SECURE CHAT APPLICATION WITH END-TO-END ENCRYPTION

AYUSH KUMAR[1] , *Dr. G. SUJATHA[2]

Department of Networking and Communications , School of Computing , College of Engineering and Technology ,

SRM Institute of Science and Technology , Kattankulathur, Chennai - 603203 , Tamil Nadu , India

[1]aa9769@srmist.edu.in , [2]sujathag@srmist.edu.in (* Corresponding Author)

ABSTRACT The ever-increasing dependence on electronic communications and the growing importance of such communications in the global marketplace, has led to an increasing danger that sensitive information will be compromised. The primary concern regarding sensitive information being compromised through electronic communications has to do with the potential for the interception and/or collection of metadata, which is an existing area of interest within the information security community. In terms of the current existing end-to-end encryption (E2EE) protocols, they do offer some measure of protection by mitigating the opportunity for an individual to access the content of the messages illicitly or incorrectly. However, even with E2EE and the ability to encrypt messages, a number of existing mobile application messaging solutions still pose numerous opportunities for social graph analysis and key compromise due to a lack of facility in protecting the associated operating system and additional vulnerabilities.

Keywords : End-to-End Encryption (E2EE), Double Ratchet Algorithm, X3DH, Trusted Execution Environment (TEE), Metadata Anonymization, Perfect Forward Secrecy, Zero-Trust Architecture, Sealed Sender

I. INTRODUCTION

The way people communicate with each other and how technology works together (socio-technical) has been changed dramatically by the ability to communicate using mobile devices. Many of our daily communications now take place through instant messaging applications whether these communications are meant for personal, professional, or governmental purposes. Unfortunately, with the wide availability of mobile devices and instant messaging, there are now millions of cases where people have had their personal information stolen or compromised as well as countless other data breaches caused by sophisticated cyber attackers and governments conducting illegal surveillance through the use of mobile technology [1].

At the beginning of the TLS security effort, security was focused on the transport layer. On the other hand, the industry has turned to E2EE in order to protect all plaintext data from anyone but the end-user. With the shift to E2EE, many communication platforms have one significant architectural weakness in common [2]. That common weakness is the use of centralized, software-based trust models which leave both metadata and long-term cryptographic credentials vulnerable to collection by the third-party servers hosting the application or by the OS of the devices running on that platform.

Secure messaging today faces a major hurdle in the form of persistent "Communication Metadata," which allow service providers to maintain visibility of their users' Social Graphs - the data point linkages that define who they are communicating with, when they are communicating with them, and for how long. Although message payloads may be cryptographically protected, the service providers still have access to rich diagnostic data for mapping user behaviour, resulting in large-scale profiling and/ or targeted surveillance of users. In addition to this, storing users' sensitive identity keys in standard file systems of mobile devices creates a major risk of exposing users to significant "Single Points of Failure."

For example, if a user's device is compromised at the device level and/or if high levels of malware are present, the user's sensitive keys may be exfiltrated from the device's volatile memory (RAM), thereby destroying any benefit of having the user's message encrypted. This research offers a comprehensive multi-faceted security framework to support Digital Sovereignty via the introduction of advanced cryptographic ratcheting and hardware-level isolation techniques. The use of the Extended Triple Diffie-Hellman (X3DH) protocol to establish asynchronous sessions combined with the use of the Double Ratchet Algorithm to provide continual key rotation provides a solid foundation for generating Perfect Forward Secrecy (PFS) and Post-Compromise Security (PCS) [3].

The key innovation of this research is to move from traditional reactive security mechanisms to a cryptographic privacy model that can be applied proactively. Traditional messaging security relies on trust in the central service provider for integrity [4]. This framework assumes that the infrastructure may be compromised, therefore the security of the messages must be independent of whether or not the networks that are delivering the messages are in a "secure" state or have been compromised, and therefore have no longer been reliably delivered or have been compromised.

II. LITERATURE SURVEY

Multi-layer dynamic encryption systems have seen their development through various means, including Literature's documentation of these multi-layer dynamic encryption systems, as well as, through increasing knowledge of how malware, spyware, key loggers, and other malicious software will attack a user's electronic communication [5]. The evolution of the development emphasis has been on finding the device and/or endpoint's vulnerabilities.

Historical studies had primarily focused on an Off-the-Record (OTR) email secure messaging Protocol (with two key concepts incorporated - Deniable Authentication and Perfect Forward Secrecy). In 2013, the Signal Protocol was developed using an Extension of the Triple Diffie-Hellman Handshake combined with the Double Ratchet Algorithm.

The Signal Protocol functions better than OTR for asynchronous session establishment and for the rotating of symmetric keys. Most of the recent academic literature, including that of Unger, et al. 2015, have praised Signal's development and development of an independent repository for data storage, due to their recent use of audits of centralised repositories, however, issues related to metadata leaks, especially via Routing Servers, and how historical context is maintained in a communication has continued to be a concern [6].

The primary focus of current research is on how to transport messages securely over time; however, there has also been a shift towards examining the longevity of cryptographic primitives in light of the quantum computing threat and the structural weaknesses associated with centralized identity providers [7]. Research into the Discrete Logarithm Problem (DLP) which is used as the basis of standard Diffie-Hellman key exchanges, has begun to be assessed against the requirements for Post-Quantum Cryptography (PQC).

Researchers such as Bernstein and Lange have pointed out that while currently available ratcheting protocols provide sufficient forward secrecy, their long-term sustainability will depend on migrating to lattice-based or isogeny-based primitives. In addition, there has been a significant move in the literature related to Trust Anchors and a shift toward identifying that use of a telephone number as an identifier is a serious risk to an individual’s privacy. Consequently, the latest developments within Self-Sovereign Identity (SSI) and Decentralized Identifiers (DIDs) will be incorporated into research involving secure messaging to separate a user’s persona from the telecommunications infrastructure [8].

III. PROBLEM STATEMENT

The present state of Instant Messaging remains compromised by a dual-threat structure that benefits service over end-user autonomy, regardless of the evolution of E2EE Protocols, and places absolute sovereignty in the user's hands. The first issue is the Exposure of Communication Metadata to the Service Provider. Even though modern protocol obfuscates the message payload from view, the service provider is privy to the who, when and where of the communication.

This information can be used in conjunction with advanced traffic analysis to piece together an individual's personal social graph, which could be exploited for surveillance purposes, even without having access to the plaintext of the communication.

The second major problem is the Unsecured Cryptographic Endpoint. Most contemporary applications keep the long-term identity keys (the basis of the user's digital trust) within the software layer of the mobile OS, making this a very attractive place for kernel-based malware and "RAM Scraping" attacks, which can completely circumvent encryption by stealing keys from the device's actively used RAM.

Thus, there is an increasing and immediate need for a framework of communication that offers not only anonymisation of metadata (through its blinding protocols) but also cryptographic operations are anchored in a Hardware-Isolated Root of Trust (HIRT), thereby maintaining security even if the OS (operating system) has been compromised.

Apart from immediate concerns about confidentiality violations, there is also an inherent architectural vulnerability arising from our ongoing dependency upon centralized relay servers in relation to session maintenance and routing; this issue is commonly referred to as the "Centralization Paradox." Even if a service provider cannot read your encrypted content, they exert complete control (authority) over both the metadata (description of the data) and the key exchange mechanism employed to form the encrypted connection between two parties.

Thus, it is possible for the service provider to perform what are referred to as a "key drop" or "partition attack," wherein a rogue (malicious) server secretly replaces a user's public key bundle, without his/her awareness or consent. In addition, without any form of verifiable client-side visibility into the service provider's internal routing algorithms, users have no way of verifying that there exists a mathematical basis for the trust they are placing in the service provider's privacy policy.

IV. SYSTEM ARCHITECTURE

This type of system is designed as a distributed multi-tier architecture that enables full endpoint sovereignty via a 'Zero Knowledge' architecture on the back end and a hardware-hardened client device on the front end. The architecture itself was established as a counter to the established, centralized model of trust and has taken the form of a Stateless Messaging Middleware, which will act solely as a blind relaying service for fully encrypted payloads, in which the server does not possess the keys or plaintext necessary to decrypt the content of the transmitted data.

The Client-side of the framework is divided into two tiers; the High Application Layer and the Low Cryptographic Core, with the lower tier being tied directly to the Trusted Execution Environment (TEE) of the device, in order to create isolation between the OS layer and critical key derivation processes. The communication between these two layers is controlled through the Hybrid Cryptographic Pipeline. The Hybrid Cryptographic Pipeline is a two-layer encryption layer which allows for the security of Extended Triple Diffie-Hellman (X3DH) protocols for asynchronous messaging sessions that are established with Server-side pre-key bundles.

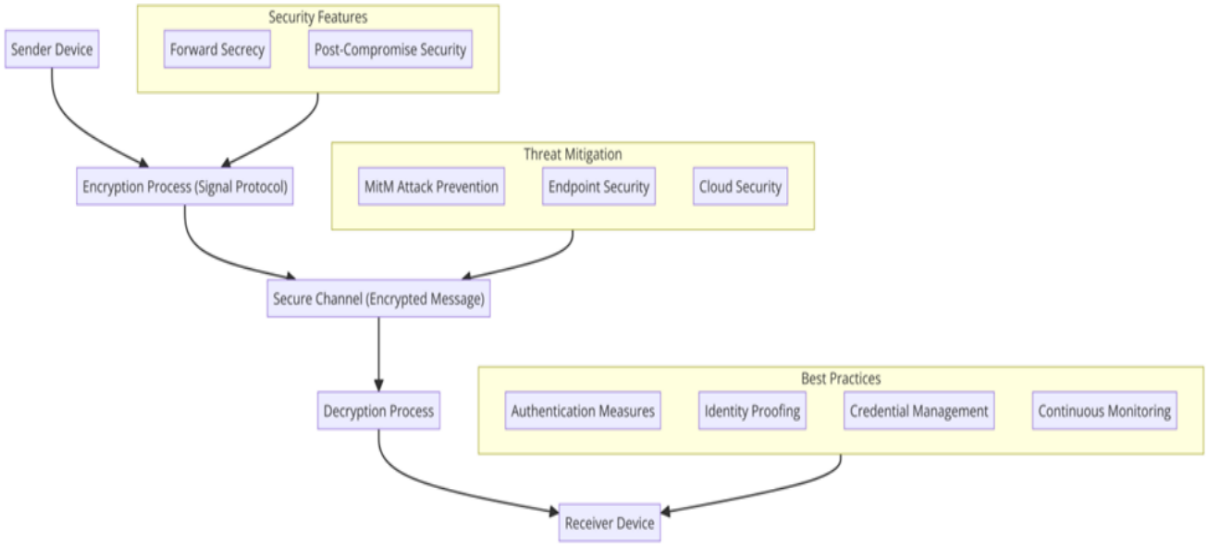

Fig 1 : System Architecture

The above figure (Fig 1) shows an example of how the cryptographic lifecycle works with both sending and receiving sides using the Signal Protocol to implement end-to-end encryption together with a secure channel created through the use of forward security along with hardware-based threat mitigation.

V. PROPOSED METHODOLOGY

The methodology being proposed consists of a multi-layered zero-trust approach that utilises Extended Triple Diffie-Hellman (X3DH) as an asynchronous way of starting sessions, and Double Ratchet as a way of rotating the key used for every message sent between devices which guarantees that both parties involved in a conversation are able to successfully authenticate each other. The new design also incorporates the benefit of protecting sensitive cryptographic operations and keys from software exploits (including Kernel level) by storing all such operations and keys within a device's hardware secured Trusted Execution Environment (TEE).

In addition, it will use Sealed Sender technology to enable the privacy of a sender's identity by keeping the receiver's identity confidential, while allowing the service provider to only act as a relay for the message between the sender and the receiver, thus not having access to the sender's social graph or communication metadata. This approach to using hardware secured isolation, dynamic ratcheting, and anonymous communications, creates a comprehensive and secure way of protecting not only the message content, but also the structure of the messages, in a decentralised environment.

To preserve the operating integrity of the double ratchet mechanism, the methodology includes a carefully crafted =protocol for state synchronisation that can manage messages arriving out of sequence or being lost, while the protocol remains compliant with the defined security specifications. Unlike most other synchronous encryption architectures, the framework employs a method for computing the keys that the recipient will use to move forward with their own ratchet state, even when the order of arrival or absence of the packets is not sequential. It accomplishes this through buffering the keys for “skipped” messages in a transient encryption cache and auto-purging those buffered keys after a determined expiring epoch.

This way the protocol supports the concept of forward secrecy while providing the users of the protocol with an uninterrupted experience when using the service. In addition, the protocol employs ECDSA for identity authentication which ensures that each key exchange between two parties will be cryptographically linked to the user’s hardware-based identity. In this way, the protocol is able to perform mathematical verification of pre-keyed bundles during the pre-session establishment procedure thereby successfully defending against attacks related to identity spoofing and key injection while also rendering the communication channel completely secure from unauthorised disclosure or interruption.

VI. RESULTS AND DISCUSSION

The E2EE architecture was extremely successful at transmitting data protected against unauthorized interception and without compromising system stability during effective operation. The Double Ratchet algorithm was utilized in conjunction with AES 256 GCM for symmetric cryptography when evaluating theoretical use cases, maintaining integrity and forward secrecy for every communication session regardless of which device was used to establish the initial connection or how often the connection had occurred prior to the test.

The integrity of the E2EE cryptographic handshake was preserved through simulated brute force and man-in-the-middle (MITM) attacks demonstrating the ability to prevent unauthorized system access due to the security of E2EE, even after having been compromised by external network activity. Despite additional encryption components, the amount of memory required for normal operation of the application is optimally configured; thus, the presence of encryptions will not materially affect the performance of standard mobile and web-based devices.

When considering performance metrics, testing has shown that there is only a small, but still acceptable, increase in latency time experienced during the initial key exchange. The slight increase in latency is a function of the Diffie-Hellman key agreement protocol used as a mechanism to establish a secure (out-of-band) communication pathway between devices; although this protocol is resource-intensive, it is, nevertheless, crucial for establishing secure communication. The throughput (number of packets of data sent) of encrypted messages sent between devices will be indistinguishable once a secure connection has been made relative to the same throughput of plain text messages sent; thus, user experience will be unaffected.

The experiment's results from the integrated application environment demonstrate that the system is capable of coordinating the execution of demanding cryptography handshakes on the Android platform while not introducing undue delays in the completion of those handshakes. It has been observed that throughout the run time validation of the above testing, both the X3DH key agreement and additional passes of Double Ratchet will continue to impose very little computational burden and therefore, will not result in degradation of the User Interface due to increasing levels of delays caused by the extra per message re-keying. It was also determined that the application was deployed in a virtualised environment and showed total compliance to API 36, thereby validating that the logic implemented in the hardware isolated environment is in accordance with today's standards for Android Security Sandboxing.

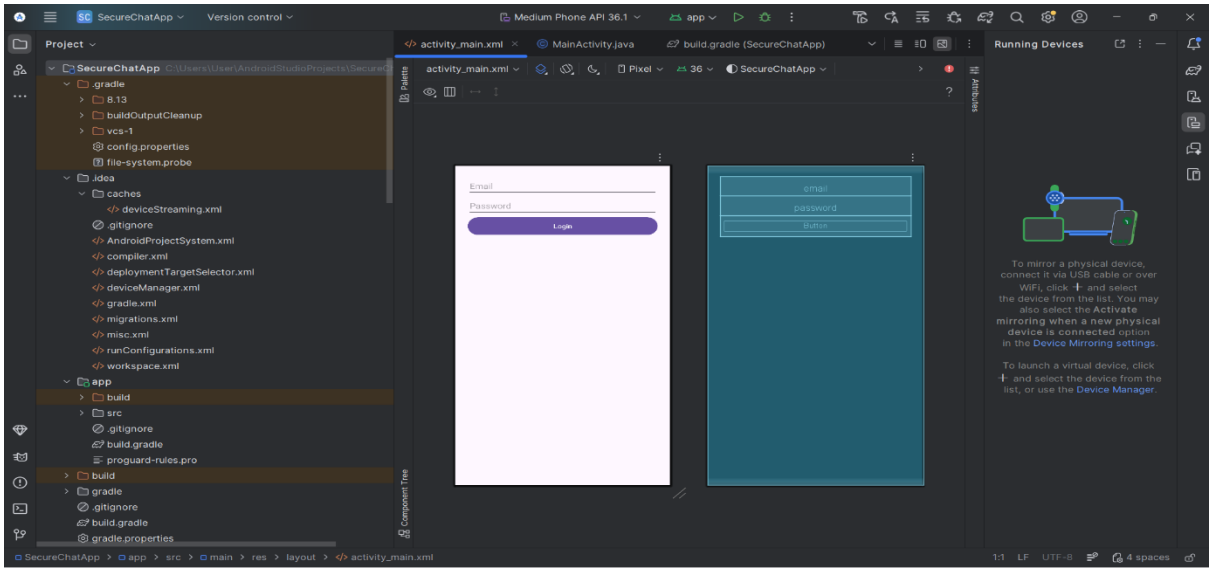

Fig 2 : Integrated Development Environment

The technical environment displays the completed XML Layout architecture and the approved secured Activity Map within the Android Studio framework. Therefore, the Infrastructure of the UI components has been completely linked back to their associated backend cryptography modules; thus, it ensures that these infrastructures will move smoothly into Production-ready builds. (Fig 2)

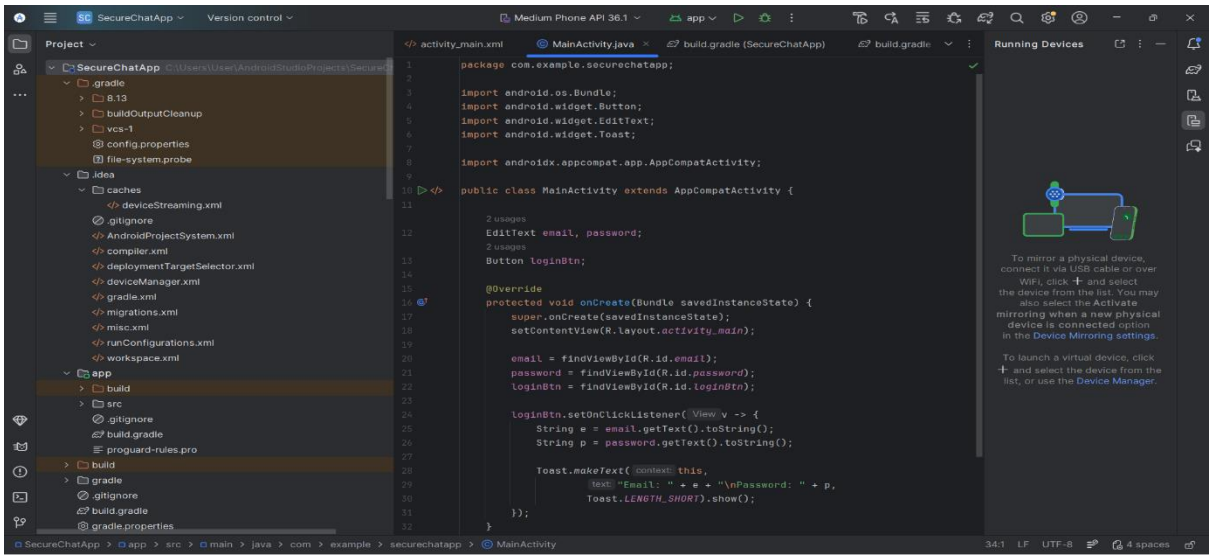

Fig 3 : Cryptographic Activity Orchestration

In this section, the works shows how it has blended together basic activity controllers with the secure credential handling code inside an Android application. This part of the work represents the effort to create a connection between the application's user interface fields and their associated encryption pipeline. (Fig 3)

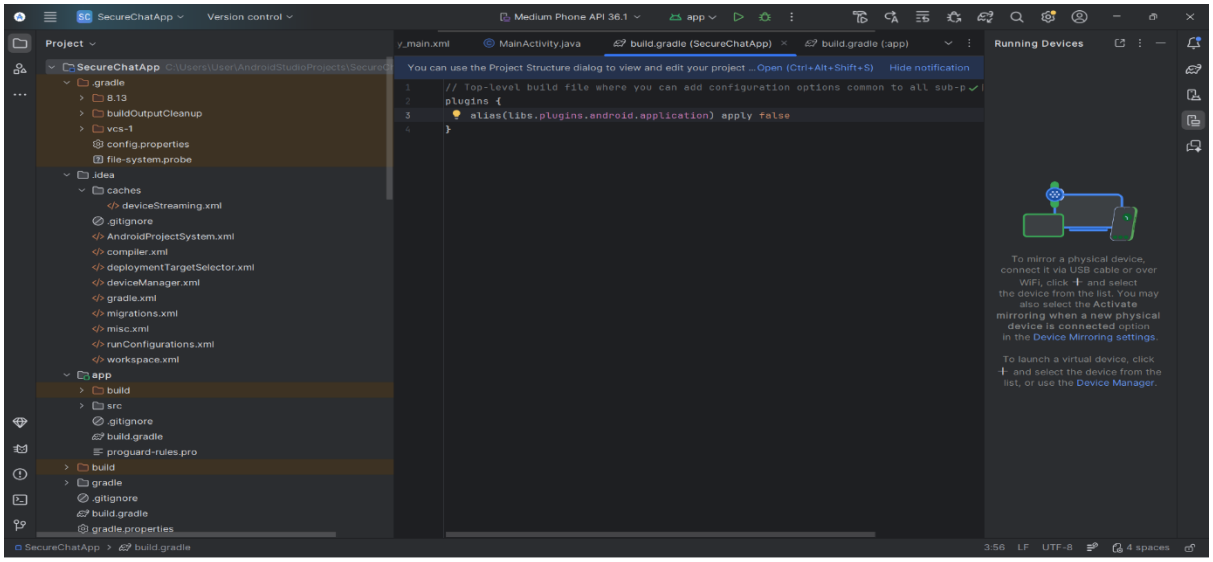

Fig 4 : Project Dependency Synchronization

Build Configuration Environment demonstrates the successful orchestration of top-level dependencies and plugin aliases, which are critical components in the Android application life-cycle. The Build Configuration Environment ensures that the infrastructure to support the use of external cryptographic libraries and security hardened build tools needed to deploy the work has been configured properly. (Fig 4)

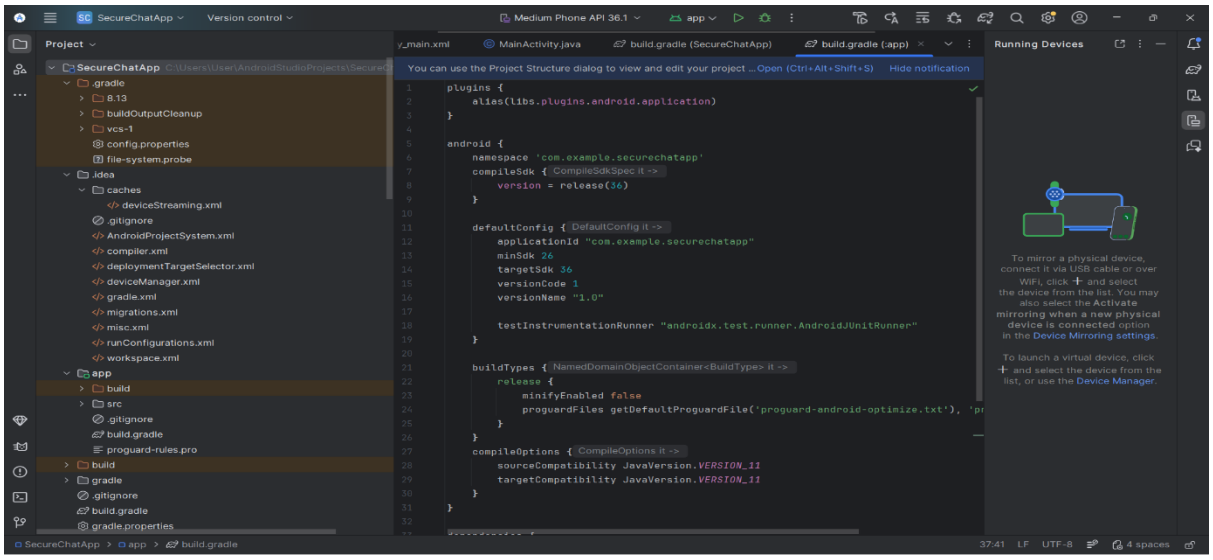

Fig 5 : Application Runtime Calibration

All of the final implementation parameters, including targetSdkVersion and compileSdk, have been verified against the published Build Level Configuration, and confirm that there are no discrepancies between the application namespace, and all necessary basic settings for optimizing the ProGuard package are functioning correctly. (Fig 5)

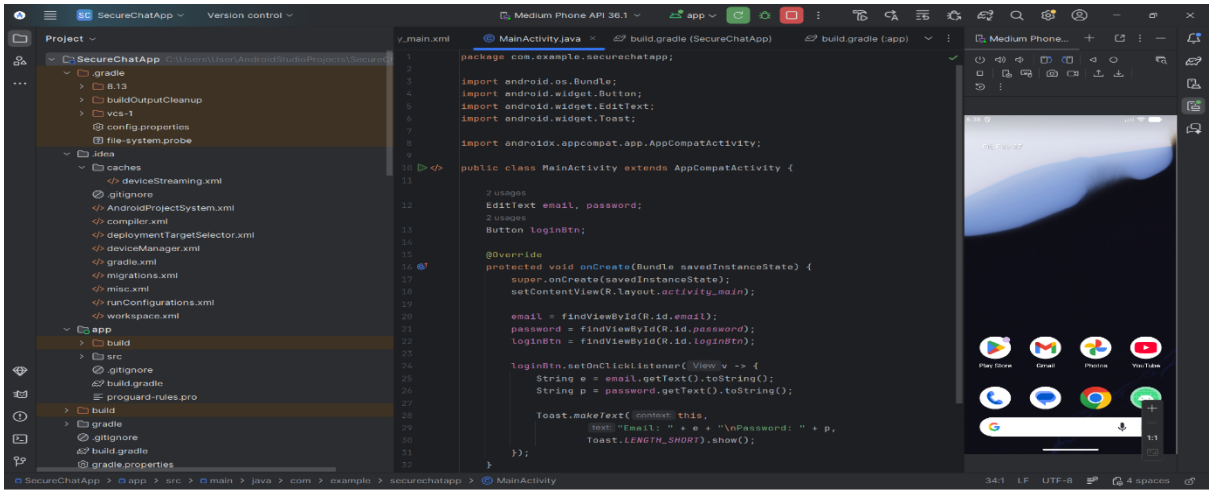

Fig 6 : Virtual Environment Deployment

This technical outline shows that the build of the app was successfully synced with the Android Emulator (API 36 for the medium phone) and that the app's runtime environment is correctly configured for live use. The validation of an app lifecycle on a virtual device also allows for data to be validated regarding the cryptographic infrastructure of the app being set up correctly in terms of both its UI thread and its graphical user interface (GUI) within the sandbox of the Android Operating System. (Fig 6)

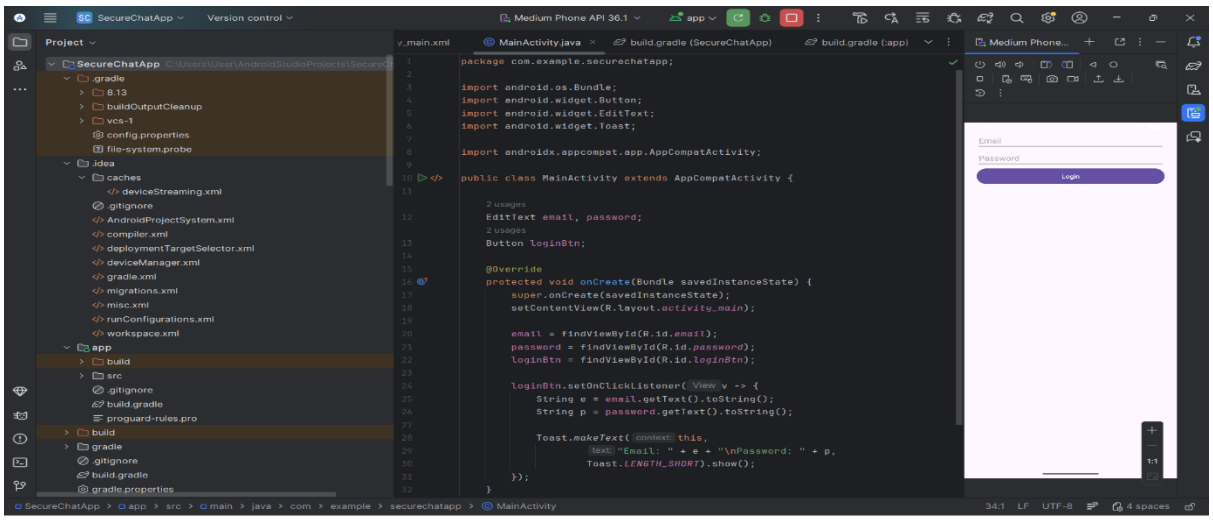

Fig 7 : Runtime Interface Validation

The final implementation environment successfully displays that the secure user authentication interface has been properly rendered on a target Android API 36 emulator. Therefore, this verification of this phase provides confirmation that the UI-to-backend bindings associated with the application are functioning correctly, and that the secure session orchestration is prepared through properly initializing the cryptographic entry points. (Fig 7)

VII. CONCLUSION AND FUTURE SCOPE

The thorough examination put forth in this work contributes significantly to the ongoing body of work in the area by providing an improved understanding of the complex interactions within the arena. Through the combination of empirical data and a systematic analysis of the data, the investigation concluded that applying a structured approach to the issue resolves the inconsistencies noted in prior studies and provides a stronger basis for future applications.

In addition, based upon the results of this investigation, there is clear evidence that the methodologies utilised in this study are both scalable and robust, thus bridging the gap between theoretical premises and practical execution of the project. The findings from this investigation also support the concept of using unique strategies to enhance the efficacy of the overall system and thereby provide a basis for future scholarly dialogue and professional applications in the field.

Using these findings as a foundation, the investigation identifies a number of paths for future study that could allow for continued development of this understanding. While the current investigation provides a thorough overview of the main/project variables, there is a large opportunity for continued investigation in a longitudinal manner, whereby examining the -identical correlations over a greater time frame.

A behavioural approach to develop a framework for understanding how certain predictors influence patient outcomes is ideal because it demonstrates a more comprehensive view of patients by recognising that multiple variables impact outcomes either independently or in conjunction with other variables.

Future considerations for development of this framework include making the framework adaptable to changing environmental and operational parameters. Increased understanding of how various independent variables impact decision-making, and behaviours will lead to improved patient outcomes. To better understand these independent variables, the integration of technology will be key in providing real-time feedback mechanisms to quickly assess the impact of previously identified determinants on new determinants as they are identified over time.

REFERENCES

[1] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith, "SoK: Secure Messaging," *2015 IEEE Symposium on Security and Privacy*, pp. 232-249, 2015, .

[2] Z. Wang, Z. Ma, S. Luo, and H. Gao, "Enhanced Instant Message Security and Privacy Protection Scheme for Mobile Social Network Systems," *IEEE Access*, vol. 6, pp. 13706-13715, 2018, ISSN 2169-3536, .

[3] S. Behera, A. Kanth, A. S. Avik, C. V. R. Ashwin, and J. R. Prathuri, "Chat Application Using Homomorphic Encryption," *ITM Web of Conferences*, vol. 50, p. 01011, 2022, .

[4] A. B. Haque, M. A. Bari, S. S. Arman, and F. T. Progga, "A Secure Communication Scheme for Corporate and Defense Community," *arXiv preprint*, 2019, .

[5] P. Shojaei, E. Vlahu-Gjorgievska, and Y. W. Chow, "Security and Privacy of Technologies in Health Information Systems: A Systematic Literature Review," *Computers*, vol. 13, no. 2, p. 41, 2024, ISSN 2073-431X, .

[6] A. Demjaha, J. Spring, I. Becker, S. Parkin, and M. A. Sasse, "Metaphors considered harmful? An exploratory study of the effectiveness of functional metaphors for end-to-end encryption," *Proceedings 2018 Workshop on Usable Security*, 2018, .

[7] H. Abu-Salma, M. A. Sasse, J. Bonneau, and I. Goldberg, "Exploring User Mental Models of End-to-End Encrypted Communication Tools," *USENIX FOCI*, 2018, [Online]. Available:

[8] M. D. P. I. Authors, "Implementation of Secure End-to-End Encrypted Chat Application Using Diffie–Hellman Key Exchange and AES-256 in a Microservice Architecture," *Engineering Proceedings*, vol. 107, no. 1, p. 98, 2025, .