Master project documentation

By Olivia Sabb

❓ Unmet needs

- Many existing messaging apps do not offer a sufficient level of security and privacy for users who handle sensitive information.
- Free messaging apps often come with security risks and data mining threats.
- Some users may desire a messaging platform that is independent of major tech companies and ad networks.
- Existing solutions may lack specific features, such as fully decentralized architecture or minimal metadata collection.
- This app aims to provide a higher level of control over data and communication privacy, filling the gap for users who require a more secure and trustworthy messaging experience.

🎯 Objectives

- Implement end-to-end encryption using AES and RSA by [*Insert Date, e.g., October 31, 2025*].
- Develop a user authentication system with hashed and salted passwords using bcrypt/argon2 by [*Insert Date, e.g., September 15, 2025*].
- Create a basic inbox/outbox system with secure message storage by [*Insert Date, e.g., October 7, 2025*].
- Achieve a message delivery latency of less than 0.5 seconds.
- Successfully complete a basic security audit to identify potential vulnerabilities by [*Insert Date, e.g., November 1, 2025*].

👥 User personas

- Security-Conscious Journalist (Sarah):
    - Demographics: 35 years old, female, tech-savvy, based in a major city.
    - Technical Skills: Proficient with various communication tools, understands encryption concepts.
    - Motivations: Needs to communicate securely with sources to protect their identities and sensitive information.
    - Pain Points: Worried about government surveillance and data breaches, needs a reliable and private communication channel.
- Privacy-Focused Small Business Owner (David):

    - Demographics: 45 years old, male, moderate technical skills, owns a consulting firm.
    - Technical Skills: Familiar with basic security practices, uses various business applications.
    - Motivations: Wants to protect confidential business information and client data from competitors and cyber threats.
    - Pain Points: Concerned about data leaks and unauthorized access, needs a secure way to communicate with employees and clients.

🦾 Jobs we want to cover

secure, encrypted message.

- Receive and decrypt a secure message.
- Securely store messages in an inbox/outbox system.
- Authenticate users with hashed and salted passwords.
- Manage contacts within the application.

📜Some history

- Inspired by the increasing need for secure communication in a world of growing digital surveillance and data breaches.
- The initial goal was to create a simple, lightweight messaging platform with end-to-end encryption.
- The project has evolved to include user authentication and secure message storage to provide a more complete secure messaging solution.

👀 Constraints

- **Technical Constraints:** Limited experience with cryptography libraries, potential challenges in implementing secure key exchange.
- **Time Constraints:** Balancing project development with other commitments, setting realistic deadlines for feature implementation.
- **Security Constraints:** Ensuring the application is resistant to various attacks, staying up-to-date with the latest security best practices.
- **Budget Constraints:** As a personal project, resources are limited, requiring cost-effective solutions and open-source libraries.

🗒️ Releases

- **Version 0.1 ([*Insert Date, e.g., August 15, 2025*]):**
    - Implemented basic user authentication with bcrypt password hashing.
    - Set up SQLite database for user and message storage.
- **Version 0.2 ([*Insert Date, e.g., September 1, 2025*]):**
    - Implemented AES encryption for message content.

    - Created basic inbox/outbox functionality.
- **Version 0.3 ([*Insert Date, e.g., September 20, 2025*]):**
    - Implemented RSA encryption for key exchange.
    - Added Flask web interface for sending and receiving messages.

👣Next steps

- Implement Argon2 password hashing as an alternative to bcrypt.
- Conduct thorough testing of encryption and authentication mechanisms.
- Develop a CLI version of the messaging platform.
- Explore integration with other security tools and services.
- Consider adding features like disappearing messages and screenshot detection

📈 Impact

- Provide a secure and private communication platform for individuals and small businesses.
- Enhance user awareness of security and privacy best practices.
- Demonstrate practical knowledge of cryptography, authentication, and secure data storage.
- Improve communication and information flow for users who require secure messaging.