A Study on Context Services Model with Location Privacy

Hoon Ko[1], Goreti Marreiros[1], Zita Vale[2], and Jongmyung Choi[3]

[1,2] Institute of Engineering Polytechnic of Porto, GECAD,
Rua Dr. Antonio Bernardino de Almeida, 431, 4200-072 Porto, Portugal
{hko,goreti}@isep.ipp.pt, zav@isep.ipp.pt
[3] Mokpo National University,
Muangun, Jeonnam, S. Korea
jmchoi@mokpo.ac.kr

Abstract. A lot of smart context-aware services would be adopting location information as context information. However, the location information is also very important information to be protected for users' privacy, security, and safety. In this paper, we propose One Time Password (OTP) in the communication between users' devices and network devices such as APs. By using this approach, APs does not keep user specific information but OTP values, so that attackers cannot get user information even though they access to the log files in APs. We also introduce context-aware service scenario and context information for the service.

Keywords: Context-Aware, Context-Services, Location Privacy, OTP (One Time Password), Security.

1 Introduction

Micle Altschul, a corporate lawyer of Cellular Telecommunications & Internet Association (CTIA), which is an international association of Wireless Communication Enterprise and Wireless Services Provider, warned the privacy problem in mobile communication. He advised not to use commercial mobile services through open Wi-Fi hotspots such as Starbucks WiFi [1]. This is because the policies and technologies for privacy protection are different among the service providers. The simple use of WiFi can reveal some user's privacy information such user's location. Actually, Sky Hook developed software that can calculate user's location in 20 meters radius using triangulation method of WiFi hotspots [1]. Another example is WaveMarket. This company has cooperated with wireless service providers and has been providing the location-based services which tracks the locations where family members or friends area. The most popular method for protecting location privacy is to use informed consent policies, in which companies let users know what information they get about users and users' locations and get consent for that. Although they try to use these policies, some arguments have been continually being issues, because disobedience of these policies may break all the privacy protection. The problem is that they have hardly mentioned technical issues for location privacy in WiFi. Assume that attackers get and analyze log files or configuration files in network devices, and then they can get users' information including user locations and their routes [2].

A M. Tjoa et al. (Eds.): ARES 2011, LNCS 6908, pp. 321–329, 2011.

In this paper, to solve location privacy problems, we propose a method of using One Time Password (OTP) in the communication between a user's device and network devices. Once the user moves from one place to other place, all user information will be erased in the log files, and only OPT values are left. This approach is totally different from the existing systems, in which all information is kept in their log files. Therefore, if attackers look inside the log files, they cannot get user specific privacy information. The only information they can get is the OTP values. If network devices have some troubles, the administrator can check the log files and the OTP values to find out the causes of the troubles, but he/she cannot identify users from the OTP values, either. The remainder of the paper is organized as follows. In Section 2, we discuss other research that is closely related to our work. Then we define the location privacy in context-aware systems in Section 3. After that we show some discussion issues for the location privacy in context-aware systems in Section 4. Finally, we reveal the conclusions of our work in Section 5.

2 Related Work

John Krumm [1] explains location privacy problem of services which provide user location information. In his paper, he defines that protecting from let other users know where user is and where path user had can be function and availability. He suggests some solutions to solve using anonymity, spatial / temporal degradation, specialized queries, and configuration privacy. However he does not consider encryption and access control. Therefore, if attackers access the log files, they can get user information rather easily. Hulsebosch et al. [2] introduce location privacy problem from anonymous accesses for users and user contexts on ambient context. In their paper, they suggest a process distinction at user location through a security level definition.

Garreti Brown and Travis Howe [3] propose a solution of social network and context-aware spam.

In their work, they mention problems of the location based service of Facebook user, and they argue unnecessary context named context spam receiving problem, by exposing of the user location. Those three researches introduce location privacy problems by their own approaches [8], but they have a common assumption, which is that if users do not leave their information in network devices such as servers and access points (APs), then users' routes or their location privacy do not matter. However, during the process for mobile communication, the mobile devices have to get helps from APs and network devices around users, and these requests cause privacy information to be open [9]. Currently, most of smart mobile devices such as iPhone and iPad are connected to the network via wireless APs and other network devices in the public area, and the users' location may be revealed by analyzing the log files of these network devices.

3 Location Privacy in Context-Aware System

3.1 Location-Based Context-Aware Service Scenario

Context-aware systems provide intelligent services, and most of them adopt user location as context information. In our work, we introduce a context-aware service which uses location information, and we show location privacy protection method.

A. One-Time Password (OTP)

A OTP is a password which is valid for only one login session or transaction. OTP usually avoids a number of shortcomings that are associated with traditional and static passwords. The most important shortcoming that is addressed by OTP is that, they are not vulnerable to replay attacks. This means that if a potential intruder manages to record an OTP that was already used to log into a service or to conduct a transaction; he or she will not be able to abuse it since it will be no longer valid. On the downside, OTPs are difficult for human beings to memorize. Therefore they require additional technology in order to work [13]. OTP generation algorithms typically make use of randomness. This is necessary because otherwise it would be easy to predict future OTPs from observing previous ones. Concrete OTP algorithms vary greatly in their details. Various approaches for the generation of OTPs are listed below.

- Based on time-synchronization between the authentication server and the client providing the password (OTPs are valid only for a short period of time)
- Using a mathematical algorithm to generate a new password based on the previous password (OTPs are, effectively a chain and must be used in a predefined order).
- Using a mathematical algorithm where the new password is based on a challenge (e.g., a random number chosen by the authentication server or transaction details) and/or a counter.

There are also different ways to make the user aware of the next OTP to use. Some systems use special electronic tokens that the user carries and that generate OTPs and show them using a small display. Other systems consist of software that runs on the user's mobile phone. Yet other systems generate OTPs on the server-side and send them to the user using an out-of-band channel such as SMS messaging. Finally, in some systems, OTPs are printed on paper that the user is required to carry with them [13]. Step (1) of OTP application in Location Hide Algorithm (LHA) in Section 4.2 points this method which connects OTP value to devices ID.

B. Scenario

Before Shella goes to bed, she inputs her tomorrow's plain in her device: 1) to wake up at 7 am, 2) to visit a cafe at 8 am near her house, 3) to meet her friend at 9 am, 4) to buy a Jacket in black in A shop, a brown boots in B shop, and 5) to visit C book store and check some books which have been published recently. [6][7] [Figure 1].

In the scenario, we have three things to consider as shown below. Case 1 and 2 show some context-aware services using location information, and Case 3 shows location privacy issues in these services.

Case 1: Shop A and Shop B are able to send new information to Shella as soon as she shows up near their stores. Furthermore, they can recommend some products to her based on her purchase history at their stores.

Case 2: Book store C also sends the new book list to her according to her book purchase history and her favorite genre.

Case 3: Location Privacy: Shella uses wireless network by connecting to wireless APs when she is in a cafe, on Outlet Street, and in bookstore. And this wireless connection enables her to get smart services from stores. However, attackers can get her locations and her movement route by analyzing data stored in APs.

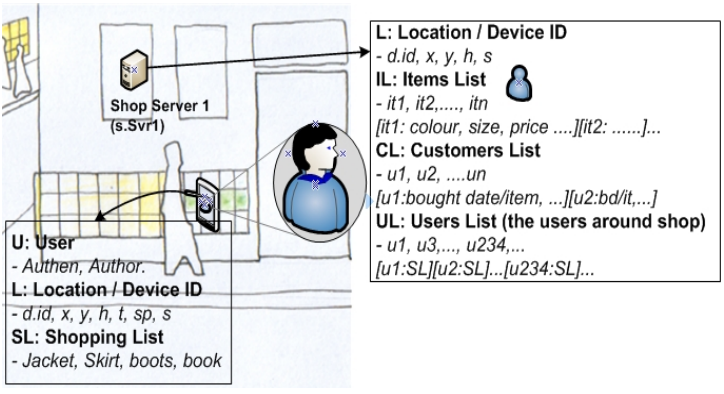

Fig. 1. Service Scenario. Shella tries to put that item (Jacket) on her AVATAR in her device without her directly wearing in that shop (one part-Book Store C and Shop B are in another area not in figure).

Fig.1 shows the service scenario in detail, and context information for the services. In that figure, Shella moves from one place to new place near AP *pn2,* and she gets services from stores near her location. Before she arrives at *pn2*, she got services from other *pn#*, and then servers and AP *pn#* keep her information in their log files or database. In this situation, if attackers access to the log files or database, then they can trace her location and s.

3.2 Context Classification

In context-aware systems, context determines service contents, and/or triggers event driven services. In Fig. 1 service scenario, context for the service consists of user location and other information such as user's purchase history and user's favorite genre. For the service, all context information is not only stored in her device, but also stored in all network devices around her [5][7]. At present location, she references Shop A, Shop B, Shop C, *pn2* and user contexts. As we see Fig. 2, *pn2* keeps all information about near around shops. Also, all shops have user's purchase history information (CLs in Shop A, B and C), and the purchase information is represented as `u1:bought date/item` in Shop A. This information is used to evaluate her purchase intention, and to recommend suitable goods to her on her visit [7]. She can also get some similar information by sharing or publishing her interests in goods of Shop A, B, and C. At this time, *pn2* are keeping information of Shop A, B and C. Also, it already has users' information that is in area (present information) or was in area (past information). Finally all shops can reference them.

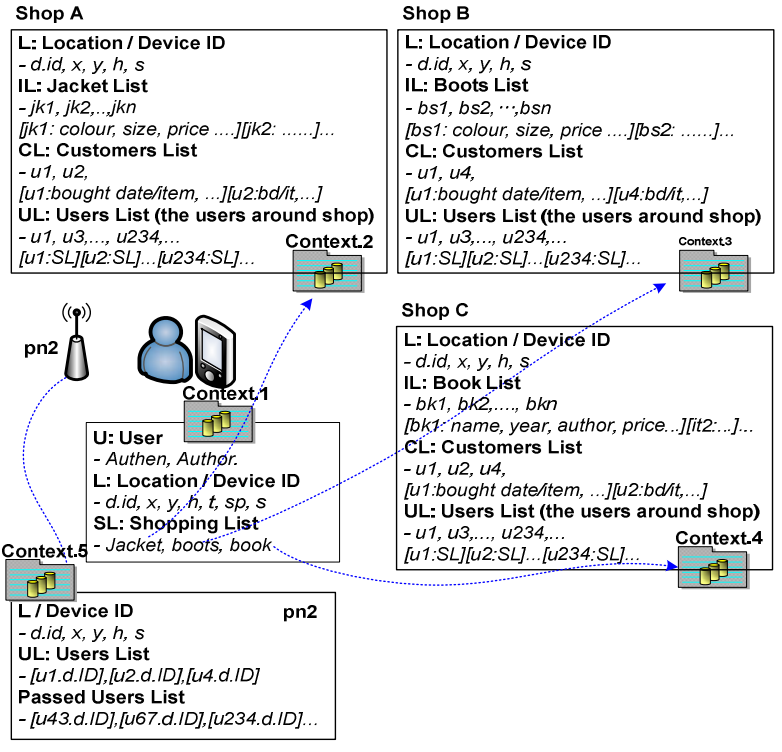

Fig. 2. Context Information for Service Scenario

4 Discussion

In section 4, we show seven UML sequence diagrams which define what actions are conducted while she moves around. Fig. 3 shows the sequence diagrams according to service scenario.

4.1 Sequence Diagrams

Fig. 3-(a) shows how her device is connected to the wireless network. Undoubtedly, it uses information in *pn2*. Fig. 3-(b) is the diagram that she adds her plan and her interesting goods to look for and to buy; usually she takes it before she goes out. Fig. 3-(c) shows her location.

Fig. 3-(d) illustrates how to evaluate Shella's preference based on her purchase history and it also shows how to recommend goods to her according to her preference information. Fig. 3-(e) is about to get detailed information about a specific goods. Fig. 3-(f) is the step to optimize the promotion for a specific user; it helps some users to organize SNS with them who are going to buy the same goods in future. At last, in Fig. 3-(g), she puts her buying goods on cyber character in her device, after that, she catches how good that before she buys.

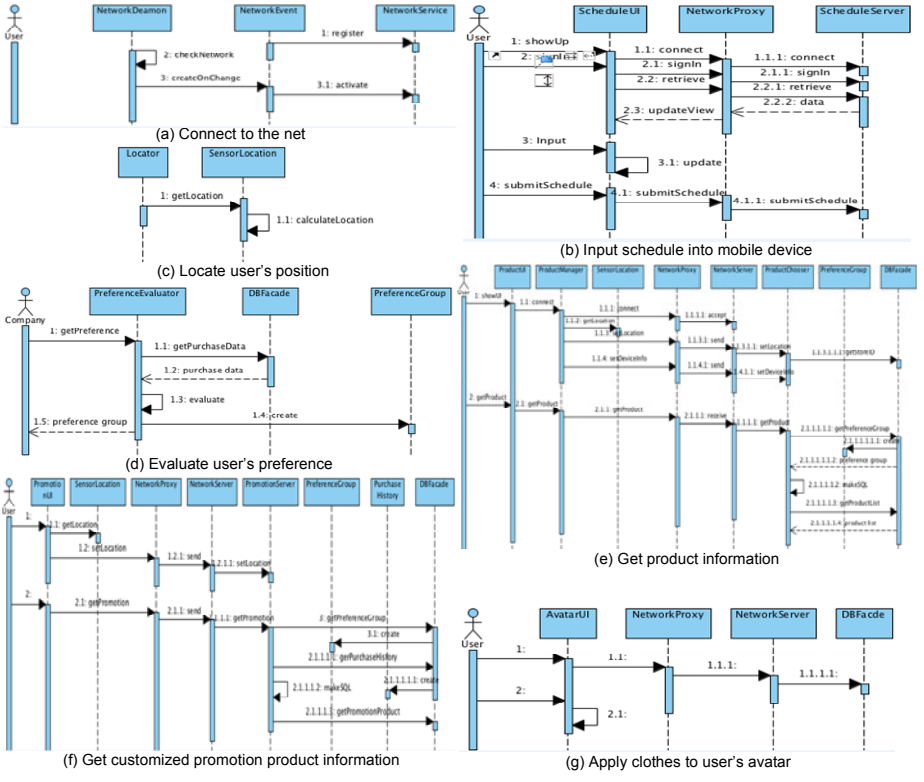

Fig. 3. Service Sequence Diagrams

4.2 Location Hide Algorithm (LHA)

Fig. 4 explains an algorithm named Location Hide Algorithm (LHA), which protects user's location and their routes with location hide flow [1].

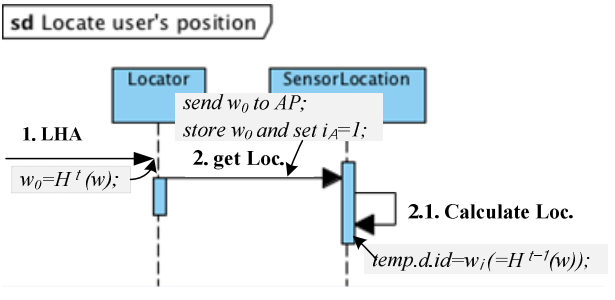

Fig. 4. Location Hide Flow

A user's location can be known by analyzing stored information in network devices or *pn#* which has provided services to users. In Fig. 4, the device information,

d.id, is registered to *pn2*, and attackers usually use *d.id* information to figure out from what *pn#* the user receives services. Finally they get privacy information, for example, what paths or what *a pn#* she has to get there etc. LHA connects user device information (*d.id*) and temporary registration information (w_i), and registers them. After then, *pn* sends all services to w_i, and it forwards them to *d.id* which is pending. Finally, it avoids the exposure of user's location by defining $d.id.w_i$. And although the user moves to other area, it is difficult to get the user's routes and present location because *d.id* is deleted in *pn#*.

The below algorithm defines LHA steps that we propose. In step (1) – step (2), it shows how it gets the value (w_i) by OTP, from step (3), it takes to connect both *d.id* and w_i which are gotten from step (1) through (2). That is, while the user moves, he/she is supposed to get services from *pn2*. At that time, *pn2* keeps $d.id.w_i$ information. Because of this process, no matter how attackers analyze log files in *pn2* and get w_i, they cannot know whose w_i it is. Finally, attackers cannot get his/her routes and where he/she is now.

```
// Location Hide Algorithm (LHA)
Initial Secret(W), HashFunction(H);
compute w0=H^t (w);                                          (1)
        send w0 to AP;
        store w0 and set iA=1;
        output temp.d.id=wi (=H^(t-1)(w)); //send A, i, wi
   process H(wi)=wi-1;
set iA <- iA+1;                                              (2)
put wi to d.id;                                              (3)
define d.id.wi; //d.id-real device ID, wi is temporal d.id
```

4.3 Recommending Items

Fig. 5 shows the simulation implementation result of context service model that we proposed using jContext[12][13], which is a java-based framework for context-aware systems. In the simulation, Shella moves through (i) Fashion Shop -> (ii) Café -> (iii) Book Store, also, before she leaves from home, she puts her interesting goods to buy today into her device.

Assume: *pn2* provides all services to (i) Fashion Shop, (ii) Café and (iii) Book Store.

In first place, (i) Fashion Shop, the shop server detects her from *pn2*, as soon as detecting her, it references all registering goods information that she had put before she leaves from her home. After that, servers in shops send the relevant notices such as recommendation of Blue Jacket and Leather Jacket. In next area, (ii) Café, that server will not recommend any coffee, because of her physical condition which had already added it by her. In last place area, (iii) Book Store, she gets recommendation book list: Black cat and White hand, because the bookstore evaluates her favorite genre as mystery based on her purchase history. Through those steps, she usually gets them step by step.

Fig. 5. Recommending Items near each area

5 Conclusions

In this paper, we studied context-aware services with hiding user location information. To avoid exposure of user devices information (*d.id*) and location (w_i), we used OPT, and then we tried to connect both *d.id* and w_i. By building and analyzing a simulation implementation, a user, Shella, can get her interesting notices according to her location which registered already before she goes out. Finally, she gets the relevant information from Fashion Store, Café and Book Store with *d.id (hidden)* and w_i *(open)* which set for service and for user location protection. In the future, we will add social network concept to this systems so that users who are in same area and have similar purpose communicate each other in ad hoc manner. Also, we will study some service scenarios illustrated in Fig. 3 in detail.

Acknowledgment. This work is partially supported under the support of the Portuguese Foundation for Science and Technology (FCT) in the aims of Ciência 2007 program for the hiring of Post-PhD researchers.

References

1. Krumm, J.: A survey of computational location privacy. Pervasive Ubiquitous Computing (13), 391–399 (2009)
2. Hulsebosch, R.J., Salden, A.H., Bargh, M.S., Ebben, P.W.G., Reitsma, J.: Context sensitive access control. In: Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies, SACMAT 2005, pp. 111–119 (2005)

3. Brown, G., Howe, T., Ihbe, M., Prakash, A., Borders, K.: Social Networks and Context-Aware Spam. In: CSCW 2008, San Diego, California, USA, November 8-12 (2008)
4. Dey, A.K.: Understanding and Using Context. Journal Personal and Ubiquitous Computing 5(1), 4–7 (2001)
5. Sadok, D.H., Souto, E., Feitosa, E., Kelner, J., Westberg, L.: RIP-A robust IP access architecture. Computer & Security, 1–22 (February 2009)
6. Yoneki, E.: Evolution of Ubiquitous computing with Sensor Networks in Urban Environments. In: Ubiquitous Computing Conference, Metapolis and Urban Life Workshop Proceedings, pp. 56–59 (September 2005)
7. Fetzer, A.: Recontextualizing context. In: Proceedings of Context Organiser Workshop, Manchester, UK, April 9-1 (1997)
8. Pieters, W.: Representing Humans in System Security Models: An Actor-Network Approach. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 2(1), 75–92 (2011)
9. Nobles, P., Ali, S., Chivers, H.: Improved Estimation of Trilateration Distances for Indoor Wireless, Intrusion Detection. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications 2(1), 93–102 (2011)
10. Bunt, H.: Context and aialogue control. In: Proceedings of CONTEX 1997 (1997)
11. Connolly, J.H.: Context in the study of human languages and computer programming languages: A comparison. In: Akman, V., Bouquet, P., Thomason, R.H., Young, R.A. (eds.) CONTEXT 2001. LNCS (LNAI), vol. 2116, p. 116. Springer, Heidelberg (2001)
12. Coutaz, J., Rey, G.: Recovering foundations for a theory of contextors. Presentation delivered at the 4th International Conference on Computer-Aided Design of User Interfaces, Valenciennes, France (May 2002)
13. One-Time Password (OTP), `http://en.wikipedia.org/wiki/One-time_password`
14. jContext, `http://sourceforge.net/projects/jmcontext/`
15. Choi, J.: jContext: A Toolkit for Context-aware Systems Using Java Reflection and Method Overloading. Journal of KIISE (April 2011) (submitted, in Korean)