IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 2 (Mar. - Apr. 2013), PP 13-18
www.iosrjournals.org

Design and Implementation of Splash It!: A Secure and Privacy Preserving Splash Messenger

Nitish Sakhawalkar, Jaideep Thube, Shantanu Kulkarni, Nalini Mhetre, Shaziya Shaikh
Computer Engineering Department, STES's Smt. Kashibai Navale College of Engineering Pune, India
Team Lead, Life Sciences and Health Care Department, Persistent Systems Pvt. Ltd. Pune, India

ABSTRACT: *The main objective of any mobile application or service is to deliver a satisfying experience to its user, since it is the user who eventually decides to keep it or select another application from a plethora of choices. Without doubt, security and privacy both have a crucial role to play when it comes to using the application. Our application blends both these features offering users the ability to chat, share geographic co-ordinates and make 'Splashes' in real-time. A 'Splash Message' is a new Location-based feature that has been implemented in our application. Splash messaging enables the user to create messages on a virtual notice board at the location of his choice. The intended users will receive this message as soon as they enter this location. Splash It! provides users with the facility of marking their points of interest on the maps and sharing them with their friends. In addition, the user can control his level of privacy by making information available only to certain groups of friends or to all of them. Splash Messaging is our attempt to make existing chat applications more interesting. Currently, many implementations provide some of these features jointly or separately, through different applications, but are found to ignore the users' privacy. Our application on the other hand, provides an acceptable level of security by utilizing both asymmetric and symmetric cryptography, and most importantly, put the user in control of personal information. We discuss several issues here, such as system architecture, security, privacy, services provided by existing applications etc. Using a prototype application implemented in Google's Android platform, we demonstrate that the proposed system is fast performing, secure and privacy preserving.*

Keywords - *Splash Messaging, Chatting, Buddy Finder, Android, Maps, Points Of Interest.*

I. INTRODUCTION

Today, the number of smart phone users is growing rapidly. With this, is also increasing, the number of applications that could run on these devices. Various applications have been developed to provide effective services to the users. Mobile networking applications have enabled users to connect via their mobile devices irrespective of their locations. But the key issue lies in the performance of these applications, based on which the user will ultimately decide its usage. Security and privacy are the critical issues that any application needs to address. Although, there are chat applications that have useful features, they lack basic privacy requirements. Identifying the drawbacks in existing applications we aim to build an application that addresses the key issues of security and privacy. The proposed application aims to provide end user privacy, with a purpose to keep the user in complete control over personal data.

II. LITERATURE SURVEY

From personalized news, magazine applications to the social ones that connect hundreds of people irrespective of their locations, mobile applications have become a crucial part of our daily activities. Users can connect, interact, and share with other users on a real time basis. Thus, it can be seen that the simple text messaging format has a come a long way with the help of platforms like Android. The sharp increase in the number of Android smart phone users has led to the development of various social networking based applications. Developers are always finding new, innovative ways of connecting people together via an application, but at the same time keeping in mind the two very important aspects, which are security and privacy. The key issue for any mobile application or service is the way it is delivered and experienced by users, who eventually may decide to keep it on their software portfolio or not. Without doubt, security and privacy have both a crucial role to play towards this goal.

Today, the advances in wireless communication technologies and the proliferation of mobile devices have enabled the realization of pervasive and intelligent environments for users to communicate with each other, interact with information processing devices, and acquire ubiquitously a plethora of mobile wireless services through various types of access networks. In fact, nowadays the employment of mobile devices such as smart-phones for quick communication and collaboration is almost synonymous to their name. Driven by these factors

several categories of consumer mobile applications have emerged. One of them is conceptualized under the general term mobile social networking. This can be defined as a special kind of social networking whereon or usually group of individuals sharing similar interests and/or common pursuits, are communicating, conversing and connecting with one another using mobile devices. Until now, though many applications have been developed that focus on user's privacy, but lack basic crucial requirements. For example, several modern applications have following shortcomings:

- Support chatting and information exchange in general between their members but they finally turn out to be insecure.
- Message exchange between members may be in clear text, but the real identity of a participating user can be easily leaked out.
- In some cases, user can be tracked and profiled based on his/her actions and the services he/she acquires.
- Providers of such services are able to collect and keep for long time detailed log files on user's actions and sometimes sell them to advertising firms for profit.The following table shows a comparison between *Splash It!* And other applications. [13]

Security Requirements	GSM Based applications	*Whatsapp*	*BuddyMob*	*Splash It!*
User Authentication	No	Yes	Yes	Yes
Server Authentication	No	Yes	No	Yes
Confidentiality	No	Yes	No	Yes
Privacy	No	No	Yes	Yes
Splash Messaging	No	No	No	Yes

In any case however, a user who participates in a virtual community needs to rest assure that any information he/she sends and receives remains confidential and that her private sphere is not violated without the his/her consent. This should be the basic idea behind developing a secure application. In general, privacy is a complex concept that affects aspects such as location, identification and authentication.

The demand for truly privacy-preserving operation becomes even greater when location-based services come into focus. This is because location information is a set of sensitive data describing an individual's location in real time. Therefore, the underlying mechanisms should have the ability to prevent other parties from arbitrarily learning one's current position. Location privacy is about controlling access to this information, which is granted by the user who must be the only one responsible to decide if someone is going to have access to his/her location data or not. This is where our application *Splash It!* comes into picture.

Splash It! serves to be attractive to young people, lightweight, secure and privacy preserving. These characteristics allow *Splash It!* to be straightforwardly useful in a variety of environments with only minor modifications. Currently, the *Splash It!* system combines four services i.e., chatting, Buddy-Finder (BF), Points of Interest (PoI) locator, and Spatial Messaging (SM) into one. SM is an advanced wireless networking service that allows users to post a message on a virtual notice board somewhere on the map for someone else to collect. This service is also known with the terms "splash messaging" or "air-graffiti". Of course, the location-based services offered by *Splash It!* (i.e., BF, PoI, and SM) require the existence of a GPS receiver either built into the device or external. Nevertheless, this is not an issue today because the majority of mobile devices are shipped with a built-in GPS receiver.

Splash It! contribution is twofold, and different from other applications, shown as below:

- It utilizes both asymmetric and symmetric cryptography to provide a high-level of security to its users
- It respects end-user privacy by putting the user in control of what private information is revealed to other parties and under what circumstances.
- Additional features towards strengthening end-users privacy are anonymity and the absence of all kind of log files about their actions.

Splash It! has mechanism to protect privacy is of a user to hide his/her position from everyone or from certain people. It tries to combine chat and location-based services and at the same time provide communication confidentiality and user privacy. *Splash It!* follows a simple and lightweight client-server architecture where all communication passes through the server.

III. SYSTEM FEATURES

The application tries to combine chat and location-based services and at the same time provide communication confidentiality and user privacy. It follows a simple and lightweight client-server architecture where all communication passes through the server, which is supposed to be trusted.

3.1 CHATTING

Every registered user has the ability to add other users to their friend list. He can send to and receive text messages over the internet from his friends. He may choose to send text messages to an individual, a group of friends or may broadcast to all. If the recipient is online, messages are delivered instantly; but if he is not available, then the sent messages are queued on the server for delivery. These queued messages are delivered to the user as soon as he logs into his account. The entire communication proceeds through an encrypted channel.

The sender of the message also receives delivery reports, indicating that the message has successfully been delivered to the recipient. Each message is time-stamped and delivered accordingly. The server does not store message history, unless the recipient is offline. Thus, privacy is preserved. The client also has the facility to set his status, as either available, unavailable, offline, invisible, away etc. as per his choice. He may also set a custom status message for his friends.

3.2 BUDDY FINDER (BF)

The user can see his current location on the map. He can request to view the current location of his friends. On approval, the location is made available and can be viewed on the map. Users have the choice to control the visibility by deciding the accuracy of their location or can deny the request altogether. In the latter case, a fuzzy message is sent to the requestor citing some technical difficulties at the other end. The user may set a time interval, after which his location is updated. Locations of all the friends are available on the map, with markers and their usernames. These location packets are transferred through encrypted channels, similar to chatting.

3.3 POINTS OF INTEREST (POI)

This facility enables the user to mark his places of interest on the map. These places can be described in detail. The description, along with the location co-ordinates and user name are stored in the database on the server. This information may be edited or deleted by the author. He can also share these PoIs with other users to see. They may view the detailed description, and may also add the location as their own PoI. Each Poi can be made visible to a desired group of individuals, or to every other user.

3.4 SPLASH MESSAGING

A Splash Message is analogous to a virtual notice board, where a text message can be splashed at a specific location, for others to see on arriving at that location.

When a user creates a splash message, it is stored on the server along with the user alias, visibility, location and expiry time. The visibility of the message may include friends, or everyone arriving at that location. The message is available within the specified range of the source. The time for which the message shall be available, is decided by the user.

Any person arriving in the range of the Splash Message is instantly notified of the incoming message. The user may set the number of times he shall receive the message. By default, he will receive a new message only once daily, on entering the location.

There may be more than one splash messages in a single region defined by the boundaries in message specifications.

IV. SYSTEM DESIGN

As mentioned earlier in this paper, *Splash It!* tries to combine chat and location-based services and at the same time provide communication confidentiality and user privacy. *Splash It!* follows a simple and lightweight client-server architecture here all communication passes through the server, which is supposed to be trusted. In the following sections we describe each component more analytically. The system comprises of a server and multiple clients.

4.1 SERVER:

The server is basically an XMPP server (Extensible Messaging and Presence Protocol). CentOS host the custom OpenFire server to listen to client communications.

The server authenticates the client after receiving the corresponding XML packet from the client, and sets the status to available, unless specified by the client, explicitly. This status is broadcasted to all the users in his roster list.[10]

For text messages, the server uses store and send mechanism, so that messages to offline clients are not lost. The server manages to send these messages to the client as soon as he is online.
The custom OpenFire server uses a MySQL database for all its operations. It stores the Points of Interest (PoI) according to the user, along with the description. For tracking buddies, the server receives custom XMPP packets sent by the client, and forwards them to the client on the other side of the encrypted communication channel.

Each new Splash Message is identified with a message ID and the location coordinates and stored at the server database. As soon as the client enters the location, the message is splashed to him, and an entry added to the server database indicating that client-x has received the message. On expiry of the message, the splash message table triggers deletion of corresponding entries in all associated tables. Thus performance is enhanced, through reduction of time to splash the message.

4.2 CLIENT:

The client side is an Android device, preferably running Android OS 4.0 or above viz. above Ice Cream Sandwich. Every android device is equipped with Sqlite database for storing local data.

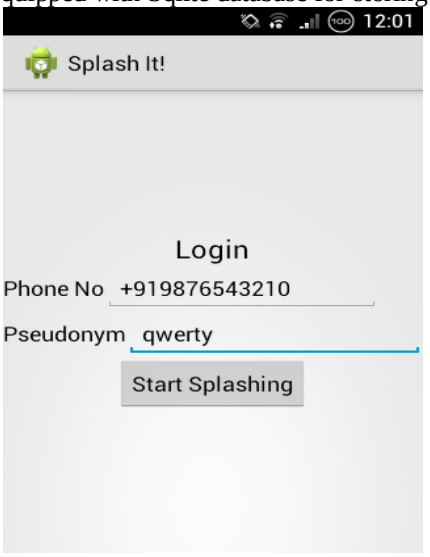

Figure 1.Client Mainscreen

After installing the application for the first time, the user is prompted to enter his phone number along with country code. This number serves as the username for the user, along with a random password generated at the server side. The user is also asked to enter his nickname, in order to maintain pseudonymity. He, however, has the freedom to change the pseudonym later on.

The Sqlite database manages local storage of chat messages received and sent. The message listener is a background service which performs the task of sending and receiving messages. Each text message is identified with a thread ID, to associate with the current chat session. The text message is encapsulated in an XML, and sent in encrypted form to the server.

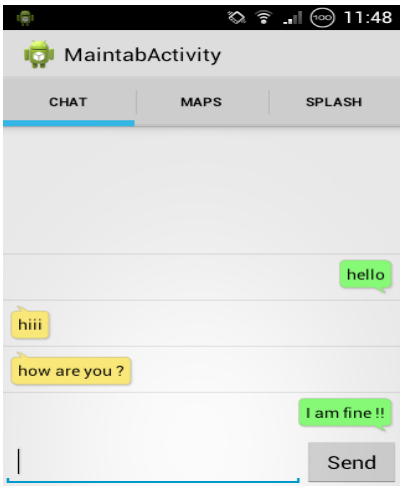

Figure 2.Chat screen

For buddy finding, the client sends custom XML packet over the server to the other client. The receiver analyses this packet and plots the corresponding location on its map. The client uses Google Maps v2, to obtain location and map details. The same map is also used with Show Buddies option turned off, for mapping the Points of Interest. The user sets a marker, which triggers a popup. A description is entered for the location, and stored on the server database, to be made visible to other users on selecting the marker.

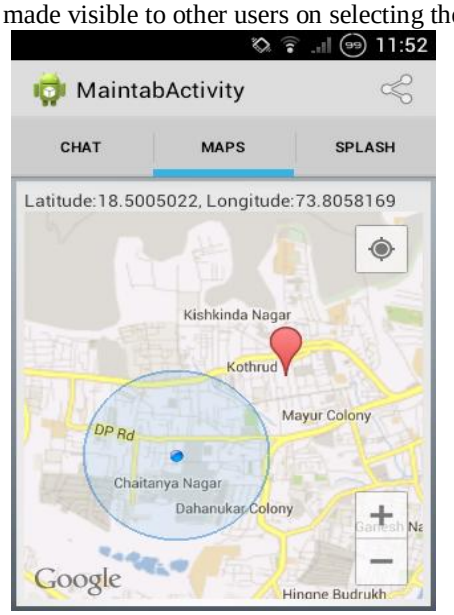

Figure 3.Maps tab

Splash Messaging fetches the current location of the user. The user also decides the radius within which this message should be available, along with an expiry for the message. These message specifications are sent to the server. For fetching the splash messages, a background service is running continuously, which checks for splash messages within current boundaries. It uses the haversine formula to obtain great circle from the stored latitude and longitude. The usage of haversine formula is optimum for performance and accuracy as desired.

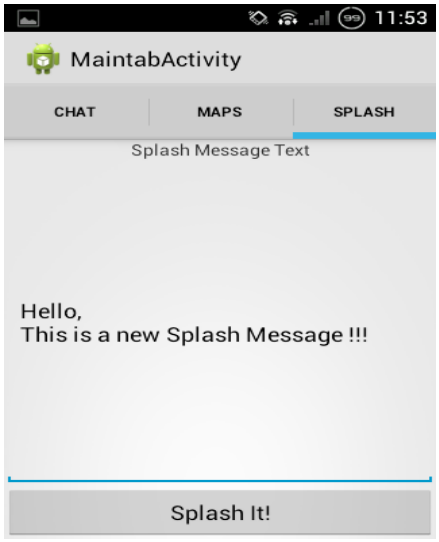

Figure 4.Splash messaging tab

V. SECURITY

The client uses Off-The-Record (OTR) [2] messaging to provide strong encryption for instant messages. As opposed to Pretty Good Privacy (PGP) Protocol, OTR provides users with better security and privacy. PGP uses long-lived encryption keys for maintaining confidentiality, and digital signatures for authentication, whereas OTR provides Perfect Forward Secrecy, which is essential for casual messaging.[6]
The encryption key is chosen using the Deffie Hellman Key agreement. Each time a new key agreement is performed, in order to achieve Perfect Forward Secrecy, by forgetting old keys.[6]

A Message Authentication Code (MAC) is used to authenticate each message. A MAC key is used for the same. This key is obtained by applying one-way hash function to the decryption key. The use of MACs to authenticate each message allows repudiation.

VI. CONCLUSION

Splash It! is useful for connecting with various users irrespective of their locations. It provides secure, real time messaging. The application ensures that the privacy of the user is preserved. It puts the user in complete control of personal data.

Splash It! serves as an on-the-go instant messaging application. It is useful for adventurous outings like trekking, hiking etc. to exchange locations. The application proves useful for a specific community, for e.g. In a University, to connect all the students. It can also be useful for rating different places like bookshops, hotels etc. using Splash Messaging feature.

Acknowledgement

The authors are grateful to Prof. P. N. Mahalle, Head, Computer Department, Smt. Kashibai Navale College of Engineering, Pune and other college professors for their timely support and guidance. We are also thankful to Persistent Systems Pvt. Ltd. for providing resources and their valuable expertise.

References

[1] Bruce Schneier, John Wiley & Sons, *Applied Cryptography.*
[2] OTR(Off-the-record) Protocol Description, http://www.cypherpunks.ca/otr/Protocol-v3-4.0.0.html
[3] Christof Paar, Jan Pelzl, A Test book for Students and Practitioners, *Understanding Cryptography, Springer*.
[4] Athanasius Loukas & Demetrious Demopoulos &Sofia A. Menesidou & Maria E. Skarkala &Georgios Kambourakis & Stefano Gritzalis, *MILC: A secure and privacy-preserving mobile instant locator with chatting, Published online: 18 August 2010# Springer Science+ Business Media, LLC 2010*
[5] Michel Deriaz and Jean-Marc Seigneur 1,*Towards Trustworthy Spatial Messaging, CUI University* of Geneva, Switzerland.
[6] Nikita Borisov, Ian Goldberg, Eric Brewer, *Off-the-record Communication, or why not to use PGP, WPES'04 October 2004*
[7] Free Software Foundation. Libgcrypt, http://directory.fsf.org/security/libgcrypt.html.
[8] IETF RFC 3920 Documentation, http://www.ietf.org/rfc/rfc3920.txt
[9] Smack API Documnetation, http://www.igniterealtime.org/builds/smack/docs/3.1.0/javadoc/org/jivesoftware/smack/package-summary.html
[10] Openfire Server Documentation, http://www.igniterealtime.org/projects/openfire/documentation.jsp
[11] Android Fragments Developer Documentation, http://developer.android.com/guide/components/fragments.html
[12] Working of Whatsapp Protocol, FunXMPP, https://github.com/TheKirk/WhatsAPINet/wiki/FunXMPP
[13] WhatsAPINet API Based on WhatsAPI, https://github.com/perezdidac/WhatsAPINet