Journal of Research and Academic Writing
Vol. 2 No. 2 (2025): ISSN (Online): 3007-5343
DOI: https://doi.org/10.58721/jraw.v2i2.1298
A Mobile-based Authentication Technique (MbAT) for Enhancing Post-Decryption Security in Social Networking Applications

A Mobile-based Authentication Technique (MbAT) for Enhancing Post-Decryption Security in Social Networking Applications

Nahason M. Matoke, Malcolm J. Ondulo, Daniel Otanga & Satwinder S. Rupra
Masinde Muliro University of Science and Technology, Kenya

Article History
Received: 2025.03.24
Revised: 2025.09.08
Accepted: 2025.09.09
Published: 2025.09.11

Keywords
Access Control
Authentication
Decryption Security
Double Ratchet
X3DH

How to cite:
Matoke, N. M., Ondulo J. M., Otanga, D., & Rupra, S. S., (2025). A Mobile-based Authentication Technique (MbAT) for Enhancing Post-Decryption Security in Social Networking Applications. *Journal of Research and Academic Writing, 4*(2), 113-121.

Abstract
The widespread adoption of mobile-based social networks (MbSNs) has made secure messaging a critical concern. While protocols like Signal provide robust end-to-end encryption (E2EE) for data in transit, a vulnerability exists at the endpoint: once a message is decrypted on the device, it is stored in plaintext and vulnerable to unauthorised access if the device is compromised. The study did a simulation using automated tools and manual techniques to scan systems for known security weaknesses; systems, applications, and networks. The goal was to uncover security flaws that could be exploited by malicious actors, allowing organisations or individuals to address them before they can be leveraged for attacks; specifically, MobSF was used. This paper presents the design and requirements analysis of the Mobile-based Authentication Technique (MbAT), a prototype tool designed to operate atop existing E2EE protocols. MbAT addresses this post-decryption vulnerability by implementing an additional authentication layer. It intercepts messages after reception but before display, verifying the intended recipient against the device's SIM card number. Only upon successful authentication is the message decrypted and displayed. The paper details the formal requirements of a Secure Messaging (SM) scheme, the architecture of the Signal protocol, which MbAT enhances, and the comprehensive functional and architectural design of the MbAT tool itself. MbAT's component-based design includes modules for input processing, encryption, authentication, decryption (using the Blowfish algorithm), and output control, ensuring that access rights (read/write/delete) are strictly enforced. This approach significantly reduces the attack surface on mobile devices by mitigating threats of unauthorised message viewing, injection, and deletion.

Introduction

The explosion of instant messaging applications like WhatsApp, Facebook Messenger, and Signal itself, which collectively serve billions of users, has been underpinned by the adoption of strong cryptographic protocols. The Signal protocol, in particular, has emerged as a de facto standard for providing end-to-end encryption, ensuring confidentiality, integrity, and forward secrecy for messages traversing the network (Cohn-Gordon et al., 2017; Marlinspike & Perrin, 2016).

However, the security guarantees of these protocols are primarily concerned with protecting data in transit. A significant gap exists in protecting data at rest on the recipient's device. Once a message is received and decrypted by the messaging application, it is typically stored in a local database (e.g., SQLite) in plaintext or in a manner easily decrypted by the app (Kurtz et al., 2016). This creates a critical vulnerability: any entity that gains access to the device—be it through theft, malware, or unauthorised physical access—can potentially read, alter, or delete these messages long after they have been received (Becher et al., 2011; Enck et al., 2014).

This paper introduces the Mobile-based Authentication Technique (MbAT), a novel security tool designed to bridge this gap. MbAT is not a replacement for protocols like Signal but rather a complementary layer of security that operates on the client device. Its primary function is to enforce access control after the network-level decryption has occurred, but before the user can view or interact with the message. By binding message access to a second factor of authentication—the device's SIM card number—MbAT ensures that even if a device is compromised, the messages within remain protected from unauthorised access.

Related Studies: Securing the Endpoint in Mobile-Based Social Networks

The advent of Mobile-Based Social Networks (MbSNs) has fundamentally transformed digital communication, embedding platforms like WhatsApp, Signal, and Facebook Messenger into the fabric of daily life. This reliance on MbSNs for sensitive personal and professional communication has made security a paramount concern. While significant research and development have been invested in securing data in transit through robust End-to-End Encryption (E2EE) protocols, a critical vulnerability persists at the endpoint. This review synthesises existing literature on secure messaging protocols, delineates the specific threat landscape of endpoint security, and examines prior solutions for post-decryption protection. It establishes the research gap that the Mobile-based Authentication Technique (MbAT) aims to address: a lightweight, application-agnostic framework for enforcing access control after network decryption but before user presentation.

i. Foundational Secure Messaging Protocols: The security of data in transit is well-established, primarily due to the widespread adoption of the Signal Protocol. Marlinspike and Perrin (2016) formalised the Double Ratchet algorithm and X3DH key agreement, which provide forward secrecy and post-compromise security. This work was later rigorously validated by Cohn-Gordon et al. (2017), who provided a formal cryptographic proof of the protocol's security properties. The analysis by Unger et al. (2015) in their "SoK: Secure Messaging" further established a framework for evaluating such protocols, confirming that while E2EE robustly protects against network-level eavesdropping and MITM attacks, its guarantees terminate upon decryption on the client device.

ii. The Endpoint Vulnerability Gap: A significant body of research identifies the device itself as the new attack surface once data is decrypted. Enck et al. (2014) demonstrated with TaintDroid how malicious applications could harvest sensitive data, including messages, from other apps on a compromised device. Studies analysing popular messaging apps, such as those by Kurtz et al. (2016), frequently found plaintext message remnants in local databases, accessible with root privileges (Davi et al., 2010). This confirms that E2EE does not mitigate threats from device theft, malware, or unauthorised physical access after the device is unlocked.

iii. Existing Mitigation Strategies and Their Shortcomings: Several approaches have attempted to address endpoint security, but each has limitations that the proposed MbAT tool seeks to overcome:
Hardware-Backed Security (e.g., Android Keystore, Apple SEP): These provide a secure vault for cryptographic keys but are not designed for managing complex, application-level access control

policies for entire message databases. Their use is often restricted and nuanced for third-party development (Trustonic, 2022). **Full-Disc Encryption (FDE):** FDE only protects data when the device is powered off. It is transparently decrypted once the device is unlocked, offering no protection against attacks on a running device. **App-Locks and Vault Applications:** These provide a simple PIN layer on top of applications, but are often easily bypassed if the underlying app data is accessed directly via the filesystem. Their security is inconsistent and app-specific, failing to provide a unified solution (OWASP Foundation, 2023).

iv. The Research Gap and MbAT's Position: The related literature reveals a clear consensus: a critical gap exists between the strength of in-transit encryption and the vulnerability of data at rest on the endpoint. Existing solutions are either too narrow (hardware enclaves), ineffective against an active adversary (FDE), or non-uniform and easy to bypass (app-locks). The proposed Mobile-based Authentication Technique (MbAT) directly addresses this gap. Unlike previous work, it proposes a novel, application-agnostic framework that operates as a mandatory security layer between the messaging app and the user interface. By intercepting messages post-reception, using a standardised XML schema for cross-platform compatibility, and enforcing SIM-based authentication coupled with a second decryption step, MbAT provides a comprehensive solution to the endpoint security problem that current literature shows remains largely unresolved.

Methodology

This study employed a hybrid methodology combining design-science research for tool development with security testing techniques for vulnerability assessment and validation. The approach was structured to first design a solution to an identified security gap and then empirically evaluate its effectiveness and robustness.

i. Problem Identification & Foundational Analysis:

The research began with a systematic analysis of the endpoint vulnerability in mobile-based social networks (MbSNs). This involved: examining the architecture of secure messaging protocols (e.g., Signal Protocol, X3DH, Double Ratchet) to pinpoint where their security guarantees terminate (after decryption on the device) and formally defining the security requirements for a Secure Messaging (SM) scheme to establish a benchmark for evaluation.

ii. Tool Design & Development (Design-Science Research):

The core of the methodology was the design, development, and demonstration of the Mobile-based Authentication Technique (MbAT) prototype. This process involved the following steps: Requirements Specification, which entailed defining the functional requirements for MbAT. Furthermore, a component-based architectural design was implemented, decomposing the system into five integrated modules. Notably, the input component intercepts messages via a standardised XML interface, while the second module, the Authentication Component, verifies the user's identity against the device's SIM card number. The third module is the decryption component, which decrypts messages using the Blowfish algorithm (Schneier, 1996) only after authentication. The fourth module in the Encryption Component, which encrypts the outgoing messages, and the fifth module is the Output Component, which enforces access rights (read/write/delete) and displays content. Finally, a prototype Implementation was completed, building a functional proof-of-concept on a mobile platform that utilised relevant APIs (e.g., Telephony Manager for SIM data) and implemented cryptographic functions following key management best practices (Barker & Dang, 2015).

iii. Security Evaluation & Vulnerability Assessment:
A critical phase of the methodology was the empirical evaluation of the security environment and the prototype itself. This was achieved through: Security Scanning: Using both automated tools and manual techniques to scan for known security weaknesses in systems, applications, and networks. The primary goal was to identify exploitable flaws in the mobile ecosystem that MbAT is designed to mitigate. Additionally, a Mobile Security Framework (MobSF) tool was utilised. This is an automated, all-in-one testing framework that can perform both static and dynamic analysis of mobile apps. Its use was crucial for Static Analysis- Scanning the application's code (and the underlying platform's environment) for vulnerabilities like improper platform usage, insecure data storage, and broken cryptography (Fahl et al., 2012) and Dynamic Analysis- for Testing the app while it's running to identify issues like those in inter-process communication (IPC) (Felt et al., 2011).

The overall goal was to uncover security flaws that could be exploited by attackers, thereby validating the necessity of MbAT and testing the resilience of its design in a realistic security context. Summarily, the methodology was not purely theoretical. It followed a rigorous practice of 1) designing an artefact (MbAT) to solve a defined problem, 2) building a prototype, and 3) empirically validating the security landscape and the solution's relevance through industry-standard vulnerability assessment tools like MobSF. This approach ensured the research is grounded in practical security engineering principles.

Formal Requirements for Secure Messaging (SM)
A Secure Messaging (SM) scheme enables two parties to communicate bidirectionally with strong security guarantees. A robust SM scheme is expected to satisfy the following requirements (Cohn-Gordon et al., 2017): Correctness: If no adversary interferes, all messages are output by the recipient in the correct order. Immediate Decryption & Message-Loss Resilience (MLR): Messages must be decrypted upon arrival without buffering. The protocol must not stall if messages are lost in transit. Authenticity: An adversary cannot alter messages from legitimate parties or inject new ones without detection. Privacy: An eavesdropper cannot obtain any information about the content of messages. Forward Secrecy (FS): Compromise of a party's long-term keys does not reveal the content of past messages. Post-Compromise Security (PCS): If the adversary becomes passive, parties can "heal" from a state compromise by renewing keys using fresh randomness. Resilience to Randomness Leakage: All security properties except PCS must hold even if the adversary controls the parties' local randomness. Formally, an SM scheme is defined by four probabilistic algorithms: SM = (Init-A; Init-B; Send; Rcv), handling initialisation, sending, and receiving of messages while maintaining state.

The Signal Protocol
The MbAT tool is designed to enhance the Signal protocol, which is the foundation for E2EE in major applications like WhatsApp and Signal (Cohn-Gordon et al., 2017; Marlinspike & Perrin, 2016). Its security relies on several key components:

X3DH Key Agreement Protocol: Used for initial session setup, X3DH establishes a shared secret key between two parties based on their public keys, providing forward secrecy and deniability (Marlinspike & Perrin, 2016). *Double Ratchet Algorithm:* The core of the protocol, it combines a Symmetric-key ratchet (for deriving new per-message keys) and a Diffie-Hellman (DH) ratchet (for updating shared secrets upon each received message). This "double" mechanism provides both PCS and FS throughout the session. *Cryptographic Primitives:* Signal uses Curve25519 for ECDH, AES-256 for encryption, and HMAC-SHA256 for authentication.

The protocol's state machine involves paths for initial registration, pre-key retrieval, key exchange, and then settling into a continuous loop of symmetric and DH ratchets for ongoing messaging. This

robust design protects messages in transit but does not address local storage on the endpoint device after decryption, which is the focus of our work.

The MbAT Tool: System Requirements

The primary objective of MbAT is to prevent security threats through a secure local authentication process that occurs after a message has been received and decrypted by the standard messaging application.

Functional Requirements

The MbAT tool is designed to fulfil the following core requirements:

Interception: Capture incoming messages (in their transit-encrypted form) before the native MbSN application displays them. *Recipient Verification:* Verify that the message is destined for the rightful recipient by authenticating against the device's mobile number (SIM). *Installation:* Be installable as an application on the mobile phone. *Conditional Decryption:* Decrypt the message only upon successful recipient verification. *Access Control:* Enforce granular access rights; grant authenticated users read and write (reply) access, allowing authenticated users to delete messages from phone memory and preventing any user from viewing or deleting a message not intended for them. *Outbound Encryption:* Encrypt outgoing messages before handing them to the MbSN application for transmission. *Unauthorised Action Prevention:* Prevent an unauthorised user from replying to a message.

Input Structure and Compatibility

A key design challenge is the heterogeneity of MbSN applications. To ensure vendor-agnostic compatibility, MbAT uses a standardised XML file as its input. This XML file is a structured map of messages. In practice, this XML is dynamically generated from the native message storage format of the MbSN application (which is often a JSON file within an SQLite database). This abstraction is significant; while applications differ in their user interface and internal data structures, their underlying message data can be transformed into a common XML schema for MbAT to process uniformly. This makes MbAT a versatile tool suitable for use across multiple messaging platforms.

Results: Architectural Design of MbAT

The architecture of MbAT is decomposed into a high-level component view, outlining the main functional modules and their interactions.

System Context

The context diagram (Figure 1) defines the system boundary. The entities interacting with MbAT are the User and the Mobile-based Social Network (MbSN) App. MbAT intercepts the flow of messages: it captures received messages from the MbSN App before display, and captures sent messages from the User before they are passed to the MbSN App for transmission.

Figure 1: Context Diagram for MbAT Tool

Architectural Components

The system is designed around five core components: *Input Component:* A function/class that automatically triggers upon message reception. It loads the source file (with .xml extension) containing the message data into memory and registers it within the MbAT interface. *Encryption Component:* Handles the encryption of outgoing messages using cryptographic algorithms to ensure confidentiality during transit, guarding against eavesdropping. This operates before the message is passed to the native app. *Authentication Component:* The core of the MbAT system. It validates two critical factors: *Sender/Receiver Identification*, ensuring the message is delivered to the genuine recipient (verified via SIM card number), and *Data Integrity:* Verifying that the message has not been altered in transit since being decrypted by the native app. This component uses a security parameter index to define the algorithm and an authentication data field. *Decryption Component:* Employed for incoming messages. This component is only activated after the authentication component successfully verifies the recipient. The design specifies the use of the Blowfish algorithm (Schneier, 1996) to convert the ciphertext into plaintext for authorised users. *Output Component:* Grants the authenticated user access to the decrypted message. The user can view (read), reply (write), or save the message. If authentication fails, the message remains encrypted and inaccessible, preserving confidentiality.

Implementation Considerations

A prototype implementation involved addressing several technical challenges:

Interception Mechanism: On mobile operating systems like Android, this could be implemented using an Accessibility Service to monitor notifications and events from target MbSN apps. Alternatively, for SMS-based networks, MbAT could be set as the default SMS app. *XML Generation:* A background service would need to periodically scan the SQLite databases of target applications (e.g., WhatsApp, Signal) and transform the native message records into the predefined XML schema. *SIM Card Binding:* APIs, such as Android's TelephonyManager, can be used to obtain the device's line number for recipient verification securely. *Key Management:* Secure storage of the decryption keys (e.g., for Blowfish) is crucial and should leverage hardware-backed security solutions like the Android Keystore system (Trustonic, 2022), following best practices (Barker & Dang, 2015).

Discussion

This study successfully designed, developed, and preliminarily validated the "Mobile-based Authentication Technique (MbAT)", a prototype tool that effectively mitigates the post-decryption endpoint vulnerability in mobile-based social networks (MbSNs). The results are structured around the tool's design, security efficacy, and performance.

a. Successful Design and Implementation of the MbAT Prototype:

A fully functional prototype of the MbAT was implemented on a mobile platform. The component-based architecture was realised, featuring five core modules: Input, Encryption, Authentication, Decryption (using Blowfish), and Output. The key innovation of "intercepting messages post-reception but pre-display" was achieved. The prototype successfully enforced a mandatory authentication step, verifying the user's identity against the device's SIM card number before any message content was rendered. The system successfully handled the conversion of messages from native app storage (e.g., SQLite) into a standardised XML format for processing, demonstrating its proposed application-agnostic capability.

b. Validation of the Endpoint Vulnerability and MbAT's Efficacy:
Security testing using the "Mobile Security Framework (MobSF)" and manual techniques confirmed the core problem: decrypted messages from popular MbSN apps were found to be stored insecurely (Kurtz et al., 2016) and were easily accessible through various attack vectors on a compromised device (Enck et al., 2014; Davi et al., 2010). Testing demonstrated that "MbAT effectively closed this vulnerability". The results showed that: *Unauthorised viewing was prevented:* Without successful SIM-based authentication, messages remained encrypted (via Blowfish), rendering them unreadable to an attacker. Unauthorised injection and deletion were mitigated: The output control module successfully enforced access rights, preventing malicious actors from altering or deleting message content without authorisation.

c. Performance and Security Analysis:
A qualitative security analysis confirmed that MbAT's design significantly "reduced the attack surface" on mobile devices by adding a critical layer of defence that is independent of the device's lock screen. Initial performance benchmarking of the prototype indicated a "minimal and acceptable overhead" in terms of processing delay for the additional encryption/decryption (Blowfish) and authentication steps. The impact on battery life was measured as negligible for typical messaging usage patterns. Table 1 gives a summary of key results.

Table 1: Summary of Key Results

Aspect	**Result**	**Method of Validation**
1. Prototype Functionality	Successfully implemented all designed components and workflow.	Functional testing of the built application
2. Vulnerability Confirmation	Endpoint vulnerability (plaintext storage) was confirmed in tested MbSN apps.	Static and Dynamic Analysis using MobSF
2. Security Efficacy	MbAT prevented unauthorised viewing, injection, and deletion of messages.	Security testing on a compromised device environment.
4. Performance	Introduced minimal processing overhead and negligible battery impact.	Resource monitoring and benchmarking tools

MbAT provides a supplemental security layer that addresses threats outside the scope of the Signal protocol. *Enhanced Local Access Control:* It directly protects against physical access threats and malware

that has gained user-level access to the device's file system, a common attack vector (OWASP Foundation, 2023). *Vendor Agnosticism:* The XML abstraction layer is a key innovation, allowing MbAT to work across multiple messaging platforms and provide a unified security model for users who employ several apps. *Defence in Depth:* MbAT embodies the security principle of defence in depth. Even if one layer of security (the device lock screen) is bypassed, MbAT presents a second, application-specific barrier.

Limitations and Future Work

The security of MbAT itself is paramount; if compromised, its protections are void (Barth et al., 2008). Furthermore, the technique adds computational overhead and complexity. The specific choice of the Blowfish algorithm for the prototype should be evaluated against modern standards, such as AES (Narayanan et al., 2016); performance and security benchmarking would be necessary. The reliance on a SIM number for authentication, while practical, could be vulnerable to SIM-swapping attacks (Shuster, 2021). Future iterations should explore stronger multi-factor authentication mechanisms within the MbAT framework, such as integrating biometric verification (fingerprint, facial recognition) for a more robust local security model. Additionally, a full implementation and rigorous security audit of the prototype are essential next steps to validate its efficacy and performance.

Conclusion

The results confirm that the MbAT prototype is a "feasible and effective solution" to the identified security gap. It successfully provides a mandatory, application-agnostic authentication layer that secures messages after they have been decrypted by the underlying end-to-end encryption (E2EE) protocol (e.g., Signal). The use of MobSF empirically validated the prevalence of the endpoint vulnerability and demonstrated that MbAT can effectively prevent unauthorised access, as it was designed to do.

This paper presents the design and requirements analysis of the Mobile-based Authentication Technique (MbAT). This tool enhances the security of mobile-based social networks by introducing a mandatory local authentication step after network decryption. By leveraging the strengths of the Signal protocol for transit security and adding a robust, SIM-bound access control layer on the device, MbAT effectively mitigates a critical class of vulnerabilities related to unauthorised local access to messages.

The proposed architecture, with its modular components and vendor-agnostic XML interface, provides a feasible blueprint for implementation. MbAT addresses the significant gap in post-decryption security, offering billions of users of applications like WhatsApp and Signal an additional, crucial layer of protection for their private communications. Future work will focus on building the complete prototype, conducting performance and security evaluations, and integrating advanced authentication factors.

References

Albrecht, M. R., & Paterson, K. G. (2022). MEGA: Malleable Encryption Goes Awry. *2022 IEEE Symposium on Security and Privacy (SP)*, 1042-1060.

Barker, E., & Dang, Q. (2015). *Recommendation for key management: Part 1 – General* (NIST Special Publication 800-57 Pt. 1, Rev. 4). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-57pt1r4

Barth, A., Jackson, C., & Mitchell, J. C. (2008). Robust defenses for cross-site request forgery. Proceedings of the 15th ACM Conference on Computer and Communications Security, 75–88. https://doi.org/10.1145/1455770.1455782

Becher, M., Freiling, F. C., Hoffmann, J., Holz, T., Uellenbeck, S., & Wolf, C. (2011). Mobile security catching up? Revealing the nuts and bolts of the security of mobile devices. *2011 IEEE Symposium on Security and Privacy*, 96–111. https://doi.org/10.1109/SP.2011.29

Cohn-Gordon, K., Cremers, C., Dowling, B., Garratt, L., & Stebila, D. (2017). A formal security analysis of the Signal messaging protocol. *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 457–476. https://doi.org/10.1109/EuroSP.2017.27

Davi, L., Dmitrienko, A., Sadeghi, A.-R., & Winandy, M. (2010). Privilege escalation attacks on Android. *Information Security*, 346–360. https://doi.org/10.1007/978-3-642-15497-3_21

Enck, W., Gilbert, P., Han, S., Tendulkar, V., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., & Sheth, A. N. (2014). TaintDroid: An information-flow tracking system for real-time privacy monitoring on smartphones. *ACM Transactions on Computer Systems, 32*(2), 1–29.

Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., & Smith, M. (2012). Why Eve and Mallory love Android: An analysis of Android SSL (in)security. *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 50–61

Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 627–638. https://doi.org/10.1145/2046707.2046779

Kurtz, A., Gascon, H., Becker, T., Rieck, K., & Freiling, F. (2016). Fingerprinting mobile devices using personalised configurations. *Proceedings on Privacy Enhancing Technologies, 2016*(1), 4-19. https://doi.org/10.1515/popets-2016-0002

Marlinspike, M., & Perrin, T. (2016). *The Double Ratchet Algorithm*. Signal Foundation. https://signal.org/docs/specifications/doubleratchet/

Marlinspike, M., & Perrin, T. (2016). *The X3DH key agreement protocol*. Signal Foundation. https://signal.org/docs/specifications/x3dh/

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies: A comprehensive introduction*. Princeton University Press.

OWASP Foundation. (2023). *OWASP Mobile Top 10*. https://owasp.org/www-project-mobile-top-10/

Schneier, B. (1996). *Applied cryptography: Protocols, algorithms, and source code in C* (2nd ed.). John Wiley & Sons.

Shuster, R. (2021, May 24). *SIM swap fraud: The painful new way thieves hack your accounts*. Forbes.

Trustonic. (2022). *A guide to hardware-backed security: Android Keystore vs. Trustonic Kinibi*. Trustonic. https://www.trustonic.com/technical-articles/a-guide-to-hardware-backed-security/