Abstract – The Internet-based instant messaging services (e.g. MSN messenger, GTalk) have become the most popular communication choice among millions of people worldwide for communicating with family, friends and business associates. Due to the huge number of mobile subscribers, there is a real demand to provide such a service to mobile clients. Despite technology's significant advantages, the primary disadvantage is that only users who are on the same network can chat with each other on that particular network. For instance, a Yahoo! User can only chat with another Yahoo! User but not with the AOL user. Thus, we propose a mobile internet instant messaging system, IIMS, which integrates all popular networks like AOL, GTalk and Yahoo designed for mobile environments. This instant messaging system is expected to provide to its members an improved, single contact point, efficient, portable, easy-to-use, inter-network connectivity within a single application.

I. INTRODUCTION

With the rapid development the mobile Internet, mobile instant messaging (IM) systems are playing increasingly important roles in our life. However, it still faces with the issues of security and connectivity. XMPP (the Extensible Messaging and Presence Protocol), which is a cross-platform and open protocol, was adopted in this paper to resolve the connectivity problem of the IM system. And a hybrid encryption algorithm to secure the system was also designed, using the AES, SHA-1, and RSA algorithms to implement the hybrid encryption policy; the realization of the security algorithm was based on the Bouncy Castle encryption library. The whole system is composed of the Openfire server based on XMPP. The mobile client comprises four modules of the interface design, XML parsing, XML packaging, and event response. The main functions implemented in this mobile IM system include user login, roster, status change and display, sending and receiving of instant and encrypted messages, buddy management, chat record management and so on. This system is not only suitable for ordinary users, but also fit for enterprise applications. Instant messaging (IM) is a form of communication over the Internet that offers quick transmission of text-based messages from sender to receiver. In push mode between two or more people using personal devices, along with shared clients, instant messaging basically offers real-time direct written language-based online chat. The user's text is conveyed over a network, such as the Internet. It may address point-to-point communications as well as multicast communications from one sender to many receivers. More advanced instant messaging allows enhanced modes of communication, such as live voice or video calling, video chat.

II. STUDYING OF INSTANT MESSAGING AND PRESENCE

Instant messaging is a type of communication service providing users with two elements; presence information and real-time messaging. As it is an essential part of instant messaging it is necessary to provide a definition for presence.
as well. Presence is a means for finding, retrieving, and subscribing to changes in the presence information (e.g. “online” or “offline”) of other users.

**Presence Service**

This displays an overview of a presence service. A presence service has two distinct types of clients: ‘presentities’ and ‘watchers’. Presentities provide presence information to the presence service, while watchers request presence information about presentities from the presence service. Naturally, the same application can act both as presentity and as a watcher (Refer Fig 1).

![Fig 1: Presence Service](image)

This shows the watchers are classified as ‘subscribers’, ‘fetchers’ and ‘pollers’. A subscriber is a watcher that has subscribed to the presence information of presentity. The presence service keeps track of the subscriptions and sends a notification to the subscriber whenever the presence information of the subscribed presentity changes. A fetcher requests the presence service for presence information about presentity. The presence service does not send notifications to fetchers, presence information is only sent upon the request of the fetcher. A poller is a special kind of a fetcher that polls the presence service for presence information about presentity on a regular basis (Refer Fig 2).

![Fig 2: Different kinds of Watcher](image)

**Instant messaging service**

This shows that equally to the presence service, the instant message service also has two kinds of clients: ‘senders’ and ‘instant inboxes’. Senders are the source of instant messages to be delivered by the instant message service. An instant inbox is a container for instant messages that are to be read by the owner of the inbox. The instant message service accepts instant messages from senders and attempts to deliver them to the instant inboxes, to which they are addressed (Refer Fig 3).
III. UNDERSTANDING THE WORKING OF INTEGRATED INSTANT MESSAGING SYSTEM.

IMPP
IETF originally chartered IMPP (Instant Messaging and Presence Protocol) in order to define protocols and data formats necessary to build an internet-scaled instant messaging system. The working group managed to produce a model for presence and instant messaging in RFC 2778 and requirements for an instant messaging protocol in RFC 2779. However, as stated in above, the working group failed to achieve a common consensus for an instant messaging protocol. This resulted in the launch of several new working groups specifying protocols based on IMPP. It was decided that although the IMPP working group was not to specify any instant messaging protocol, it would carry on with its work. It would focus on producing standards for enabling interoperability between instant messaging systems.

JABBER
Jabber is the most widespread open source platform, using an XML encoded protocol, especially tailored to provide instant messaging (IM) and presence services over the Internet; however, Jabber is not designed just for this purpose, but several are the applications that may benefit and use the Jabber protocol suite. The protocol is totally free from legacy rights; both on the server and on the client side, which means that anyone can design its own Jabber client and even that any organization can freely implement an internal jabber server. Many are the advantages that come out from this approach:

1. The fact that the protocol is open lead to a better understanding of it, as everyone can learn from the work previously done and make available its code to other developers for the same purposes.

2. XML allows easy extensibility to the main features of the protocol. The Jabber Software Foundation accounts for the common extensions.

3. Decentralized approach. Since any organization can have its Jabber server, the resulting architecture is more scalable as lighter load is posed on the single servers, compared to a centralized approach.

XMPP
The Extensible Messaging and Presence Protocol (XMPP) is beyond doubt the strongest challenger to the SIMPLE standard in the Internet. Like SIMPLE, XMPP is also administered by an IETF working group, i.e. the XMPP working group. XMPP has been formed from the basis of the Jabber protocol. The Jabber protocol is the result of a research started by Jeremie Miller in 1998. The goal of the research was to produce an interoperable and open instant messaging protocol, an alternative to the proprietary solutions. The first public release of the protocol took place in May 2000. In June 2000 research members sent an Internet Draft of the Jabber protocol to the IMPP working group as an instant messaging protocol proposal. However, the organization of the Jabber research was not mature enough at the time and the Internet Draft was left to expire. In 2001 the Jabber Software Foundation was formed to organize the researchs and commercial bodies involved in the Jabber community. Following the reorganization, a new Internet Draft was submitted to the IETF in February 2002, eventually leading to the birth of...
the XMPP working group in October 2002. XMPP is in essence the core of the XML (Extensible Markup Language) based Jabber protocol. XMPP has been made IMPP compliant by the working group. The Jabber Software Foundation will continue to work on the parts of the Jabber protocol that are not part of XMPP or IMPP, exploring features such as: multi-user chat, calendaring and whiteboarding.

IV. IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system. Many implementations may exist for a given specification or standard. In implementation of a software research the design model prepared is converted into a robust, reusable code using appropriate coding language. The implementation process runs along with the unit testing for checking the correctness of each coding unite. Different modules of the software are implemented separately and after appropriate testing these are integrated with each other logically to form the entire product.

LOGIN

- The user must enter the account username, which is his JID (jabbered). The jabber id could be a Gmail username as well.

<< private void checkUsername (String username){} >>

- The user must enter the account password.

<< private void checkPassword(String password){} >>

- If the user is using a proxy server, he must enter the IP address of the server.

- The IIMS Application uses xmppConnectionAdapter to create a socket connection to XMPP server. This is a default connection to a Jabber/gmail server.

- The jabber/gmail server authenticates the username and password.

IIMS CONTACTS

- Once the user is authenticated.

  - The buddy list is retrieved from the existing accounts database from the Xmpp server.

  - If the buddy/contact list is empty the user can add contacts to the buddy list using the add contact option in the menu.

- Buddy list is the dialog showing the buddies associated to the connected accounts. Additionally the status of active connections is displayed using the presence service.

<<public class PresenceAdapter >>.
This class imports a smack library from the jive software.
<< import org.jivesoftware.smack.packet.Presence; >>

- The user can scroll the contact list using the android.app interface.

- The contact can be added to specific groups. When adding a contact not already existing in the buddy list the contact name, alias and group is to be specified.

ADD CONTACT

If the user is added via the IIMS application.
Contacts are added using the AddContact.java file. The user must first click on the menu button on the bottom and select addcontact.
On click (onClick(view v)) the add contact layout appears. The email id is entered in the contact text box. The email id is matched to check whether the email id is valid.

**OTR MESSAGING**

Off-the-record Messaging, commonly referred to as OTR, is a cryptographic protocol that provides strong encryption for instant messaging conversation. OTR uses a combination of the AES symmetric-key algorithm, the Diffie-Hellman key exchange, and the SHA-1 hash function.

- The iims application provides a support to have private conversations over instant messaging by using off-the-record (OTR) Messaging feature.
- While chatting both the user can start an OTR session by using the OTR actions from the chat menu.
- When the OTR session is activated from both the users the text gets encrypted on the communication channel.

**V. CONCLUSION**

We had brief research of android app development, development tools and the components of the research that we have to develop. Android is a disruptive technology, which has introduced initially on mobile handsets, but has much wider potential. Android is a big revolution in the field of operating system. Their custom virtual machines optimize memory and hardware resources in a mobile environment. It can be liberally extended to incorporate new cutting edge technologies as they emerge. The open source platform works together to build innovative mobile applications. The system supports any number of concurrently connect clients, which are synchronized so that a client can connect and disconnect at any time without losing any messages. The system is designed to inter-operate with different current instant messaging systems such as ICQ or MSN Messenger. It can run on different platforms with different capabilities such as tablets and mobile phones and can work over temporary low bandwidth connections. It illustrates some of the design principles that are necessary in a mobile environment. Thus using android tools we can implement mobile integrated instant messaging system and achieve various features.

**REFERENCES**