RESEARCH ARTICLE OPEN ACCESS

A Modular Instant Messaging System

Mohamad Raad, Zouhair Bazzal, Majd Ghareeb, Hanan Farhat, Semar Bahmad
Department of Computer and Communications Engineering
Lebanese International University
Beirut, Lebanon

***ABSTRACT*:**
Instant Messaging (IM) Android applications are a trend nowadays. These applications are categorized according to their features: usability, flexibility, privacy and security. However, IM applications tend to be inflexible in terms of functionality offered. The "Dble-U" system was developed as a solution to this inflexibility, with a focus on privacy as an example use case. "Dble-U" is a configurable modular system consisting of an Android chatting application, a privacy controller application, along with a tracking server application. This modularity allows for groups of individuals to build custom features into their IM application.
Keywords: Instant messaging, modular, configurable.

I. INTRODUCTION

Instant Messaging (IM) applications have been popular since America On Line Instant Messenger (AIM) was first introduced in the mid 1990's [1]. The popularity and social impact of IM applications, especially Mobile IM (MIM) applications has been very significant [2] [3]. For example, MIM use has been associated with job satisfaction and better job performance [3]. However, it is also known that mobile application selection is influenced by a person's age group and social setting [2]. IM applications tend to provide a fixed set of features and capabilities per application. That is, each application typically provides the base features (communication) and some additional features that encourage more users to adopt that application as their application of choice for communicating with others. Some applications, such as WhatsApp allow for more focused communication, others like Instagram allow for broader communication and so on.

Adopters of each application have no control over the technical features included in them. For example, although many of today's applications include emoji ediograms, users cannot generally add a unique set of such ediograms for use within a specific group (e.g. an organization may elect to design a unique set of such characters for internal communication). Another example is the use of privacy features. Current IM applications provide privacy features (e.g. encryption) but do not provide the ability for an organization or a group to install their own privacy solutions (e.g. their own encryption algorithms).

This paper introduces a modular MIM that allows users the freedom to tailor it by modifying core technical features. The use case focused on in this paper is that of privacy. The data of instant messaging applications are accessible and are frequently mined, sorted and sold. With recent exposures of how little privacy there is whilst communicating over the Internet, the need for a truly private chatting application is clear.

During the course of the development of the system described in this paper, WhatsApp introduced the end-to-end encryption feature to its clients on April 5th of 2016. However, there is no direct evidence that no third parties can decrypt the encrypted data. The questions arise primarily because the user does not have direct control over the technical features in the application.

The usual Instant Messaging (IM) process' logical model works as follows: a connection is created, a session is opened, and two types of messages handling exist: receiving and sending. Those messages have destinations, and pass through buffers and get saved in databases. This paper describes an altered model, in which the end user has much more control over the way that messages are protected. The model is instantiated in an Android IM application named "Dble-U".

II. EXISTING SYSTEMS

A. Typical features

A list of the features in popular smartphone messaging applications, is shown in Figure 1. The security features of these applications are varied. For example, encryption of data exchanged is feature number one [8], whether encryption/decryption is on client sides with or without the provider access to the keys. Yet, some applications (e.g.: Telegram) add to this feature the re-initiation of encryption keys after each hundred messages [9], or after a week of usage. Another security measure is timed-messaging followed by self-destruction of the messages. This is found in BlackBerry messenger [10], Telegram, SnapChat, etc. Another feature is the Whisper (Bleep) in which the message is destroyed just after being

read by the recipient [11]. Other applications made it more specific, by introducing a secret chat feature beside the normal conversations. In those secret chats, timestamps are hidden, nicknames are hidden, and messages can't be forwarded (e.g.: BBM). Moreover, some application support the users by an editing option where they can edit the sent message after being sent and retract the sent one, and sometimes delete the sent messages preventing the recipient from reading them. Some application disable the screen capturing feature too. Figure 2 summarizes the most important privacy features available.

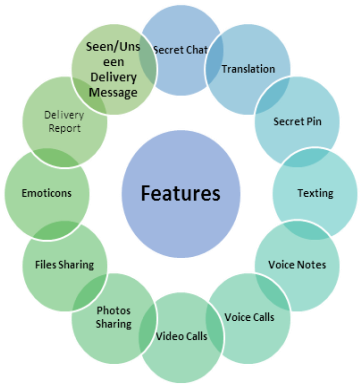

Figure 1 Main features of popular IM applications

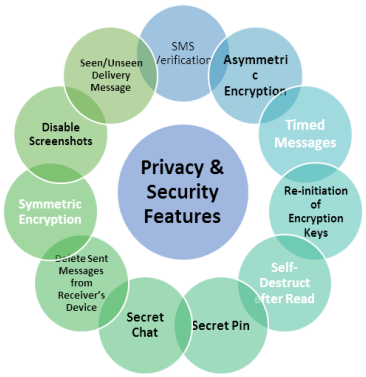

Figure 2 Privacy featured of popular IM applications

B. Design of typical IM systems

*a) **Server***

A sever is typically used for data exchange and saving in backend databases. Examples of such servers are: Microsoft Mobile Information Server (for messenger.) and BlackBerry Enterprise Server (BES).

Application Servers [4] are mainly used in 3-tiers web-based applications, where the third tier is the database end, the second tier is the Application Server, and the first one is the GUI at the client end.

*b) **Client***

There are various ways to implement an IM client. Typically these are native applications with the majority being native Android applications. However, not all of them support the aim of developing a secure chatting system. One popular way is to implement the system with server-to-client push notification using GCM (Google Cloud Messaging) [5], specifically its CSS (Google Cloud Connection Server) via XMPP (Extensible Messaging and Presence Protocol). XMPP has a Java library specific for Instant Messaging named Smack [6]. Yet, this means data will pass through the server, and most likely get stored on Google's cloud, which contradicts the aims of a privacy focused application.

The components of an instant messaging system in general follows a broadly similar logic, with the specific features concatenated later. The main component of the common logic between Android IM applications is the asynchronous processing mechanism [7] that organizes the chatting applications data transfer and communication. This mechanism is also used in the system described in what follows.

III. SYSTEM DESIGN

A. High level design

In "Dble-U", the privacy/security management of the Instant Messaging process takes place on a separate application. This system (Figure 3) is a combination of three applications, a web-based Privacy Controller, a tracking server and an instant messaging Android application. The high level design of the system can be classified as a 3-tier architecture, divided into a Client tier, Logic/Server tier and Data tier. However, this is used only for the establishment of a connection between clients, then the communication process becomes of 2-tier architecture (client tier & data tier/ Peer to Peer).

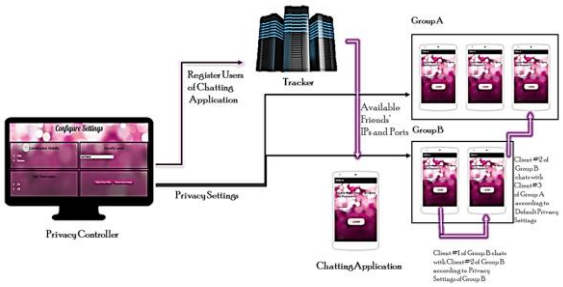

Figure 3 High level design

The existing prototype of Dble-U has three privacy options that an administrator of the Privacy Controller can choose values for: self-destruct, Hide/restore, and encryption level. The encryption level in turn has three options: no encryption, AES-256 with a fixed key, and RSA with a generated pair of keys. In the existing prototype, users are divided into groups and a user may belong to multiple groups. Each group then has a common security level set through the privacy controller.

As for encryption, each type has its own class and methods that are run when switched. If self-destruct option is turned on, the chat will disappear

just as the user leaves the chat. Similarly, if hide is turned on, the chat history will not show when the chatting panel is left, but will show when the restore option is chosen instead.

B. Design concepts

The Privacy Controller was designed as a separate application to guarantee the authenticity of its users. That is, the main concept being applied in this design is that of isolation. An Administrator can log in to the Privacy Controller using his/her credentials only, and can't configure the privacy settings of any Android application user unless the Android application user is added to the Administrator's list of Android application users. To add to the autonomy of the Android application user, an Administrator of the Privacy Controller can't configure any group's privacy settings unless all its members, who are Android application users, are connected locally to the Privacy Controller. This is applied as well to updating the privacy settings of an already existing group of Android application users.

Communication between the clients and the servers relies on HTTP rather than on directly using transport protocols. The reason is simple, wherever the World Wide Web is accessible, HTTP is accessible. There is a cost in terms of efficiency because of this choice, but the increased reliability was judged to be worth the cost.

Finally, the tracking server is only used as a directory to identify the network location of clients that wish to communicate with one another and is not involved in the communication chain between clients. This approach was chosen to enable easy connectivity between the clients, although it does create a weakness in the system in that if the tracker is somehow rendered inaccessible then the clients may not be able to communicate.

a. State diagrams

The following state diagrams illustrate the operation of the components of the Dble-U system and the transitions that take place between them.

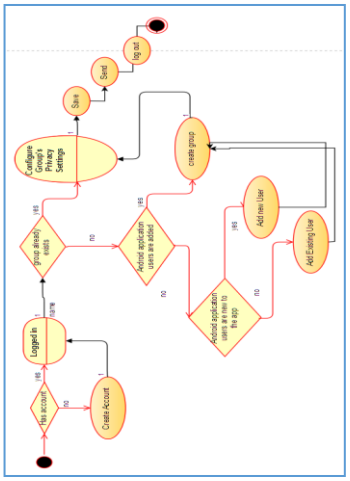

Figure 4 State diagram of the privacy controller component

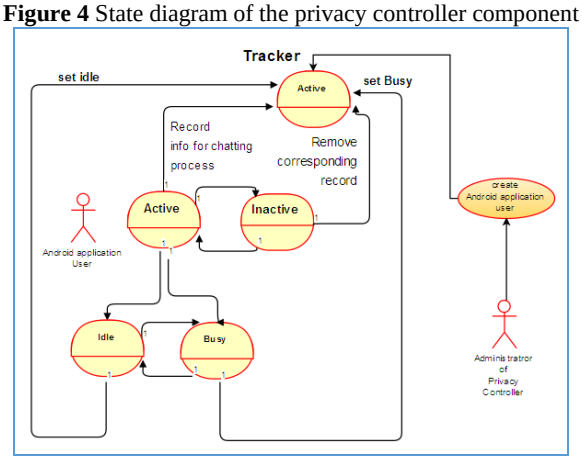

Figure 5 State diagram of the tracker component

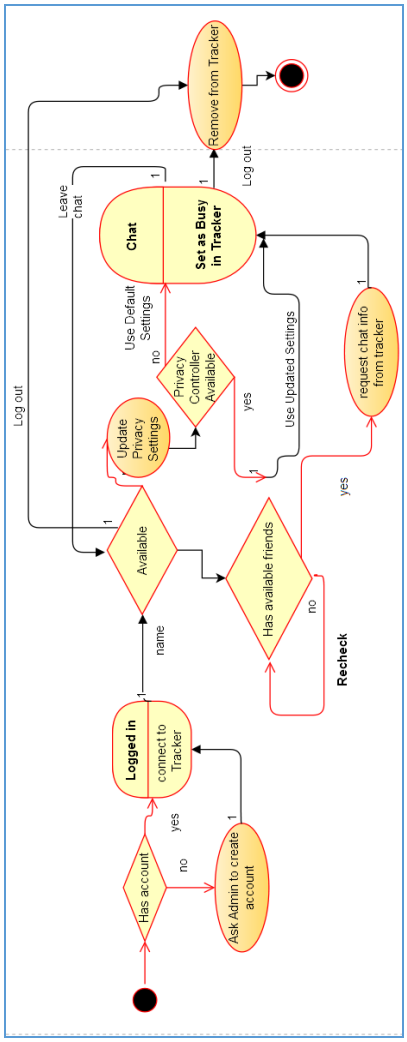

Figure 6 State diagram of the Android application

IV. IMPLEMENTATION AND TESTING

All three components of the chatting system described above have been implemented and tested. The testing process took place through a local network as a first step. Also, during the initial testing, the IP addresses had to be fixed to make the testing simpler, rather than changing it every time the Tracker and the Privacy Controller were disconnected from the local network.

The testing of the developed system focused on exercising all 3 components. All the states of the user Android application were tested. That is, users could communicate as part of multiple groups with ease, as expected, when they belonged to that group. The privacy controller allowed each group to have different privacy settings and these were observed to perform as expected during testing. That is, members of the groups would communicate with one another in an encrypted manner or in the clear. The messages were captured and analysed to substantiate whether or not the required level of encryption was taking place. The tracker was observed to be able to keep the clients aware of the IP addresses of the other members of the groups to which they belong. The speed with which the tracker updated its database depended on the frequency of the updates received from the clients. The clients send an update address message to the tracker every time a change in the IP address is detected. Other clients are in turn informed by the tracker of the change in address. The communication between the clients and the tracker are all HTTP based messages – allowing for access on almost all networks since this is the protocol on which the WWW is built.

V. CONCLUSION

As can be seen from the steps taken by WhatsApp, to introduce end-to-end encryption to its clients, privacy is a real user concern in IM. This concern will continue to grow over time as these applications continue to play a significant role in the way that people around the world interact with one another.

The solution described in this paper introduces a mechanism by which the user is given control over how the user messages are protected. This mechanism requires that the users know each other or are at least connected through a common point of contact. Although this means that users have a lower level of freedom when it comes to creating connections with other users, it is a necessary constraint for the increased level of privacy that it allows. Anonymity is valuable in Cyberspace, but in designing this system it has been assumed that the anonymity that users typically want is from prying third parties rather than from the people they intend to communicate with. By giving users the ability to apply multiple levels of security for different groups within which they communicate, the user is being forced to make a conscious decision regarding the cost of increased privacy and whether such a cost is worthwhile. That is valuable since it allows users to optimize their usage of their device's resources.

The described system is an instantiation of the modular design concept described in this paper, in which adopters of a system can have complete control over the mail technical features of the system. This may introduce some inconvenience to the adopter of a system (given that some knowhow will have to be developed regarding how to take advantage of this additional flexibility), but it may also be used in the production of different "flavors" of the same application.

REFERENCES

[1] G. E. Jacobs, "Instant Messaging and Texting," in *Handbook of Research on the Societal Impact of Digital Media*, IGI Global, 2016, pp. 493-527.

[2] R. M. Frey, R. Xu and A. Ilic, "Mobile app adoption in different life stages: An empirical analysis," *Pervasive and Mobile Computing*, pp. 1574-1192, 2017.

[3] V. C. Sheer and R. E. Rice, "Mobile instant messaging use and social capital: Direct and indirect associations with employee outcomes," *Information & Management*, vol. 54, no. 1, pp. 90-102, 2017.

[4] A. Srivastava and A. Bhargava, "Understanding application servers," January 2003. [Online]. Available: http://hosteddocs.ittoolbox.com/AS030504.pdf. [Accessed 24 February 2017].

[5] "Cloud messaging | Google developers," Google, [Online]. Available: https://developers.google.com/cloud-messaging/. [Accessed 24 February 2017].

[6] "Ignite Realtime: Smack API," [Online]. Available: https://www.igniterealtime.org/projects/smack/. [Accessed 24 February 2017].

[7] "AsyncTask | Android developers," [Online]. Available: https://developer.android.com/reference/android/os/AsyncTask.html. [Accessed 24 February 2017].

[8] J. Z. Florez, C. R. Logreira, M. Munoz and F. J. Vargas, "Architecture of instant messaging systems for secure data transmision," in *Security Technology (ICCST), 2016 IEEE International Carnahan Conference on*, 2016.

[9] "Secret chats, end-to-end encryption," Telegram, [Online]. Available: https://core.telegram.org/api/end-to-end. [Accessed 24 February 2017].

[10] "Get the New BBM for Privacy, Control and Much, Much More | Inside BlackBerry," 31 October 2014. [Online]. Available: Get the New BBM for Privacy, Control and Much, Much More | Inside BlackBerry. [Accessed 24 February 2017].

[11] "Twitch Introduces Whisper Feature," 11 June 2015. [Online]. Available: http://www.techtimes.com/articles/59663/20150611/twitch-introduces-whisper-feature-heres-how-to-use-the-private-messaging-system.htm. [Accessed 24 February 2017].