LiveS Cube: An Experiment for Mobile Social Network

Yu-qian Li[1], Yang Liu[1], Zhi-fang Liu[1], Chao Liu[1], Zhao-Nan Li[1],

Fu-ye Han[1,3], Jun Li[1,3], Ming Xu[1,3], Xin Guan[1,3], Zhen Chen[2,3]

Department of Computer Science & Technologies[1]
Research Institute of Information Technology[2]
Tsinghua National Laboratory for Information Science and Technology (TNList) [3]
Tsinghua University, Beijing 100084, China

***Abstract*—Contacts Address Book (CAB)'s maintenance is a big problem because each user has to deal with hundreds of contact information in his or her address book. Data Lock-in also hurdle their desire to change mobile platform for new applications. In this paper, we solve this problem by exploiting the online social network (OSN), making mobile users maintain their own contacts information and still get every friend's most recent information by sharing the information through the social network. We transform the existing contacts address books function with social network aware function and build a social network since a huge network with enormous number of nodes (users) and edges (relations) could be built very quickly as there are billions of mobile users and almost every mobile user has an contacts address book inside the mobile phone and that address book is the real relationships (contacts). This social network is named LiveS Cube because its design objective is making mobile devices simple, secured and shared (LiveS3). We deploy the LiveS Cube in the real campus environment for about a half year and collect interesting data and problems. The implementation of this social network is also presented and some unexpected problems came out and it turned out to be quite difficult to be built from scratch.**

Keywords-Mobile Social Network, Contacts address book, Prototype, LAMP, Mobile Platform.

I. Introduction

Social networks have attracted billions of active users under major online social network (OSN)[1,2] systems such as Facebook, MySpace, LinkedIn, RenRen, QQ, etc. These systems allow people with common interests to come together and form online communities. Nowadays the social networks are increasingly accessed via mobile devices thus rendering a new research field of mobile social networks (MSNs) [3,4]. A few interesting results of research and development of MSNs have been reported and concluded in literature [6]. The majority of them focus on online applications running on mobile devices and pay little attention to transforming the underlying mobile client's existing function such as contacts address book to be socialized and the specific features of mobile phones can be utilized to augment the OSNs from traditional stationary PCs to mobile devices.

The main contribution of this paper is a system design and implementation of LiveS Cube, a Mobile Social Network based on common contact address book in most of mobile platform. We transform the common used contact address book function with a social network-aware function, and will automatically retrieve and update the mobile user's information and make this information visible to his or her friends in OSN. This transformation's main benefits for mobile users are the reduced contacts' maintenance and alleviate the data-lock-in cost when they change mobile phones.

In this paper, we will discuss our motivations to design and implement LiveS Cube. Section 2 presents the system architecture and implementation of LiveS Cube, the deployment in a campus network is given. Some discussions and experiences are given in Section 3. Finally, in Section 4 we conclude this paper.

II. LiveS Cube

In this section, we present the system architecture and the implementation of LiveS Cube system. Figure 1 shows our system navigation route.

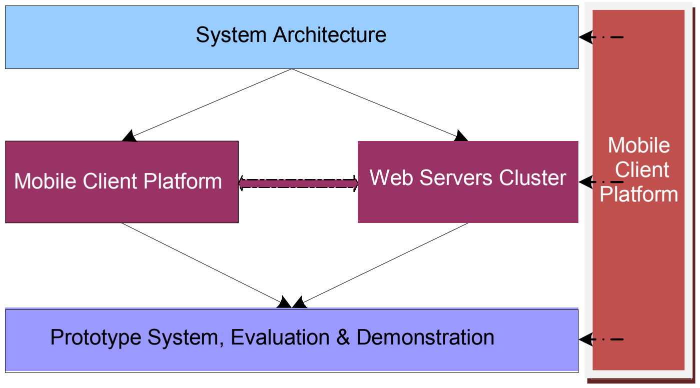


Figure 1. Navigation route

A. *System Architecture*

Mobile Social network consists of three critical components, i.e. physical social network, Mobile Client Platform, and Rendezvous Web Server Clusters. A basic principle of LiveS Cube is shown in Figure 2.

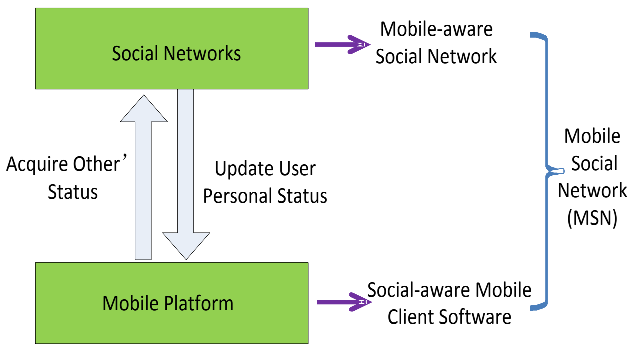

Social Networks
Mobile-aware Social Network
Acquire Other' Status
Update User Personal Status
Mobile Social Network (MSN)
Mobile Platform
Social-aware Mobile Client Software

Figure 2. Mobile Social Networks LiveS Cube

In consideration of diversity of mobile platform, the contacts address book client software needs to be ported to a lot of mobile platform, such as Nokia Symbian OS, Google Android, and Apple IOS etc. Figure 3 gives more detail about the technical implementation of LiveS Cube.

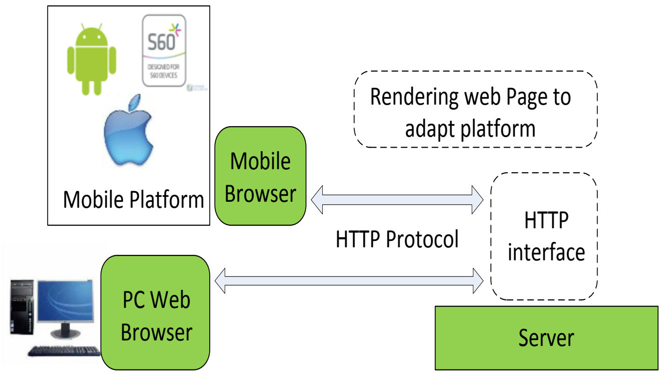


Figure 3. LiveS Cube's Scheme

B. Implementation

1) Mobile Client Side

We implement our new contacts address book client in mobile platform such as Nokia Symbian S60 and Google Android to simulate the original contacts address book in mobile phone, and make less modification in UI as possible.

2) Web Server Side Frontend

On the Web Server, PHP provides four major functions which are registration, searching, checking and editing personal information related with one-way users information.

Common	CSS, Banner, Navigation Bar, Official Website Link, Copyright information, User panel
Home Page	Search bar, high-level search, label, Recommended business list
Search result	Telephone number, Name, State text(*abbreviated*), User information(*abbreviated*), Label
Business information	Telephone number, Name, State text, User information, Label
Modified information	Telephone number, Name, modified password, confirmed password, State text, User information, Label, Submit button
Others	Login box, Information box

Table 1. Content list on the Web Server side

As can be seen for all one-way user information, so all the one-way user's information (including categories, business information and other fields that two-way user does not have) is stored in plaintext data table in the database. Besides, in order to make the mobile client get the one-way user information conveniently, the basic information (telephone number, name and state text) is also simultaneously stored in ciphertext information tables, through the Virtual Client encryption module to achieve.

Because LiveS Cube is mainly designed for mobile users, WAP page is provided for mobile browser that cannot parse HTML pages. The WAP pages are available users to search information anytime and anywhere, especially for the OSes which cannot use our client software.

Besides, on the Web Server, we implement the function of extracting and pushing information automatically. User can add some pre-customized virtual one-way user (jokes etc.) or one-way user group (news etc.). These virtual one-way user's states can be updated by Web Server regularly, means while, the content of states are extracted from the Internet RSS resource by server.

3) Web Server Side Backend

Server side consists of two components: One is pure encrypting and decrypting back-end module, which is designed to implement by SafeCore in order to guarantee security. Currently, it has been implemented by a software module implemented by C/C++. Another part is the front-end processing module, which is implemented by PHP, coordinates with Apache/Mysql to handle HTTP requests and split them into some basic processing unit after parsing them, then send them to encryption and decryption module to handle and write them into database, or read encrypted data from database, decrypt it with the encryption and decryption module, package it and return the result to user. Encryption and decryption module is designed as simple as possible that can process simple requests, which are scheduled and built up by front-end module, can achieve completely all the logic functions.

On the server side, considering the system's stability, we use the Apache2 / PHP / Mysql as the basic server development tools, regardless of the requests originated from the Web user or from mobile client to access our server. The Apache2 back-end dispatches the operation in form of HTTP request to PHP program, and creates a real-time process to handle requests.

As noted above, server-side program development is based on PHP and MySQL. A number of encryption and interacted with SafeCore operation will be implemented by C/C++ to ensure flexibility and efficiency. The C/C++ component is mainly in the form of the process called by PHP program and communicates binary data in the form of files on the ramdisk.

As shown in the figure below, Server side includes 6 main modules, including SMS text messaging service module is that we have not yet implemented. HTTP module is the front-end server, which will handle HTTP requests to classify, and generate a corresponding request to downstream components. The Encrypted Interface and Open Interface which places the middle layer is logic layer of encrypted requests and non-encrypted requests. The non-encrypted request is the Web-initiated request for one-way user information, and an encrypted request is a mobile client request to the ordinary user. SafeCore which is the core of our server hardware places in this layer of encrypted interface. Between the logic layer and the front-end, there is a small Virtual Client module; this module can make some plaintext data operation to disguise as the ciphertext data operation that sends to Encrypted Interface. It ensures that a user information item in plaintext database has a corresponding item in the cipher text database, so that clients can access them undifferentiated. Finally, there is a database operating layer in the bottom which provides facilitate database operations to the up logic layer.

4) *Communication Protocol Design*

USER REQUEST OP CODE DEFINITION

Server classifies and judges the request based on op codes and the domain specified by the PHP homepage.

Domain	OP	PHP file related	Function
server.lives3.net (Mobile Phone Server)	11	mobil_register.inc.php	two-way user registration
	12	mobil_update.inc.php	Updated user information
	13	mobil_getUpdatePackage.inc.php	Updated contact information
info.lives3.net (Web Server)	22	web_search.inc.php	Searched one-way users
	23	web_profile.inc.php	Displayed one-way user information
	24	web_register.inc.php	One-way user registration

Table 2 . Op Codes and the Domain Specified

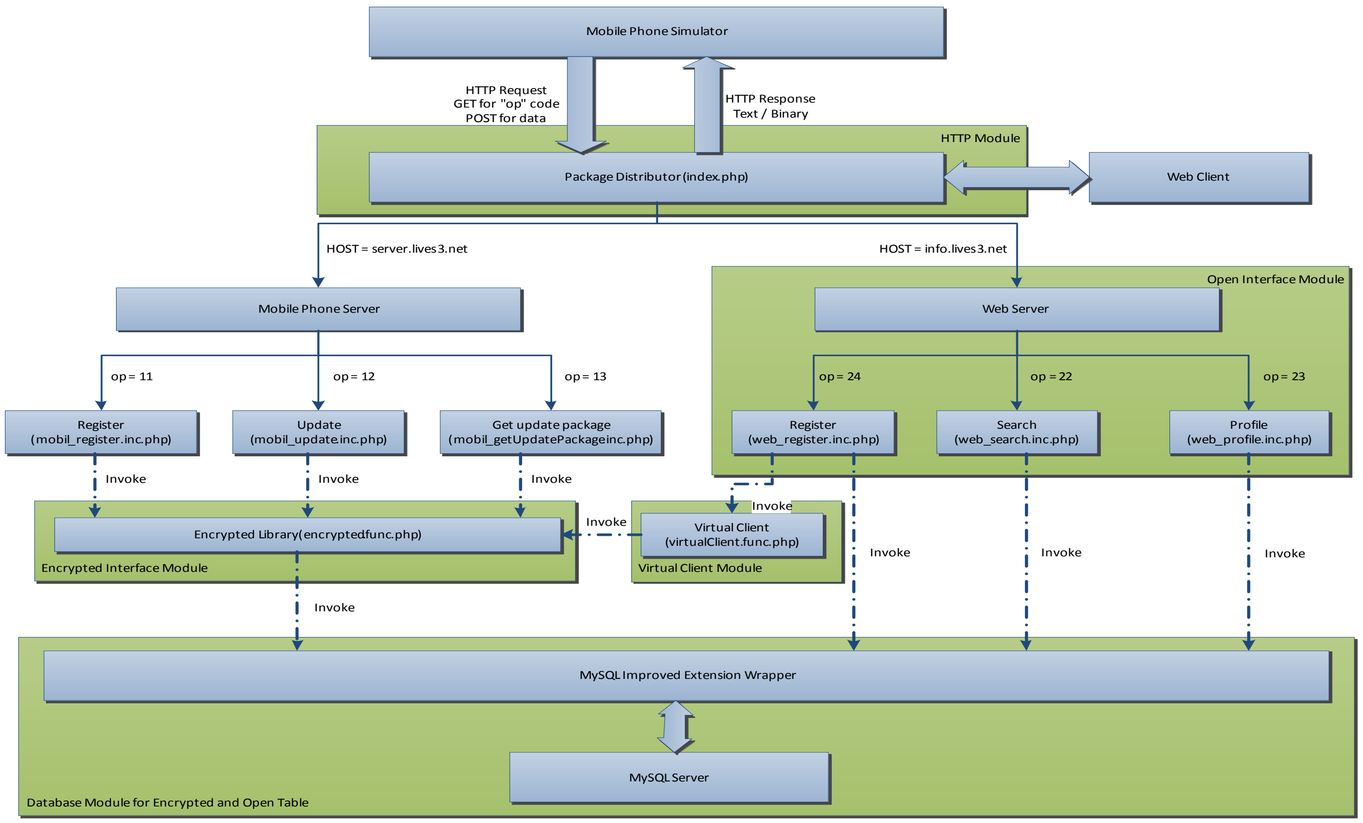

Mobile Phone Simulator
HTTP Request
GET for "op" code
POST for data
HTTP Response
Text / Binary
HTTP Module
Package Distributor (index.php)
Web Client
HOST = server.lives3.net
HOST = info.lives3.net
Open Interface Module
Mobile Phone Server
Web Server
op = 11
op = 12
op = 13
op = 24
op = 22
op = 23
Register (mobil_register.inc.php)
Update (mobil_update.inc.php)
Get update package (mobil_getUpdatePackage.inc.php)
Register (web_register.inc.php)
Search (web_search.inc.php)
Profile (web_profile.inc.php)
Invoke
Encrypted Library (encryptedfunc.php)
Encrypted Interface Module
Virtual Client (virtualClient.func.php)
Virtual Client Module
MySQL Improved Extension Wrapper
MySQL Server
Database Module for Encrypted and Open Table

Figure 4. Server Side Functions Blocks

INDEX GENERATION FOR QUERYING MOBILE PHONE NUMBER

Mobile phone number index is generated by SafeCore. Its input is a size of 256 bytes of binary encrypted information. The 256 bytes of information is generated by the client. Encryption processes as follows: Appending 24 bytes of random number to 8 bytes of mobile phone number, and encrypting the generated 32 bytes of information with SafeCore, and resulted in 256 bytes encrypted information. The encrypted information can be decrypted only with SafeCore private key, so this information cannot be intercepted without SafeCore.

Firstly SafeCore decrypts the input encrypted information with private key. Then SafeCore removes 24 bytes of random number, and obtains the original phone number. Filled with a string with 8 bytes of zero, we get 16 bytes of information. The 16 bytes of information is also encrypted with AES symmetric key, which is also generated by SafeCore's private key. We get 16 bytes of encrypted information, and obtain the index. As 8 bytes of zero in padded in the 16 bytes before symmetric encryption, an attacker cannot get original mobile phone number by using the index value with reverse encryption process. More theoretical details are presented in [6].

REGISTRATION OR MODIFICATION OF USER PERSONAL INFORMATION

When users register or update their personal information, they can encrypt original personal information with owned AES symmetric secret key, which is also encrypted with the public key of server. If they want to update personal information, users must encrypt information using the original private key, in order to pass server identity verification. However, users may envelope a new key in their information, in order to update the older keys. For new users, they can choose a symmetric key as their initial symmetric keys.

When updating user's personal information, SafeCore is feed with the encrypted update information A from client, and original user information B which generated by query server database index. Each user information in the server database (all in cipertext) is encrypted with AES symmetric key in SafeCore as same as which used in index process, and so does B. The index used to retrieve B from server database is also generated by SafeCore.

After receiving A and B, SafeCore decrypts B first, then checks the symmetric key in A using the symmetric key stored in B. After that, SafeCore decrypts A to obtain the plain user information in A with symmetric key which decrypted by SafeCore private key. After receiving the plaintext, SafeCore encrypts user information again with its symmetric key, and sends back to database.

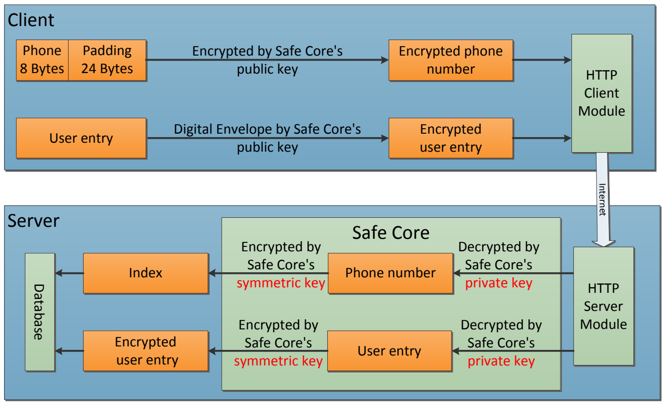

Client
Phone 8 Bytes
Padding 24 Bytes
Encrypted by Safe Core's public key
Encrypted phone number
User entry
Digital Envelope by Safe Core's public key
Encrypted user entry
HTTP Client Module
Internet
Server
Safe Core
Database
Index
Encrypted by Safe Core's symmetric key
Phone number
Decrypted by Safe Core's private key
HTTP Server Module
Encrypted user entry
Encrypted by Safe Core's symmetric key
User entry
Decrypted by Safe Core's private key

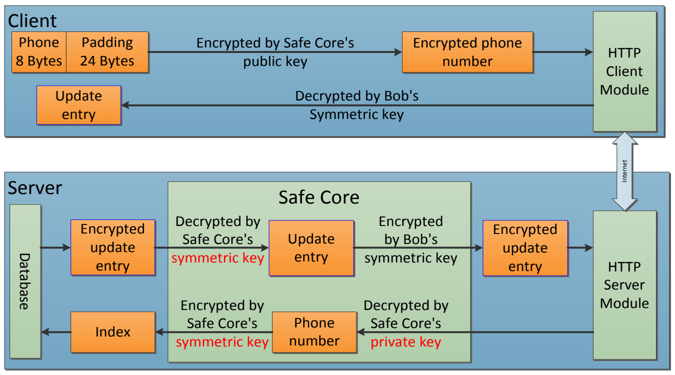
Figure 5. Retrieval of User's Personal Contacts' Information

RETRIEVAL OF CONTACTS' INFORMATION

When user A wants to acquire the information of his or her contact B, he or she need send encrypted user information with a time label t of A and B to SafeCore. The SafeCore will output B's updated information C and a boolean variable u. The variable u indicates whether the update time of B is after time t or not. So that the system on the upper layer can make an incremental computation to user's a group of requests, and return the exactly B's updated information.

Receiving user's information, first of all, we decrypt the information of user A and B. Then we determine whether user A can get user B's information. In other words, only if B's contacts contains user A's number, or users B is an one-way user, user A can obtain B' information. (User A send requests for obtaining the user B's information, so we assume that B's number in A's address book). If user A pass the verification, SafeCore will grab B's name and B's state (in plaintext), then encrypt these information with A's public key to get C and generate an output in order to ensure that the output information only can be decrypted with A's private key in client A.

The system will separately deal with the access request for acquiring a group of contacts (currently a group contains at most 16 encryption numbers, their index is queried from database and sends these items to SafeCore. It judges the time increment interval from the returned results of SafeCore, and pack the updated information to the upper system.



Client
Phone 8 Bytes
Padding 24 Bytes
Encrypted by Safe Core's public key
Encrypted phone number
HTTP Client Module
Update entry
Decrypted by Bob's Symmetric key
Internet
Server
Database
Encrypted update entry
Safe Core
Decrypted by Safe Core's symmetric key
Update entry
Encrypted by Bob's symmetric key
Encrypted update entry
HTTP Server Module
Index
Encrypted by Safe Core's symmetric key
Phone number
Decrypted by Safe Core's private key

Figure 6. Synchronization of contact information in LiveS Cube

III. PROTOTYPE

A. *Deployment*

We have constructed and deployed a web cluster for practical experiment application. The Web cluster consists of one load balancer and two servers, which provides better scalability.

Symbian S60 client: http://www.lives3.net/blog/s60/intro/
Android client: http://www.lives3.net/blog/android/intro/

B. *Lessons and Constraints*

Here we give some discussion on our implementation and the lessons we learn from practical deployment.

In Development Phase, we found, for mobile user, mobile phone software are constrained by its mobile platform. Google Android is much easy for use compared with Symbian[10], because S60v3 devices does not have touch screen support, and S60 SDK provides less APIs about UI. There are also difficulties in download and setup because S60 has no popular AppStore support like Android Market.

We have reduced the client software with the minimal functions, but development in Symbian S60 is still challenging task. e.g. in Symbian S60 C++ SDK, there are about 20 classes used for describing a string. The garbage collection scheme is intriguing: compared with Symbian, the ANSI C++ new/delete or malloc/free is much simpler.

Our expectation is by sharing in social network, each one using contacts address book will be much easier, so the value of our social network is improved. But when the social network starts up as a humble root with a small volume of users, it has not been the potential to possess a large precious resource where its user will spontaneously and voluntarily to offer for its user. Hence we need to maintain some useful and interesting information to attract the user and make it trustworthy for a common user.

IV. CONCLUSION

Contacts Address Book (CAB)'s maintenance in mobile phone is considered as an intrigue issue. In this paper a social network named LiveS Cube is constructed by transforming the existing contacts address books function with social network aware function. The new contact address book will automatically retrieve and update the mobile user's information and make this information visible to his or her friends in OSN. This transformation's main benefits for mobile user are the reduced contacts' maintenance and alleviate the data-lock-in cost when they change mobile phones. Hence, the LiveS cube has great potentials to connect a large volume of person.

ACKNOWLEDGEMENT

This work is supported by Ministry of Science and Technology of China under National 973 Basic Research Program (grants No. 2011CB302805).

REFERENCES

[1] Alan Mislove et al., "Measurement and analysis of online social networks," IMC' 2007.

[2] Ravi Kumar, Jasmine Novak, Andrew Tomkins, "Structure and evolution of online social networks ," KDD'2006.

[3] Olafur Helgason, Emre Yavuz, Sylvia Kouyoumdjieva, Ljubica Pajevic, and Gunnar Karlsson, "Demonstrating a Mobile Peer-to-Peer System for Opportunistic Content-Centric Networking," Sigcomm MobiHeld' 2010.

[4] Te-Yuan Huang, Kok-Kiong Yap, Ben Dodson, Monica S. Lam, and Nick McKeown, "

[5] PhoneNet: a Phone-to-Phone Network for Group Communication within an Administrative Domain," Sigcomm MobiHeld' 2010.

[6] MSN wikipedia, http://en.wikipedia.org/wiki/Mobile_social_network

[7] Yuqian Li, Yang Liu, Zhifang Liu, Jiwei Huang, Zhen Chen, "The Power of Refresh: a Novel Mechanism for Securing Low Entropy PII," CMC'2011.

[8] S. Buchegger and A. Datta, "A case for p2p infrastructure for social networks - opportunities & challenges," in WONS'09: Proceedings of the Sixth international conference on Wireless On-Demand Network Systems and Services. Piscataway, NJ, USA: IEEE Press, 2009, pp. 149–156.

[9] S. Buchegger, D. Schi¨oberg, L.-H. Vu, and A. Datta, "Peerson: P2p social networking: early experiences and insights," in SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems. New York, NY, USA: ACM, 2009, pp. 46–52.

[10] Randy Baden, Adam Bender, Neil Spring, Bobby Bhattacharjee, Daniel Starin, "Persona: An Online Social Network with User-Defined Privacy," Sigcomm WOSN'2009.

[11] Marti Motoyama, George Varghese, "CrossTalk: Scalably Interconnecting Instant Messaging Networks," Sigcomm WOSN'2009.

[12] ZDnet, Nokia, Microsoft in Windows Phone alliance - Communications - News, 2011.