



Offline Peer-to-Peer Messaging System with End-to-End Encryption and On-Device Hate Speech Detection Using Deep Learning

Vinit Shinde¹, Rumit Dongre², Harshal Pawar³, Shivhar Gaikwad⁴, *Vaishali Chaudhari⁵*,
Sushama Telrandhe⁶

^{1,2,3,4}UG Student Department of Computer Science and Engineering,
Guru Nanak Institute of Engineering and Technology Nagpur, India

⁵Assistant Professor, Department of Computer Science and Engineering
Guru Nanak Institute of Engineering and Technology Nagpur, India

⁶Associate Professor, Department Computer Science and Engineering
Guru Nanak Institute of Engineering and Technology Nagpur, India

How to Cite this Article:

Shinde, V., Dongre, R., Pawar, H., Gaikwad, S. & Chaudhari, V. (2026). Offline Peer-to-Peer Messaging System with End-to-End Encryption and On-Device Hate Speech Detection Using Deep Learning. International Journal of Creative and Open Research in Engineering and Management, <i>02</i>(10), 1-9.
<https://doi.org/10.55041/ijcope.v2i10.014>

License:

This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.



<https://doi.org/10.55041/ijcope.v2i10.014>

ABSTRACT

Modern messaging applications provide convenient communication but commonly depend on centralized Internet infrastructure and cloud services. This paper presents an offline peer-to-peer (P2P) messaging system designed for direct communication between nearby mobile devices without requiring a conventional messaging server or continuous Internet access. The proposed system combines WiFi/Bluetooth-based local communication, end-to-end encryption, private one-to-one and group messaging, message history, and local file sharing with an on-device natural language processing (NLP) module for hate-speech detection. The communication layer is designed to discover nearby peers, establish authenticated sessions, exchange encrypted messages, and maintain local conversation history. The machine-learning component processes message text locally so that moderation decisions can be made without transmitting message content to a remote inference service. A deep-learning pipeline based on text preprocessing, tokenization, sequence representation, and a recurrent neural network (RNN)-style classifier is considered for the prototype, with PyTorch used for model development and inference integration. The combined architecture addresses two related requirements: communication resilience when Internet infrastructure is unavailable

and privacy-preserving content analysis at the device. The paper describes the system architecture, literature background, methodology, security model, implementation workflow, evaluation criteria, and limitations. Because project-specific benchmark measurements were not supplied for this draft, no fabricated accuracy, latency, or security performance values are reported; instead, the evaluation framework identifies the measurements required during experimental validation.

1. INTRODUCTION

Instant messaging has become a primary mode of personal and group communication. Conventional messaging systems generally use Internet connectivity and centralized or cloud-assisted infrastructure to relay messages, store account information, synchronize devices, and provide additional services. This architecture is convenient, but it creates dependencies on network availability, intermediary infrastructure, and remote processing. In situations such as temporary network outages, infrastructure disruption, remote environments, or local communication scenarios, users may need a direct communication mechanism that does not depend on a conventional Internet connection.

Peer-to-peer communication provides an alternative architecture in which participating devices communicate directly or through nearby peers. Offline mobile communication research has explored decentralized communication using technologies such as Bluetooth Low Energy and other short-range wireless mechanisms. Such approaches demonstrate the feasibility of secure communication without external networking infrastructure,



although practical issues such as discovery, device compatibility, message routing, packet constraints, background execution, and battery consumption remain important engineering considerations.

Security is a second major requirement. End-to-end encryption (E2EE) aims to protect message content so that only intended endpoints can decrypt it. The Signal protocol is a prominent example of a secure messaging design, and formal analysis has examined its X3DH key agreement and Double Ratchet mechanisms. The proposed project takes the general principle of endpoint confidentiality as a design requirement while adapting the overall architecture to an offline, device-to-device setting rather than claiming to reproduce the full Signal protocol.

A third requirement concerns harmful textual content. Hate speech detection has become an established NLP research problem, with studies examining machine-learning and deep-learning methods for automatically classifying harmful or hateful language. Recent literature includes recurrent and hybrid deep-learning architectures, while systematic reviews identify continuing challenges involving dataset quality, language variation, context, ambiguity, and generalization. The proposed system therefore places the detection model on the device, allowing message text to be analyzed locally rather than uploaded to a remote moderation API.

The main contribution of this project is the integration of these components into a single prototype: offline P2P messaging, encrypted communication, local storage, file exchange, and on-device hate-speech detection. The goal is not to replace large-scale Internet messaging platforms, but to investigate a privacy-oriented communication model that remains useful in local or infrastructure-limited environments.

2. LITERATURE REVIEW

Research on secure messaging, decentralized communication, and automated hate-speech detection has developed along partially separate paths. Secure-messaging research has focused strongly on cryptographic protocols and protection against compromised intermediaries, while offline communication research has focused on networking mechanisms that can operate without conventional infrastructure. NLP research has separately investigated automated classification of offensive and hateful language. The proposed project connects these areas at the application level.

2.1 Secure Messaging and End-to-End Encryption.

Cohn-Gordon et al. formally analyzed the Signal messaging protocol and its key agreement and Double Ratchet components, showing how modern secure messaging can provide properties including forward and post-compromise security under a formal model. Comparative work on secure mobile messaging has also examined how different applications implement end-to-end encrypted communication and related privacy mechanisms. These studies establish E2EE as a central architectural concept for protecting message content, but they generally assume ordinary network connectivity and do not by themselves solve offline peer discovery and transport.

2.2 Offline and Decentralized Communication.

Offline communication research has investigated direct mobile-to-mobile architectures using Bluetooth Low Energy and related decentralized networking mechanisms. One research direction explicitly targets secure peer-to-peer communication without external networking infrastructure, demonstrating the feasibility of local device-to-device exchange. Recent work on BLE mesh chat applications also explores multi-hop message forwarding and infrastructure-less communication, while identifying practical constraints such as advertisement packet size, Android background restrictions, heterogeneous devices, mobility, interference, and battery optimization.

2.3 Hate Speech Detection Using NLP.

Automatic hate-speech detection commonly involves text normalization, tokenization or subword representation, feature extraction, and supervised classification. A systematic review of NLP-based hate-speech detection identifies deep learning as an important research direction while emphasizing that performance remains dependent on dataset characteristics, language, annotation quality, and context. Recent research has also explored hybrid recurrent architectures and explainable-AI methods for improving classification and interpretability.

2.4 Research Gap.

The literature indicates that secure messaging, offline networking, and hate-speech detection are individually well studied, but an integrated mobile prototype combining direct offline communication, encrypted message transport,



local file exchange, and on-device hate-speech analysis is a distinct engineering combination. The project therefore focuses on integration, privacy, and local inference rather than proposing a new cryptographic primitive or claiming a universally superior classification algorithm. **Table I: Summary of Relevant Research Directions**

Area	Common Approach	Observed Limitation / Gap
Secure messaging	E2EE, authenticated key exchange, ratcheting	Usually assumes network infrastructure; metadata and usability remain concerns.
Offline communication	Bluetooth/BLE, Wi-Fi Direct, mesh/P2P	Discovery, routing, background execution, range, and battery constraints.
Hate-speech detection	ML/NLP, CNN/RNN/transformer and hybrid models	Dataset bias, language variation, context and false positives.
Integrated system	Communication + security + local NLP	Requires coordinated design across networking, cryptography, storage and ML.

3. PROBLEM STATEMENT AND OBJECTIVES

The problem addressed by this project is the design of a mobile messaging system that can support nearby communication when conventional Internet-based messaging is unavailable, while preserving message confidentiality and providing local detection of potentially hateful text. The system must coordinate wireless peer discovery, secure session establishment, message exchange, storage, file transfer, and local NLP inference without making the message content dependent on a remote moderation server.

Objectives:

- (1) develop an offline P2P messaging prototype; (2) support private and group communication; (3) implement encrypted message exchange; (4) support local message history and file sharing; (5) integrate an on-device hate-speech classifier;
- (6) evaluate communication reliability, security properties, and classifier performance using reproducible measurements.

4. PROPOSED SYSTEM

The proposed system is organized into five major layers: the user interface layer, application/service layer, peer communication layer, security layer, and on-device intelligence layer. The Flutter application provides the user-facing chat interface. The communication service manages peer discovery and connection establishment through supported local wireless technologies. The security layer protects messages before transmission and handles key material and session state. The NLP layer receives message text locally, performs preprocessing and classification, and returns a moderation label or confidence value to the application.

4.1 User Interface Layer.

The interface provides nearby-peer discovery, contact selection, one-to-one chat, group chat, message history, file selection, transfer status, and hate-speech warning indicators. The interface should clearly distinguish between a message being transmitted, delivered locally, stored, or flagged by the local classifier.

4.2 Peer Communication Layer.

Nearby devices advertise or discover their availability through a supported local transport. Depending on the platform and implementation, Wi-Fi-based communication can provide higher-throughput transfers, while Bluetooth/BLE can support discovery and lower-bandwidth local exchange. The architecture should treat the transport as replaceable so that application-level message formats remain independent of the wireless mechanism.

4.3 Security Layer.

A secure session requires authenticated peers and protected message payloads. The project should use established cryptographic primitives and well-reviewed libraries rather than implementing encryption algorithms from scratch. The design target is that plaintext message content is available only at the sender and intended recipient devices. The exact protocol and library used in the final implementation must be documented in the experimental version of the project.



4.4 Local Storage.

Conversation records, peer identifiers, timestamps, delivery state, and file metadata can be maintained locally. Sensitive stored content should be protected using platform-appropriate secure storage and encrypted databases or encrypted fields where required. The storage design should also define how deleted messages and temporary decrypted files are handled.

4.5 On-Device Hate-Speech Detection.

Message text is passed directly from the local chat layer to the NLP model. A preprocessing stage normalizes text and converts it into a representation accepted by the trained classifier. The model produces a classification result, such as non-hateful or potentially hateful, which is returned to the UI. Because the inference is local, the system does not need to send message text to a cloud moderation endpoint for this feature. **Figure 1: High-Level System Architecture**

Flutter UI Chat • Groups • Files • Alerts
Application Services Peer Manager • Message Manager • File Manager • History
Security Layer Identity • Session Keys • Encryption • Authentication
Offline Transport Wi-Fi / Wi-Fi Direct / Bluetooth / BLE
On-Device AI NLP Preprocessing → RNN-based Classifier → Local Result

5. METHODOLOGY

The development methodology is divided into dataset preparation, model development, application development, security integration, transport integration, and system evaluation. The project follows an incremental approach so that messaging can first be validated locally before the encryption, file-transfer, and machine-learning components are integrated.

5.1 Dataset and NLP Pipeline.

A labeled hate-speech dataset appropriate to the language and intended use case must be selected and documented. The dataset is divided into training, validation, and test partitions without leakage between partitions. Preprocessing may include normalization, tokenization, removal or preservation of selected punctuation, handling of URLs and mentions, and conversion into numerical sequences. The final preprocessing rules must match those used during inference.

5.2 Deep Learning Model

The proposed prototype uses an RNN-oriented architecture because recurrent networks are designed to process sequential text representations. A practical pipeline can contain an embedding layer followed by an RNN, LSTM, or GRU component and a classification layer. The precise architecture should be selected after comparing validation results and deployment constraints. The model is trained in PyTorch and exported or packaged in a form suitable for mobile inference. For an on-device system, model size, inference time, memory consumption, and compatibility with the target mobile runtime are important alongside classification quality.

The output should not be interpreted as an absolute determination that a person is hateful. It is a machine-learning classification of the message according to the labels and annotation rules of the training dataset. The application can therefore present a warning, request confirmation, or apply a project-defined moderation action rather than automatically imposing an irreversible penalty.

5.3 Peer Discovery and Connection Workflow

The communication workflow begins with peer discovery. A device enters discoverable mode and identifies nearby compatible devices. After a peer is selected, the application establishes a transport connection and performs an



application-level handshake. The handshake exchanges the information required to authenticate the peer and establish or retrieve the session-security state. Once the secure channel is established, application messages are serialized, encrypted, transmitted, authenticated, and reconstructed at the receiver.

For file transfer, the system should separate control messages from file payloads. Metadata such as filename, MIME type, size, transfer identifier, and integrity information can be sent first, followed by chunked encrypted data. Chunking allows transfers to be resumed or rejected cleanly when a connection is interrupted. Temporary files should not remain in plaintext storage after a failed or completed transfer unless explicitly required by the application.

5.4 Message Processing Workflow

Step	Operation	Expected Output
1	Discover nearby devices	Available peer list
2	Select peer / group	Communication target
3	Authenticate and establish secure session	Session state / keys
4	Enter message	Plaintext local message
5	Run local NLP inference	Classification result
6	Encrypt and transmit	Protected message payload
7	Decrypt and verify at receiver	Authenticated plaintext
8	Store local history	Conversation record

Table II: Proposed Message Processing Workflow

6. SECURITY AND PRIVACY CONSIDERATIONS

Security is treated as a system property rather than only an encryption function. The prototype should consider confidentiality, integrity, peer authentication, replay resistance, key protection, local storage protection, and secure deletion. The transport itself should not be considered trusted merely because it is local or wireless. Message payloads should remain cryptographically protected across the transport.

The project should avoid presenting its implementation as equivalent to a formally verified production protocol unless the exact protocol and implementation have been independently analyzed. Where an established protocol or library is used, the paper should identify it precisely and preserve its documented security assumptions. Local hate-speech inference also improves privacy by reducing the need to transmit message content to a third-party service, but model execution does not automatically protect all metadata, logs, notifications, screenshots, backups, or operating-system-level information.

7. RESULTS AND DISCUSSION

The results section for this project must distinguish between implemented functionality and experimentally measured performance. The current paper draft is based on the project specification and literature background; no measured dataset accuracy, end-to-end latency, throughput, battery consumption, or penetration-test results were supplied. Accordingly, the following discussion defines the expected evaluation structure rather than inventing numerical results.

7.1 Functional Evaluation

The prototype should be tested for peer discovery, connection establishment, one-to-one messaging, group messaging, message persistence, file transfer, reconnection after interruption, encryption/decryption, and local hate-speech inference. Each feature should be tested under both normal and failure conditions. For example, a file transfer



should be tested when the connection is interrupted and restored, while a message should be tested for correct rejection if authentication or integrity verification fails.

7.2 Communication Evaluation.

Useful measurements include peer-discovery time, connection-establishment time, message delivery latency, successful-delivery rate, file-transfer throughput, maximum practical communication range, and behavior when devices move out of range. These measurements should be collected across repeated trials and reported with the test environment, device models, operating-system versions, wireless technology, and environmental conditions.

7.3 Machine-Learning Evaluation.

The hate-speech classifier should be evaluated on a held-out test set using accuracy, precision, recall, F1-score, and a confusion matrix. Because false positives can incorrectly flag benign communication and false negatives can allow harmful content to pass undetected, precision and recall should be reported separately rather than relying only on accuracy. If the dataset contains multiple classes, per-class metrics and macro/weighted averages should be reported.

Table III: Recommended Evaluation Metrics

Component	Metric	Purpose
P2P communication	Discovery time, connection time	Measures local-network responsiveness.
Messaging	Delivery success rate, latency	Measures reliability of message exchange.
File sharing	Throughput, transfer success rate	Measures practical file-transfer performance.
Security	Authentication failures, tamper detection	Tests protection against invalid or modified messages.
NLP model	Precision, recall, F1, accuracy	Measures classification quality.
On-device inference	Inference time, model size, memory	Measures suitability for mobile deployment.

7.4 Discussion

The integrated design provides a clear privacy and resilience objective. Local peer communication can reduce dependence on remote messaging infrastructure, while end-to-end encryption is intended to protect message contents over the wireless channel. On-device hate-speech detection further reduces the need to transfer message text to an external inference service. These benefits are balanced by practical constraints: nearby communication has limited range, peer availability is dynamic, wireless links may be unstable, and mobile operating systems can restrict background networking.

The machine-learning component introduces a separate set of limitations. Hate speech is highly context-dependent, and the meaning of a phrase can change according to language, dialect, sarcasm, quotation, target group, and conversational context. A model trained on one dataset should therefore not be assumed to generalize to every messaging environment. Dataset documentation, class balance, annotation policy, language coverage, and test-set construction must be reported with the final experimental results.

From an engineering perspective, the strongest value of the prototype is the combination of functions into one local architecture. Rather than treating messaging, encryption, and NLP moderation as separate demonstrations, the system provides a single workflow in which a message can be classified locally, encrypted, transmitted directly, decrypted by the intended peer, and stored locally.



8. LIMITATIONS

The current prototype concept has several limitations. First, offline communication is inherently constrained by device proximity and the capabilities of the chosen wireless technologies. A user cannot communicate with a distant peer unless an appropriate multi-hop or relay architecture is implemented. Second, platform restrictions may affect Bluetooth permissions, Wi-Fi APIs, background execution, and device-to-device connectivity. These issues can make behavior differ across Android versions and hardware.

Third, end-to-end encryption does not automatically conceal all metadata. Peer identifiers, timing, device information, traffic patterns, local notifications, or operating-system backups may expose information outside the encrypted payload. The final implementation should document exactly what information is protected and what remains visible to the operating system or wireless environment.

Fourth, the hate-speech model is limited by its training data and classification definition. A model cannot reliably determine social intent in every context, and a warning system can produce both false positives and false negatives. The application should therefore avoid representing model output as a definitive judgment about a person. Thresholds, confidence handling, and user-facing actions should be documented.

9. FUTURE SCOPE

Future development can extend the system in several directions. Multi-hop peer routing could allow messages to travel through intermediate devices while retaining message confidentiality. Store-and-forward mechanisms could support temporary disconnections. More efficient wireless discovery could reduce battery usage, and adaptive transport selection could choose Bluetooth, Wi-Fi Direct, or another supported mechanism based on payload size and device capability.

The machine-learning component could be expanded to multilingual and code-mixed text, including Indian-language messaging scenarios. Model compression, quantization, pruning, and optimized mobile inference could reduce memory and latency. Explainable-AI techniques could also be investigated so that the application can provide more transparent reasons for a classification without exposing sensitive message content.

From a security perspective, future work could include a formal threat model, independent security review, stronger peer identity verification, secure key backup strategies, and systematic testing against replay, impersonation, message tampering, and local storage attacks. A larger controlled evaluation involving multiple devices and real-world wireless conditions would provide stronger evidence for scalability and reliability.

10. CONCLUSION

This paper presented an integrated offline peer-to-peer messaging architecture combining direct local communication, end-to-end encrypted messaging, group communication, message history, file sharing, and on-device hate-speech detection. The design addresses two practical requirements that are often treated separately: communication resilience when conventional Internet infrastructure is unavailable and privacy-preserving analysis of potentially harmful message content.

The literature supports the use of established secure-messaging principles, decentralized local communication mechanisms, and deeplearning-based NLP for the individual components. The proposed project brings these components together in a mobile prototype while keeping the machine-learning inference local. The final contribution should be evaluated experimentally using reproducible communication, security, and classification metrics. Since project-specific measurements were not supplied for this draft, the paper deliberately avoids unsupported numerical claims. Future work can extend the prototype toward multi-hop communication, multilingual detection, model optimization, formal security analysis, and broader real-device evaluation.



REFERENCES

- [1] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, doi:10.1109/EuroSP.2017.27.
- [2] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol," *Journal of Cryptology*, vol. 33, pp. 1914–1983, 2020, doi:10.1007/s00145-020-09360-1.
- [3] C. Johansen, "The Snowden Phone: A Comparative Survey of Secure Instant Messaging Mobile Applications," *Security and Communication Networks*, vol. 2021, Article 9965573, 2021, doi:10.1155/2021/9965573.
- [4] "An offline mobile access control system based on self-sovereign identity standards," *Computer Networks*, 2022, doi:10.1016/j.comnet.2022.109434.
- [5] "Decentralized Bluetooth Low Energy (BLE) Mesh Chat Application for Infrastructure-Less Communication," 2025/2026 researchpreprint, doi:10.13140/RG.2.2.21183.16806.
- [6] "A systematic review of hate speech automatic detection using natural language processing," *Neurocomputing*, vol. 546, Article 126232, 2023, doi:10.1016/j.neucom.2023.126232.
- [7] "Towards safer online communities: Deep learning and explainable AI for hate speech detection and classification," *Computers and Electrical Engineering*, vol. 116, Article 109153, 2024, doi:10.1016/j.compeleceng.2024.109153.
- [8] R. Prabhu and V. Seethalakshmi, "A comprehensive framework for multi-modal hate speech detection in social media using deeplearning," *Scientific Reports*, vol. 15, Article 13020, 2025, doi:10.1038/s41598-025-94069-Z.
- [9] K. P. Impana, K. B. Vikhyath, M. Pavana, and A. N. Hemalatha, "A Novel Deep Learning Approach for Hate Speech Detection on SocialPlatforms," in *Data Science and Exploration in Artificial Intelligence (CODE-AI 2025)*, Springer, 2026.