A secure client-server mobile chat application implementing elliptic curve integrated encryption system (ECIES) and other security features.

Daniel Furnivall

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

1st April 2022

Abstract

abstract goes here

Acknowledgements

acknowledgements go here

Contents

Chapter 1: Introduction

In an increasingly digital world, we are constantly producing data (and, of course, corresponding metadata). It's estimated that humanity created somewhere in the order of 2.5 exabytes of digital data per day in 2018[1] and the growth of digital data creation by individuals is constantly accelerating.

As storage costs decrease over time[14], economic and political incentives have begun to develop for nation state actors and major organisations to develop profiling systems using large-scale data collection and mining (the much vaunted "Big Data"). These systems are already being used for targeted advertising[4], market segmentation[9], criminal investigations[15] and sentencing[10]. In the political sphere, these models have been used for increasingly effective traditional campaigning, [5] as well as (alleged) psychological manipulation[2] and disinformation[12] campaigns.

The question of whether the average individual is able to enjoy the fruits of such data collection is less clear. Do these large actors have the best interests of the subjects of their data in mind? If not, methods for obfuscating or hiding sensitive data become valuable considerations.

In a pre-digital world, an individual could avoid eavesdropping or data collection by malicious entities by simply speaking in a hushed voice, shredding documents, moving quietly or looking over his or her shoulder. In the current landscape, it is much more difficult to avoid surveillance without eschewing technology entirely. Our mobile phones are constantly communicating our triangulated locations, and even if our web traffic is encrypted by default, browser or ISP metadata still contains useful profiling information.

Secure messaging applications aim to allow individuals to communicate with each other individuals around the globe while avoiding the potential for eavesdropping. In theory, this means the user can enjoy the benefits of a globally connected world while preserving their privacy. However, in practice there are of course many implementation difficulties to be considered.

1.1 Why are secure chat applications needed?

Secure messaging applications provide a means of communication between individuals or groups across a network of some kind. This can take the form of text, audio or video messaging, document or file sharing.

To begin to understand the users of secure chat applications, two important questions need to be answered:

1. Who desires secure messaging?
2. Who are they aiming to protect their data from?

There are many categories of potential users of such applications, and from wildly different settings. These can be mundane and innocuous, such as the organisation of a surprise party for a friend or family member.

However, another group of potential users are those who seek to hide criminal behaviour from law enforcement organisations. A high-profile example of this would be the EncroChat network of encrypted phones, predominantly used by organised crime, which was unveiled after a Europe-wide infiltration and investigation of the network by law enforcement groups (leading to several thousand arrests) [11].

Another group who may wish to evade police or law enforcement are political dissidents. In what has become known as the ”Million Dollar Dissident” [6] case, a dissident in the UAE, Ahmed Mansoor, was targeted by the NSO group (an Israeli cybersecurity firm which produces spyware for government use) and subsequently had his passport confiscated as well as being beaten, having his car stolen and finally imprisoned by the UAE authorities within a week of posting anti-government posts online [8]. There are many parallels here with other groups who benefit from secure messaging - whistleblowers sharing information with journalists, and police informants who need to share data secretly with law enforcement.

In reality, there are plenty of reasons for *everyone* to use secure messaging such as protecting data in case of device theft, avoiding embarrassment, limiting exposure to blackmail. Continuing the comparison to real-world communication - when speaking out loud, people don’t tend to shout all the time, and tend to consciously limit and monitor who is listening to a conversation.

It should be clear that although there are many potential categories of secure messaging users, there are also many potential adversaries for these kinds of platforms, which means the development of such services is a complex undertaking. There are major tradeoffs which need to be made between usability and data security - for example, how can we preserve security of messages while also storing them on a mobile device?

1.2 Objectives

The primary development objective of this project was to develop an Android mobile application (and corresponding server) that used complex security features including End-to-end encryption (E2EE), pseudonymous identity and self-destructing messages.

During the development journey of this application, the intrinsic motivation was to come to a greater understanding of the complexities, assumptions and tradeoffs involved in creating these kinds of platforms.

Chapter 2: Requirements and Analysis

¡Discussion should go here about Unger et al's paper and the three key challenges (Trust establishment, conversation security and transport privacy)[13]

2.1 Existing applications in this field

2.1.1 Telegram

2.1.2 Whatsapp

2.1.3 Signal

2.2 Issues

2.2.1 Closed source

2.2.2 Tradeoffs between security and usability features

2.2.3 Nation state control

2.3 Features

2.3.1 Self-destructing messages

2.3.2 End-to-end encryption

DRAFT - potentially worth mentioning possible attacks on crypto chats and metadata e.g. message length attacks [3]

Chapter 3: Design and Implementation

3.1 Development tooling

There are two primary components to the overall system - one or more Android mobile client applications which communicate with a central, deployed server.

3.1.1 Client

The final client application was written entirely in Kotlin (v1.6.10) using the Android Studio IDE.

Key Dependencies

- Socket.io - v2.0.0 - the developers of socket.io provide a native Java implementation. Due to Java's seamless interoperability with Kotlin, this did not cause any implementation problems with the version implemented.
- BouncyCastle - v1.67 - BouncyCastle is the cryptography API that performs a lot of the cryptographic operations within the application, including generation of elliptic curve keypairs. There is a built-in version of BouncyCastle which exists within the Android SDK, but this does not support ECC, which meant that the client application needed to replace the inbuilt library with a more updated version.
- Room ORM - v2.4.0 - Room provides an object-relational mapping (ORM) over the SQLite database built into the Android SDK. When developing the application, the main data model to consider on the client side was that of an individual chat message, including metadata (e.g. recipient) and content. Working with Room allowed for a more streamlined development experience and reduction of boilerplate code, as it meant working directly with Kotlin (Java) entities instead of writing complex queries for inserting messages to our viewmodel. SQL queries were still used to capture relevant data to display to the user in chat windows.

3.1.2 Server

The server application was written in Python v3.9.1 in the PyCharm IDE and utilised several libraries which are highlighted below.

Key Dependencies

- python-socketio - v5.5.0 - a Python server implementation of socket.io, funded by the original socket.io developers.
- aiohttp - v3.8.1 - an asynchronous http server which (importantly) supports websockets. The socket.io process attaches itself to the aiohttp server which allows information to be transmitted and received from clients.

Deployment

To allow users to message other users, it was necessary to deploy the server application on a static IP. To do this, a 1GB/1CPU Azure Virtual Machine instance was used, running Ubuntu 20.04 in the West Europe region.

3.1.3 Final report

This dissertation was written entirely in LaTeX using Neovim v0.5.1 with the VimTex plugin.

3.1.4 Version Control

Git was the version control solution used throughout the project. Discomfort with the idea of storing the written report with a private company (Overleaf) while writing meant that it was worth taking the step of compiling the document locally. This had the fortunate benefit of meaning the entire project could be stored in a single git repository (as well as an opportunity to learn a lot of new things).

The upstream repository host of choice for this project was GitHub, and the entirety of the project can be found at the following link:

https://github.com/furnivall/SEng_Final_Project

3.2 Encryption Implementation

[7]Here is a placeholder for a piece of text about ECIES.

3.2.1 Asymmetric component

3.2.2 Symmetric component

3.2.3 Difficulties

3.2.4 Storage of encryption keypair

3.3 Design Patterns

3.3.1 Dependency injection

3.3.2 Command pattern

3.3.3 Singleton pattern

3.3.4 Observer pattern

3.3.5 Adapter pattern

3.3.6 Data Access Object pattern

3.4 Persistence and storage considerations

3.4.1 Storage of self-destruct duration

3.4.2 Existing user persistence and reconnection flow

Chapter 4: Evaluation & Testing

Chapter 5: Conclusion

Appendix A: First appendix

A.1 Section of first appendix

Appendix B: Second appendix

Bibliography

[1] R. Baeza-Yates and U. M. Fayyad. The attention economy and the impact of artificial intelligence. In *Perspectives on Digital Humanism*, pages 123–134. Springer, Cham, 2022.

[2] H. Berghel. Malice domestic: the cambridge analytica dystopia. *Computer*, 51(5):84–89, 2018.

[3] J. P. Degabriele. Hiding the lengths of encrypted messages via gaussian padding. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1549–1565, 2021.

[4] A. Farahat and M. C. Bailey. How effective is targeted advertising? In *Proceedings of the 21st international conference on World Wide Web*, pages 111–120, 2012.

[5] D. Kreiss and S. C. McGregor. The "arbiters of what our voters see": facebook and google's struggle with policy, process, and enforcement around political advertising. *Political Communication*, 36(4):499–522, 2019.

[6] B. Marczak, J. Scott-Railton, S. McKune, B. Abdul Razzak, and R. Deibert. HIDE AND SEEK: Tracking NSO Group's Pegasus Spyware to operations in 45 countries. Technical report, 2018.

[7] V. G. Martınez, L. H. Encinas, et al. A comparison of the standardized versions of ecies. In *2010 Sixth International Conference on Information Assurance and Security*, pages 1–4. IEEE, 2010.

[8] M. Mazzetti, A. Goldman, R. Bergman, and N. Perlroth. A new age of warfare: how internet mercenaries do battle for authoritarian governments. *The New York Times*, 21:2019, 2019.

[9] K. Pantelis and L. Aija. Understanding the value of (big) data. In *2013 IEEE International Conference on Big Data*, pages 38–42. IEEE, 2013.

[10] R. Simmons. Big data and procedural justice: legitimizing algorithms in the criminal justice system. *Ohio St. J. Crim. L.*, 15:573, 2017.

[11] P. Sommer. Evidence from hacking: a few tiresome problems. *Forensic Science International: Digital Investigation*, 40:301333, 2022.

[12] C. Stöcker. How facebook and google accidentally created a perfect ecosystem for targeted disinformation. In *Multidisciplinary International Symposium on Disinformation in Open Online Media*, pages 129–149. Springer, 2019.

[13] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith. Sok: secure messaging. In *2015 IEEE Symposium on Security and Privacy*, pages 232–249. IEEE, 2015.

[14] C. Walter. Kryder's law. *Scientific American*, 293(2):32–33, 2005.

[15] S. Zawoad and R. Hasan. Digital forensics in the age of big data: challenges, approaches, and opportunities. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pages 1320–1325. IEEE, 2015.