Technology.	Smart Cities
[21]	Blockchain Technology and The Interplanetary File System (IPFS)	Common.

Steganography and cyber security are closely related fields because bad actors often use hidden data in their attacks, enclosing information, dangerous tools, or instructions in files that appear innocuous.

Biometric Authentication:
Because biometric authentication relies on distinct physical attributes or characteristics that are only possessed by authorized personnel, it adds an extra layer of protection and greatly lowers the possibility of fraudulent activities or unauthorized penetration.

App Permissions and Access Controls:
Limit the amount of resources that an application can use from the device by using Android's permission system.

The possibility of unauthorized access to private information is reduced by minimizing needless permissions.

Access control systems are in charge of allocating and overseeing the users of the programmer who have access to specific data. These strategies seek to precisely restrict access rights by establishing privileges and permissions in line with user roles and requirements. By preventing unauthorized access to sensitive information, access limits help to strengthen data confidentiality and prevent security breaches.

2. Literature review

The twenty-first-century digital landscape has brought about an era in which information technology that becoming more active and fast dissemination, necessitating both a global shift and a significant improvement in competitive capacities. that means to ensure the highest level of data security during sending and receiving of information, hiding, and encryption must function in unison. Zero-day vulnerabilities are created when virus and phishing site signatures are updated more slowly than expected. In order to safeguard users from potential impersonation of these pages, a list of frequently attacked pages (with screenshots and domain names) is maintained using Visual Similarity methods. The content of a page is compared to the trusted ones if a user accesses one that is not on the trusted list. A high degree of visual similarity indicates that the page is phishing because it appears to be one of the reliable pages. This technique's drawback is that it solely takes into account visual similarities [5]. This problem statement of the discussion aims to present an overview of up-to-date research endeavors that have explored the various definitions and fundamental concepts related to steganography, in addition to undertaking an examination of the diverse forms of steganography that are commonly. Furthermore, it delves into the researches that spectrum of attacks to which steganography is susceptible, tracing the emergence of steganography and cryptography as direct responses to the escalating tide of attacks and threats targeting smartphone technologies. Table 1 presents a summarized of Previous Studies.

3. Concept of steganography and cryptology

Steganography necessitates numerous indispensable characteristics in order to efficiently obscure information within an image. An elemental prerequisite is the un noticeability of the steganography component to the human visual perception. Factors such as file format independence, unsuspecting

files, payload capacity, resilience against attacks, and concealment contribute to making steganography distinctive. The primary goal is to ensure that an image appears normal and doesn't indicate the presence of concealed information, preserving the integrity of the steganography algorithm. Payload efficiency, as defined in [22], pertains to the capacity of an image to contain a substantial amount of information while minimizing any substantial deterioration in its overall quality. This efficiency enables effective concealment without noticeable alterations. Additionally, the independence of the file format allows for increased versatility, while resilience against image distortions, such as rotation or cropping, ensures that the steganography content remains intact. However, the repeated use of image steganography on the internet has led to various attacks, including graphical and statistical attacks employing steganography [23]. Researchers continually devise novel methods to counter these assaults, aiming to enhance the security of steganography techniques against such attacks.

Cryptography encompasses a broader scope than solely encryption; it entails the examination of secure methods of data communication in the face of possible adversaries, as exemplified by [24]. The principles of cryptology and cryptanalysis are interlinked within cryptography, where cryptanalysis involves deciphering hidden messages—a parallel to steganalysis in steganography.

Encryption, an intrinsic component of cryptography, encompasses the transformation of plain text into an incomprehensible configuration, which is commonly referred to as a cipher. Encryption, using keys to encode and decode, is an elementary method to ensure cryptographic data security. The four main objectives of cryptography data integrity, confidentiality, authentication, and non-repudiation of data are vital components. Confidentiality guarantees that solely the designated addressee possesses the capability to peruse the communication, while authentication ascertains the sender's trustworthiness. Data integrity is an essential component that helps prevent any unwanted changes to the information being sent, giving the recipients the ability to confirm and authenticate the accuracy and integrity of the message being sent during its journey.

4. Cryptography

One of the most important aspects of information security is cryptography.

In the past, encryption was the mainstay of cryptography, converting data into an unintelligible the Greek terms "kryptos" and "logos," which signify the idea of a hidden word, are the source of this name.

Structure to protect private correspondence. However, in today's information-centric economy, its purpose has evolved beyond its original application in protecting diplomatic and military correspondence. Encryption now serves various purposes, extending beyond privacy concerns. These purposes encompass ensuring message integrity, authentication, and non-repudiation of transfers.

Presently, the term "cryptography" encompasses a broad spectrum of methods and applications that protect both data in transit and data at rest.

4.1. Advanced encryption standard algorithm

The escalating dependence on the internet underscores the essential need for continuous safeguarding of data and information. The integrity of communication is predominantly contingent upon encryption techniques, particularly the Advanced Encryption Standard (AES). AES stands out as one of the most extensively employed encryption algorithms, having been officially recognized as a Federal Information Processing Standard (FIPS) in 2011 following a rigorous competition organized by the National Standards and Technology. [25].

For developers seeking to construct highly secure Android applications, the importance of security is equally significant as that of performance. In order to bolster the security of their content, message-based Android applications must utilize cryptographic algorithms such as AES. Given that Android devices are purposely designed with limited resources, optimizing application performance on this platform becomes a crucial consideration. AES exhibits remarkable resistance against traditional attacks such as differential and linear assaults, which employ statistical analysis for password cracking. Noteworthy characteristics of AES include its resistance against known password analysis, adaptability to diverse hardware and software contexts, suitability for hash functions, applicability to devices requiring swift key agility, and effectiveness in stream ciphers These attributes make AES a solid choice for enhancing security in Android apps, specifically prioritize secure message delivery.

The primary technique AES is now widely recognized as used in both the public and private sectors to prevent hackers from obtaining illegal access to private data. Encryption and decryption are fundamental building blocks of data protection techniques against malicious operations. With its robust security features and dependability, AES stands out from the various algorithms designed expressly to encrypt

and decode sensitive data. The Advanced Encryption Standard (AES) has many benefits over previous encryption techniques, which contribute to widespread acceptance.

The implementation: AES encryption is simpler to understand and use for those who deal with encryption systems.

The effectiveness: AES performs best in general since it can support faster encryption and decryption times and requires less memory and system resources.

The flexibility: If more security is required, effortlessly paired with existing security measures to fortify data protection.

The security: Security: AES is widely for its high degree of security and also can be applied to software and hardware to strongly safeguard sensitive data.

The key's length: larger key sizes of AES (128, 192, and 256 bits) strongly offer powerful defiance against hacking attempts. Among its numerous uses are secure data storage, wireless communications, financial transactions, and e-commerce. It is shown that the Open Standard and Adaptability concept can be used for any particular purpose in commercial, private, or public situations.

5. Access control by MAC

Manufacturers assign alphanumeric addresses that serve as unique identifiers, similar to digital fingerprints, for categorizing networked equipment. Access control uses MAC address filtering, a complex procedure. This advanced security solution allows network administrators to create whitelists, restricting access to approved devices [26].

Careful curtain reduces the risk of network breaches and improves resilience through strict access rules.

The authentication paradigm leverages MAC addresses' inherent uniqueness for application access.

Integrating these identifiers with authentication protocols adds immutable credentials to traditional authentication methods. It primarily strengthens network security by blocking attempts by unauthorized users to get access. It is difficult to obtain network or application access without a corresponding permitted MAC address, making circumvention more difficult—even in cases where credentials have been hacked. Safeguards against safety. Moreover, complex device permission management is made easier with MAC-based access control. MAC address specifications give administrators precise control over which programs or resources can be accessed. This meticulous approach to management is very helpful in settings where strict access control is necessary, such as business networks and extremely sensitive data enclaves. Access control methods that use MAC addresses improve security postures, boost network resilience, and provide a flexible foundation for cautious.

6. Secure data application

The software known as "Secure Data App" is well-crafted and thoroughly tested programmer that utilises the programming languages Flutter and Dart in conjunction with PHP-based interfaces that are closely linked to MySQL databases. The "Secure Data App" is a major advancement in data security, usability a cooperation. Notable improvements include end -to-end encryption, two-factor authentication, sophisticated access controls, a range of collaborative features. The application's security posture is greatly enhanced by the implementation of robust access controls, real-time activity monitoring, audit logs, and state-of-the-art data loss prevention techniques. Additionally, application features a secure file transfer feature that guarantees that only the sender and receiver may access the secret information by limiting access to a single device.

7. Use-case diagram of the system

A use case diagram offers a visual representation of the interactions between users and a system. In the context of a secure data system, the primary actors typically involve users and the system itself, as depicted in Fig. 1. Here is an overview of key use cases typically illustrated in a secure data system.

Upload File: This use case involves a user uploading a file to the system. It encompasses the selection of a file from the user's device and its transfer to the system for encryption before sharing.

Download File: In this particular scenario, a user obtains a file from the system. This action encompasses the user's act of choosing a file from the system and commencing the download procedure, which is subsequently trailed by decrypting the file in order to access and peruse its contents. This process ensures that the user can securely view the information contained within the file while maintaining the integrity and confidentiality of the data. Additionally, the user must ensure that they have the appropriate permissions and software to handle the decrypted file, facilitating a smooth and efficient workflow in accessing the necessary information.

File Sharing: This particular use case entails a user engaging in the act of sharing a file with either another individual user or a collective group of users. The user selects the file and specifies the recipients

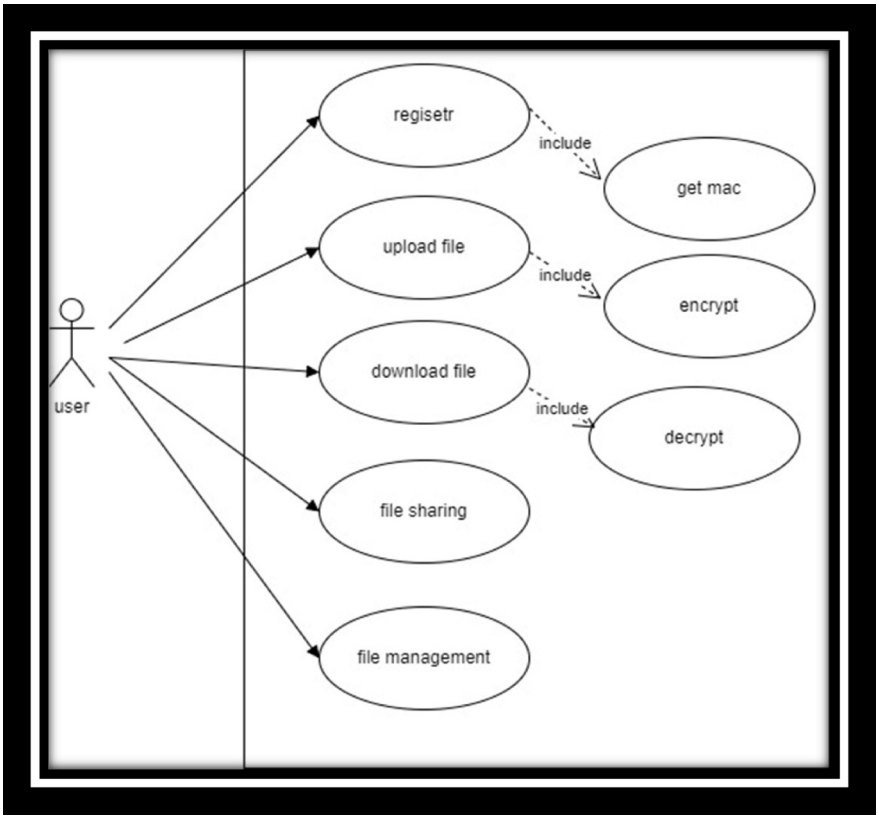

Fig. 1. Use-Case Diagram secure system.

or groups they wish to share the file with. The system facilitates the sharing process.

File Management: This use case includes actions related to managing files within the system It encompasses features such as the arrangement of files into directories, alteration of file names, removal of files, and the exploration for particular files.

User Registration: This use case encompasses obtaining user information, including email, password, and MAC address, for registration purposes.

8. Class diagram

The class diagram delineating secure data encompasses several key classes, depicted in Fig. 2.

File Class: This category encompasses the characteristics and operations associated with discrete files within the overarching framework. It embodies files that necessitate encryption before being disseminated.

User Class: This class embodies the individuals engaging in communication with the system; it is these individuals who possess the capability to execute operations pertaining to file encryption and sharing.

File Manager Class: This class assumes the role of a central component within the system; it orchestrates and supervises operations that revolve around files, ensuring that files are managed and manipulated seamlessly.

Encryption Class: This class is specifically dedicated to functions that are associated with encryption; it supervises and manages the encryption operations that are indispensable for ensuring file security within the system.

Account Class: This class represents the entities that engage in interaction with the system, and it signifies the individuals who possess the competence to execute operations pertaining to file encryption and sharing.

9. Activity diagram

Unified Modeling Language (UML), is an activity diagram includes the arrangement of tasks, their sequential execution, instances of decision-making, as well as any concurrent or parallel undertakings occurring within the system under examination, are all systematically presented. This diagrammatic approach serves to enhance comprehension of system behavior and processes, by facilitating the grasp of the operational flow and interrelationships among activities.

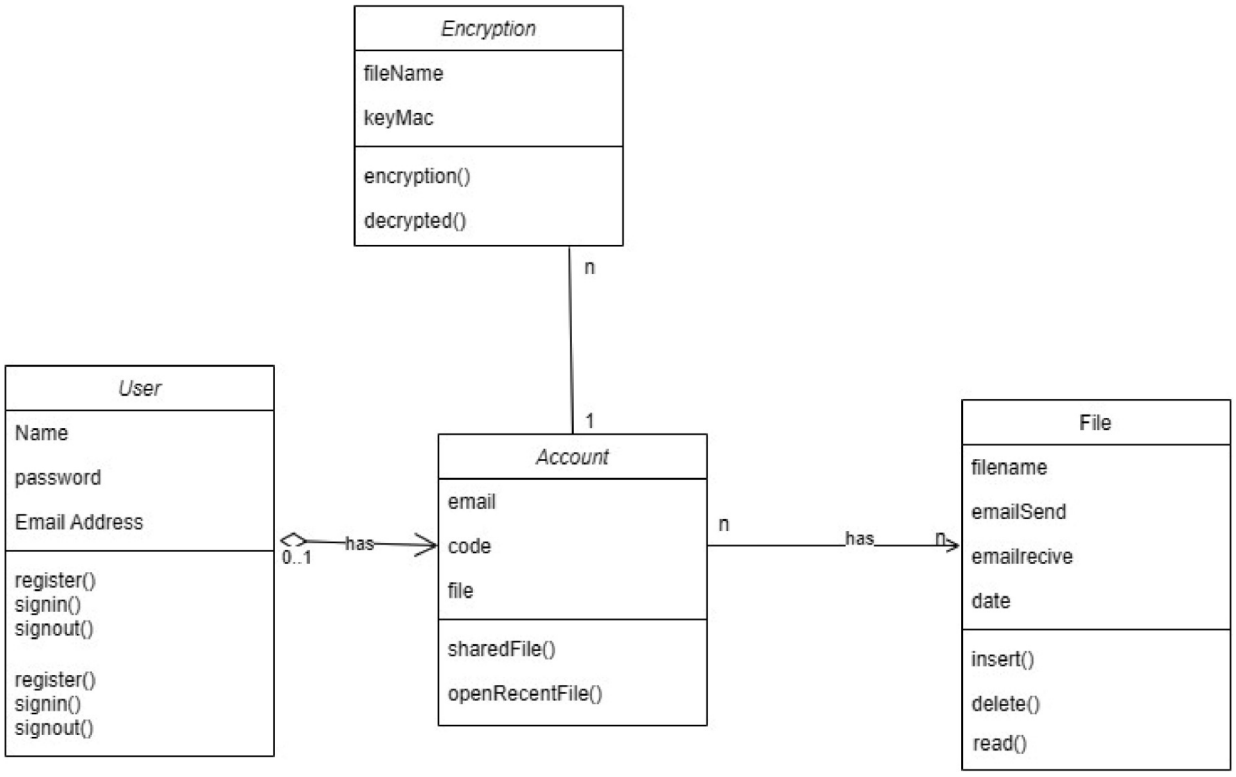

Fig. 2. Class Diagram.

9.1. Sign in the application

The login process is shown in the Fig. 3, using email and password and secret code whether the user have account or login for the first time, After the inputted information has been successfully validated and it is confirmed that the login attempt is originating from the device that has been previously registered, access to the secure data system will be granted.

9.2. Shared file

To initiate file sharing, the user navigates to the 'share' option within the application. Upon selection, the user specifies the file's name and chooses the file intended for sharing. Once the 'share' action is confirmed, the application proceeds to encrypt the selected file and transmits it securely to the database, as delineated in Fig. 4.

9.3. Manage files

The application's file operations, illustrated in Fig. 5, encompass the following functionalities:

Share File: Users can designate a file for sharing and specify the recipient by providing their email address.

Delete File: Enables users to remove selected files from the application.

Show Emails: Allows users to view a list of emails to whom a specific file was sent.

10. Functional requirements testing

The process of confirming whether a system or piece of software meets with the stated functional requirements is known as functional requirements testing [27]. System testing, as used in this study, confirms that the features, actions, and operations of the system work as intended.

In order to verify the system's accurate response to a variety of inputs, user interactions, and anticipated outputs, testing include developing and running test cases. Making sure the system successfully satisfies the functional expectations given in the requirements is the main goal [28]. Testing for functional requirements provides numerous advantages.

Validation guarantees: that the system carries out the intended functions, meets the expectations of users and stakeholders, and is implemented with precision.

Error detection: the technique reduces the problems occurring in the production by specifically examining the functional requirements to identify

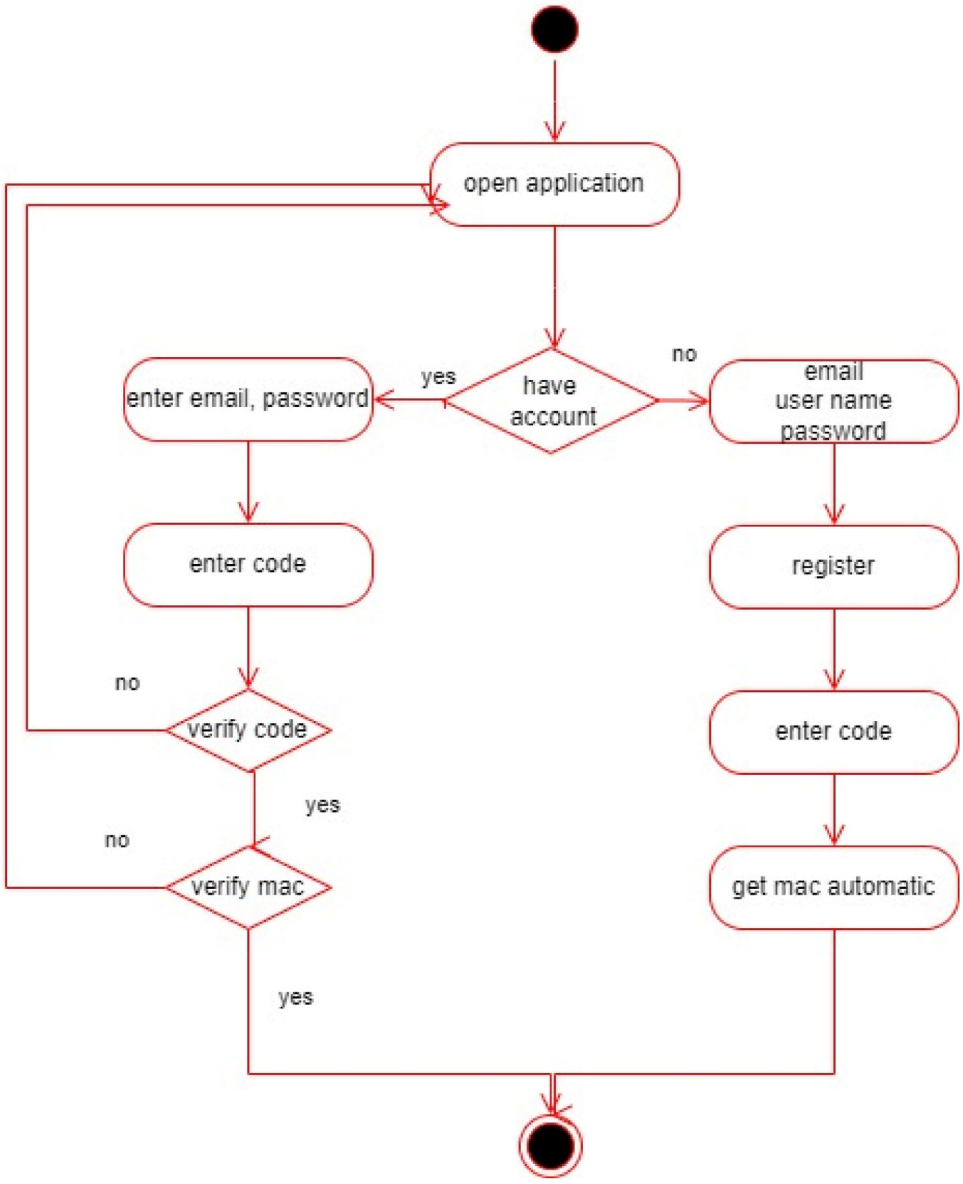

Fig. 3. Sign in The Application – Activity Diagram.

and resolve behavioral flaws or difficulties in the system early in the development cycle.

User testing: verifying that all planned features and functionalities operate as intended, testing for functional requirements ensure the ideal user experience.

11. Non-functional requirements testing

Non-functional requirements are a collection of rules that define the capabilities and boundaries of a system to enhance its functionality. Examine of non-functional requirements has several benefits [29].

Compliance: Verifying compliance with industry standards, regulatory requirements, and security rules is made easier by assessing non-functional criteria. It makes identifying and resolving any compliance-related issues easier.

Risk Mitigation: Possible risks and vulnerabilities can be identified and addressed early in the development process by evaluating the demands of non-functional needs. As a result, the probability of running into issues with usability, performance, or security is reduced. [30].

Compliance: Verifying compliance with industry standards, regulatory requirements, and security rules can be aided by testing against non-functional

Fig. 4. Shared File – Activity Diagram.

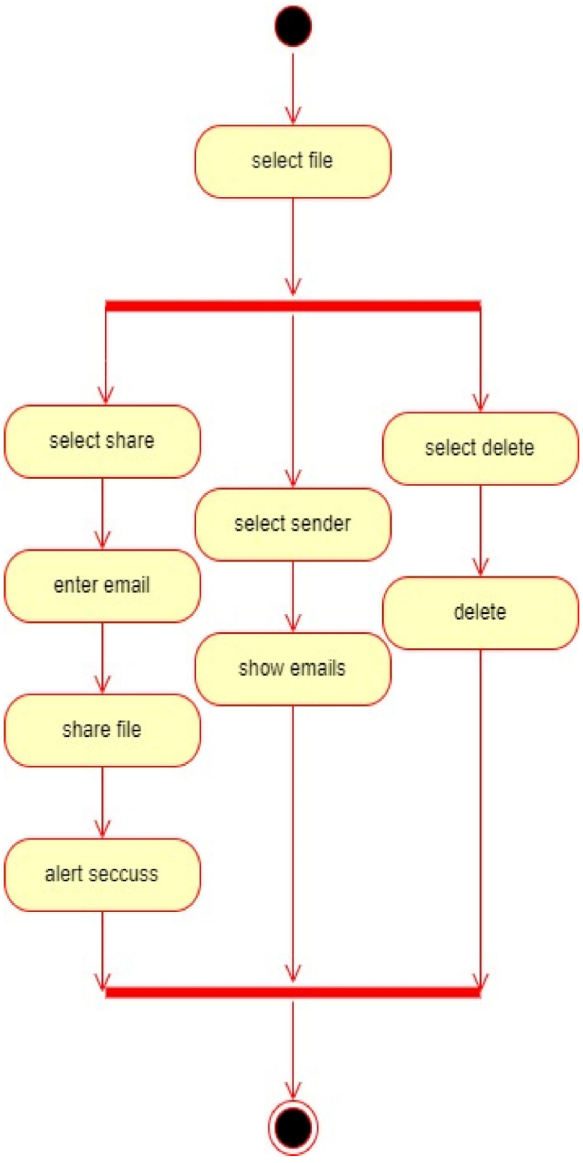

Fig. 5. Manage Files – Activity Diagram.

criteria. It helps find and fix any problems pertaining to compliance.

Risk Mitigation: Potential risks and vulnerabilities can be identified early in the development cycle and addressed by evaluating non-functional demands. There are consequently fewer opportunities for usability issues, performance hiccups, or security breaches. [31].

System Optimization: Prior to deployment, system performance, security, or usability may be enhanced by identifying areas that call for testing non-functional needs. This allows for optimization and refinement.

This section contains proof that the application satisfies non-functional requirements

NFR01: Performance testing includes evaluating the system's response time, scalability, throughput, and resource consumption under various load scenarios to make sure it meets predicted performance requirements. The programme satisfies this requirement since it responds to user actions promptly.

NFR02: Reliability Testing considers error control, recovery strategies, and fault tolerance as it assesses the system's ability to function consistently and dependably over an extended period of time. This

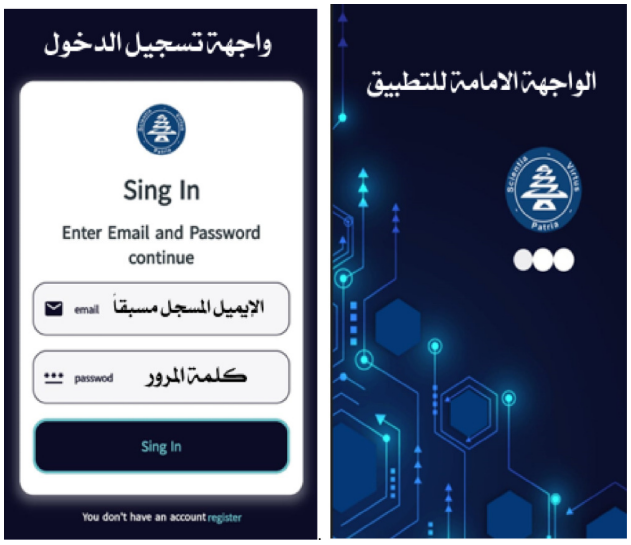

Fig. 6. Initial Application Access.

condition is met by the program's effective handling of a heavy user load.

NFR0: Security testing evaluates the effectiveness of the system's encryption, authorization, and authentication in protecting sensitive data and minimizing vulnerabilities. The application meets this need by encrypting passwords, shared files, and user data.

NFR04: Part of the process of usability testing is evaluating a system's user interface, navigation, and overall user experience to ensure that it meets established usability criteria. The programmer complies by providing a user-friendly interface and simple navigation.

12. Implementation

12.1. Initial application access

When the programmer is first launched, new users must finish the registration process in order to use its different features. On the other hand, existing users possess the ability to promptly log in by furnishing their password and the email address that is associated with their registered account. Additionally, the system requires a confidential code to ascertain the legitimacy of the accounts. This secure login procedure enables users to seamlessly exchange and receive files within the program, as depicted in Fig. 6.

12.2. Registration process

When an individual who is unfamiliar with the system opts for the registration alternative, a window dedicated to registration appears on the screen. Facilitation of the data input process, such as the individual's name, email address, and password, is made more convenient with the assistance of this particular window, as shown in Fig. 7. Subsequently, the system presents a window where the individual may input the confidential code that is required to verify the authenticity of their account subsequent to the provision of the aforementioned personal details.

12.3. Sending encrypted files

Clicking on the "Send" button initiates the commencement of the procedure for sharing files in a format that ensures encryption. This process activates the appearance of a window that provides the opportunity to select files from the device. Subsequently, opt for the option to "View Document" in order to peruse and select the required file. Finally, conclude this process by clicking on the "Add File" button, as depicted in Fig. 8.

12.4. Sharing a file

In order to begin file sharing, click the share icon that is conveniently situated beneath the file that is to be shared. Next, once the target recipient's email address has been entered, you need to click on the "Add File" option.

13. UI testing

Software evaluation must include UI testing to guarantee the strength and effectiveness of an application's user interface. This type of testing includes a thorough assessment of the visual components, layout, interactions, and overall user experience in order to determine the functioning and user satisfaction of the interface [32]. The following crucial duties are included in the entire nature of user interface testing [33]:

User interface element verification involves making sure that interactive elements like buttons, checkboxes, dropdown menus, and text fields are accurate and presented in the right way.

Layout validation: is the process of assessing the placement and organization of user interface elements with respect to a variety of screen sizes and design criteria.

Analyzing the application's navigation: This entails confirming that users can easily navigate between menus, tabs, and other navigational elements.

Input validation: is the process of carefully reviewing input fields to ensure that user input is processed accurately and that relevant feedback is sent.

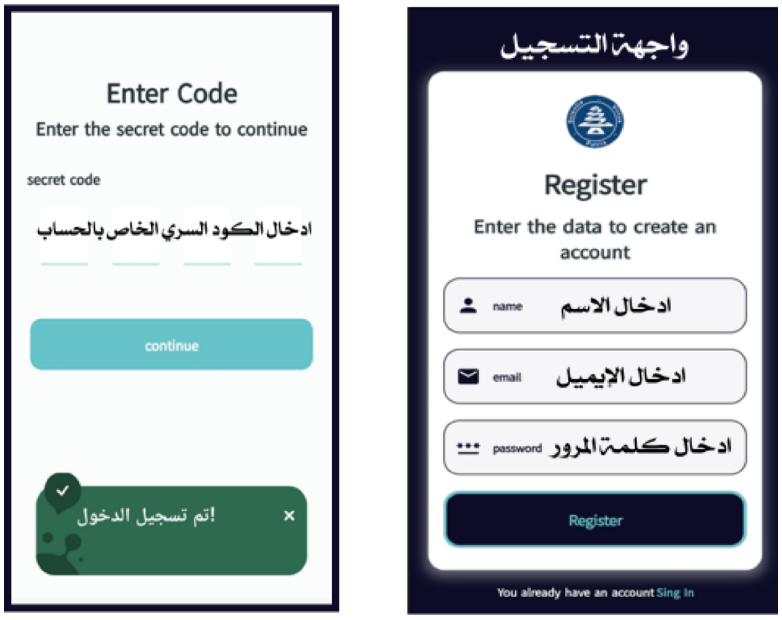

Fig. 7. Registration Process.

Fig. 8. Registration Process.

Evaluation of usability: The process of assessing an interface's degree of usability, intuitiveness, and overall user experience while taking consistency, readability, and adaptability into account.

Compatibility analysis comprises assessing an interface's performance on a range of platforms, such as different operating systems, browsers, and devices, while also making sure that its functionality and aesthetics are consistently maintained.

A multitude of crucial software development domains benefit greatly from UI testing [34].

User pleasure is increased by fixing any issues that could negatively impact the user experience and by ensuring that the interface is visually appealing, easy to use, and intuitive.

Usability Improvement: By assessing user engagement and simplicity of navigation, UI testing helps to ensure that user interactions run smoothly and reduces user dissatisfaction.

Brand reputation: A well-designed user interface demonstrates professionalism and dependability, which improves the brand's reputation in general.

Retaining consistency in design elements and layout patterns promotes comprehension, reduces ambiguity, and enhances user experience.

Error identification: Prompt identification of visual irregularities or design defects enables prompt remediation, hence reducing unfavorable user experiences.

Cross-Platform compatibility: is essential to guaranteeing a smooth user experience by providing uniform functionality and appearance across various platforms, browsers, and devices.

14. Evaluation of the application

The preceding sections have addressed the methodologies employed to assess and verify the application's functionalities from the developer's perspective. In this section, we delve into an evaluation of the application from the user's standpoint in Fig. 9

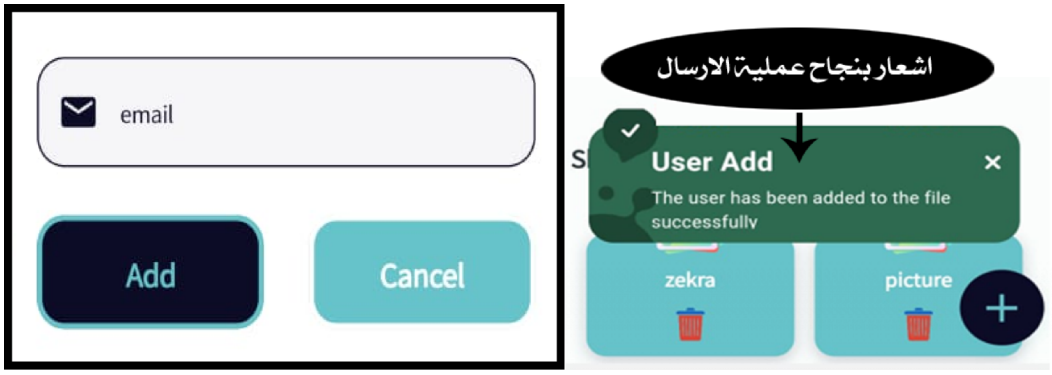

Fig. 9. Sharing a File.

showing the sharing a file. Expert-driven evaluation methods encompass approaches to assess usability that rely on the proficiency and acumen of usability experts or evaluators. A procedure for expert-based examination was implemented to ascertain whether the application adhered to the stipulated requirements. The subsequent techniques were employed:

Heuristic Analysis: is a reference standard aimed at evaluating the user interface. With this approach, the interface is systematically assessed using a predetermined set of usability norms or principles also known as heuristics to determine its suitability.

Evaluation via Cognitive Walkthrough: Using pre-existing user activities, evaluators replicate user interactions with the user interface design through the use of cognitive walkthrough evaluation. Evaluators take note of potential usability issues and evaluate the ease of learning and simplicity of use of the interface. To enhance evaluators' comprehension, the Cognitive Walkthrough assessment aimed to elucidate the requisite procedures for each function within the program. The subsequent table succinctly outlines the procedural elements designed to enhance the clarity of usability.

Task	Steps
Register	1. Click on "Register" link in the start interface. 2. Fill out the registration form.3. Click on "Register" button.
Sign in	1. Click on "Sign in" link in the start interface. 2. Fill out the login form. 3. Click on "Sign in" button.
Add a file	1. Click on "Plus" icon in the main interface. 2. Fill out the file's information (name and comment). 3. Upload the file from the user's device. 4. Click on "Add file" button.
Enter secret code	1. Type four numbers as a secret code. 2. Click on "Continue" button.
Share a file	1. Click on "Share" icon in the main interface. 2. Type the email of the user that will share the file. 3. Click on "Add" button.
Download a file	1. Click on any file in "Shared Files" section in the main interface.
Delete a file	1. Click on "Delete" icon in the main interface.
Manage files access	1. Click on "Users" icon at the button of any file in "My files" section.
Log out	1. Click on "Log out" icon in the main interface.

After conducting comprehensive testing of the user interface and ensuring the successful execution of all functional requirements, the programmed presented no problems to the reviewers.

The essential features of the programmer and offered insightful analysis on a range of topics, emphasizing the critical need of safe data management.

The expert panel duly acknowledged

The application features an intuitive user interface that makes it easier for users to navigate and access features.

The design's color scheme makes it simple to view, which enhances the user experience in general.

Users can easily navigate the file transfer procedure, which ensures a quick and effective interchange of data, whether it is being uploaded or downloaded.

The application's strong and dependable data functionality, in particular, greatly increases its value by guaranteeing the shared data's integrity and secrecy, which is crucial.

Essential for protecting sensitive data's privacy during transfer and preventing any unwanted access.

This noteworthy feature is that the examiners' extremely insightful feedback emphasis's how crucial it is to put strong data security procedures in place.

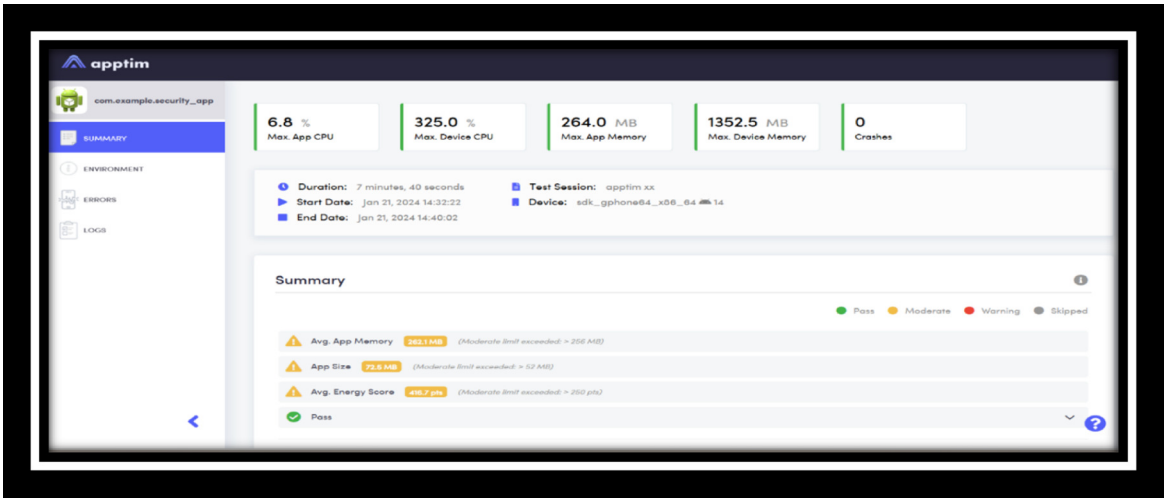

Fig. 10. Resources of the application.

Additionally, it emphasis's the importance of and demonstrates how the programmer may provide safe data processing concealing and safeguarding information, thereby accentuating its value. Throughout the evaluation process, the application performed exceptionally well, showcasing its exceptional capabilities and reliability.

15. Performance analysis

To ensure maximum effectiveness and user pleasure, a thorough review of the Android application must evaluate a number of factors. We performed an analysis of the application's performance, paying particular attention to how it used resources (memory and processing). Furthermore, we assessed the application's security features.

15.1. Resources of the application

The App tool is used for analyzing the performance of the application. Results of analysis shown in the Fig. 10.

1. **Maximum Central Processing Unit (CPU) Usage:** The value indicates that the application utilizes the CPU at a rate of 6.8%. This means that the application consumes a small portion of the processor's capabilities. he assessment of this metric is typically determined by considering the level of the application's promptness and its influence on the overall functioning of the system. The low value of this percentage indicates efficient and lightweight performance on the processor.
2. **Maximum Application Memory**: The value (264 megabytes) represents the maximum amount of system memory the application can consume.

15.2. Security of the application

The process of hiding and protecting information depends mainly on the AES encryption algorithm. In order to test this algorithm, we uploaded a PDF file of 443 KB to the application. The application encrypts this file to obtain an encrypted file of 605 KB. Encryption ratio can be calculated using Eq. (1)

$$\text{Encryption Ratio} = (\text{Size of Encrypted File}/\text{Size of Original File}) \times 100 \quad (1)$$

In this case:

$$\text{Original Size} = 605\ \text{KB}$$

$$\text{Encrypted Size} = 443\ \text{KB}$$

$$= (605\ \text{KB}/443\ \text{KB}) \times 100$$

$$D = (443\ \text{KB}/605\ \text{KB}) \times 100$$

Calculating this gives: $\approx 1.365 \times 100 \approx 136.5\%$

$$D \approx 1.365 \times 100 \approx 136.5\%$$

This indicates that the size of the file after encryption is 136.5% of the size of the original file, which is an increase of 36.5%. This means that the encryption process added 162 kilobytes of additional data to the file. The additional data is what makes the encrypted file more secure.

16. Discussion results

The thorough analysis and evaluation of the Android application, with particular focus on its functionality, security features, and user interface, has shown positive results. The panel of specialists' thorough analysis confirmed the application's high degree of data security and highlighted several important features, including its user-friendly layout. Together, these qualities improve user experience while ensuring data integrity and confidentiality—two essential factors in the field of data security. Furthermore, the performance analysis performed with the App tool produced insightful findings about resource allocation and encryption effectiveness. With only a little amount of RAM available to it and little use of the central processor unit, the application demonstrated efficient resource utilisation. These results imply that the programme operates effectively and makes the best use of its resources. Moreover, the encryption process—which is based mostly on the AES algorithm—showed a noteworthy encryption ratio and a corresponding rise in file size following encryption, strengthening the data's security. An analysis of the programming techniques employed highlights the meticulous approach taken to conceal sensitive information, ensure robust security protocols, and establish safe channels of communication. The application's structure reinforces its defense's by using advanced encryption techniques, safe key management protocols, and stringent access controls.

17. Conclusion

The principal objective of the study was to examine efficient methodologies for safeguarding the secrecy and soundness of information, with a particular focus on developing a tailored Android application. This programmer was Meticulously created to conceal critical information, ensure the highest level of security, and offer secure Communication channels. The system architecture was created by combining both strong privacy and security in the characteristics with up-to-date methods such as encryption, safe key management, information hiding, and strict Access limits.

Programming process and inaccessible to unauthorized parties. Reinforces the application's defensive wall in conjunction with encryption, ensuring that sensitive data may only be Limitations on access and two-factor Authentication accessed by authorized users. Through a range of techniques, like data obfuscation, pseud anymization, and restricted data retention laws, the programmer values user privacy. Data saved in the AES secret form is more secure. applications' repositories and safeguards it while it's being transmitted, supporting a complete privacy framework. The 'Secure Data App' boosts security by utilizing device-level authentication processes and skillfully using the device's Mac Address as an extra unique identifier on top of the preexisting text. The programmer also makes use of the Advanced Encryption Standard (AES) technology, a vital component of contemporary encryption that guarantees strong data privacy across storage devices and communication channels.

Funding

This research was supported by (Computer and Communication Engineering Islamic University of Lebanon, Lebanon and Computer Science Department, College of Science, Mustansiriyah University, Baghdad, Iraq), The authors would like to express their gratitude for the financial support, which enabled the development and testing of the 'Secure Data Applications' project. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agency.

Acknowledgement

The researcher would be thankful the Islamic University of Lebanon and Department of Computer Science / college of science / Mustansiriyah University for supporting this research.

18. Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this research. All financial and material support for this study was disclosed, and no competing interests exist that could influence the findings or interpretations presented in this paper.

References

1. M. M. Kuyucu, "Mobile media as a digital communication tool. New searches and studies in social and humanities sciences," 31.
2. S. N. Abd, M. Alsajri, and H. R. Ibraheem, "Rao-SVM machine learning algorithm for intrusion detection system," *Iraqi Journal For Computer Science and Mathematics*, vol. 1, no. 1, pp. 23–27, 2020.
3. K. Cabaj, L. Caviglione, W. Mazurczyk, S. Wendzel, A. Woodward, and S. Zander, "The new threats of information hiding:

The road ahead," *IT professional*, vol. 20, no. 3, pp. 31–39, 2018.
4. V. Sihag, M. Vardhan, and P. Singh, A survey of android application and malware hardening. *Comput. Sci. Rev.*, vol. 39, p. 100365, 2021.
5. L. E. George, "Phishing attacks detection by using artificial neural networks," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 3, pp. 159–166, 2023.
6. T. Rasul, R. Latif, and N. S. M. Jamail, "A computational forensic framework for detection of hidden applications on Android," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 353–360, 2020.
7. K. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7669–7678, 2021. doi: 10.1109/tii.2021.3049141.
8. D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for secure ehrs sharing of mobile cloud based e-health systems," *IEEE access*, vol. 7, pp. 66792–66806, 2019.
9. Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020. doi: 10.1109/tii.2019.2942190.
10. X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.
11. I. Al_Barazanchi, A. Murthy, A. A. Al Rababah, G. Khader, H. R. Abdulshaheed, H. T. Rauf, and Y. Niu, "Blockchain technology-based solutions for IOT security," *Iraqi Journal for Computer Science and Mathematics*, vol. 3, no. 1, pp. 53–63, 2022.
12. I. Almomani, A. Alkhayer, and W. El-Shafai, "A crypto-steganography approach for hiding ransomware within HEVC streams in android IoT devices," *Sensors*, vol. 22, no. 6, p. 2281, 2022. doi: 10.3390/s22062281.
13. R. Kumar *et al.*, "An integration of blockchain and AI for secure data sharing and detection of CT images for the hospitals," *Computerized Medical Imaging and Graphics*, vol. 87, p. 101812, 2021. doi: 10.1016/j.compmedimag.2020.101812.
14. C. Feng *et al.*, "Efficient and secure data sharing for 5G flying drones: A blockchain-enabled approach," *IEEE Network*, vol. 35, no. 1, pp. 130–137, 2021. doi: 10.1109/mnet.011.2000223.
15. K. Pandey, R. Saxena, A. Awasthi, and M. P. Sunil, "Privacy preserved data sharing using blockchain and support vector machine for industrial IOT applications," *Measurement: Sensors*, vol. 29, p. 100891, 2023. doi: 10.1016/j.measen.2023.100891.116.
16. T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," *Applied Sciences*, vol. 10, no. 2, p. 488, 2020.
17. P. B. Prince and S. P. J. Lovesum, "Privacy enforced access control model for secured data handling in cloud-based pervasive health care system," *SN Computer Science*, vol. 1, no. 5, 2020. doi: 10.1007/s42979-020-00246-4.
18. L. Liu, J. Feng, Q. Pei, C. Chen, Y. Ming, B. Shang, and M. Dong, "Blockchain-enabled secure data sharing scheme in mobile-edge computing: An asynchronous advantage actor–critic learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2342–2353, 2020.
19. H. Qiu, M. Qiu, M. Liu, and G. Memmi, "Secure health data sharing for medical cyber-physical systems for the Healthcare 4.0," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 9, pp. 2499–2505, 2020. doi: 10.1109/jbhi.2020.2973467.
20. I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Computers & Security*, vol. 88, p. 101653, 2020.
21. P. Bateman and H. G Schaathun, "Image steganography and steganalysis. Department Of Computing, Faculty of Engineering and Physical Sciences," University of Surrey, Guildford, Surrey, United Kingdom, 4th August, University of Surrey, Guildford, Surrey, United Kingdom, 4th August, 2008.
22. B. Furht, E. Muharemagic, and D. Socek, "An overview of modern cryptography," *Multimedia Encryption and Watermarking*, pp. 31–51, 2005.
23. P. Kumar and S. B. dan Rana, "Development of modified AES algorithm for data security Opt.," *Int. J. Light Electron Opt.*, vol. 127, pp. 2341–5, 2016.
24. J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown. "A study of MAC address randomization in mobile devices and when it fails," *Proceedings on Privacy Enhancing Technologies*, pp. 365–383, 2017.
25. N. Sanchez-Gomez, J. Torres-Valderrama, J. A. Garcia-Garcia, J. J. Gutierrez, and M. J. Escalona, "Model-based software design and testing in blockchain smart contracts: A systematic literature review," *IEEE Access*, vol. 8, pp. 164556–164569, 2020. doi: 10.1109/access.2020.3021502.
26. M. N. A. Khan, A. M. Mirza, R. A. Wagan, M. Shahid, and I. Saleem, "A literature review on software testing techniques for smartphone applications," *Engineering, Technology & Applied Science Research*, vol. 10, no. 6, pp. 6578–6583, 2020.
27. V. Riccio, G. Jahangirova, A. Stocco, N. Humbatova, M. Weiss, and P. Tonella, "Testing machine learning based systems: a systematic mapping," *Empirical Software Engineering*, vol. 25, no. 6, pp. 5193–5254, 2020. doi: 10.1007/s10664-020-09881-0.
28. M. C. Júnior, D. Amalfitano, L. Garcés, A. R. Fasolino, S. A. Andrade, and M. Delamaro, "Dynamic testing techniques of non-functional requirements in mobile apps: A systematic mapping study," *ACM Computing Surveys*, vol. 54, no. 10, pp. 1–38, 2022. doi: 10.1145/3507903.
29. E. Pourabbas, C. Parretti, F. Rolli, and F. Pecoraro, "Entropy-based assessment of nonfunctional requirements in axiomatic design," *IEEE Access*, vol. 9, pp. 156831–156845, 2021. doi: 10.1109/access.2021.3128686.
30. "UI WEB Automation Testing Using Testng LOG4J," *Journal of Xidian University*, vol. 14, no. 7, 2020. doi: 10.37896/jxu14.7/102.
31. E. Alégroth, L. Ardito, R. Coppola, and R. Feldt, "Special issue on new generations of UI testing," *Software Testing, Verification and Reliability*, vol. 31, no. 3, 2021. doi: 10.1002/stvr.1770.
32. M. Cho, "Deep learning-based UI testing automation technology of in-vehicle infotainment," *KIISE Transactions on Computing Practices*, vol. 25, no. 2, pp. 124–129, 2019. doi: 10.5626/ktcp.2019.25.2.124.
33. G. Tambunan and L. M. Ginting. "Comparison of heuristic evaluation and cognitive walkthrough methods in doing usability evaluation of mobile-based del egov centre hospital information system," *Prosiding Seminastika*, vol. 3, no. 1, pp. 99–106, 2021.
34. M. Li, W. Lou, and K. Ren, "Data security and privacy in wireless body area networks," *IEEE Wireless communications*, vol. 17, no. 1, pp. 51–58, 2010.