PPNP: A Privacy Profile Negotiation Protocol for Services in Public Spaces

Shuhei Tamaru[1] **Jin Nakazawa**[2] **Kazunori Takashio**[1] **Hideyuki Tokuda**[3] [1]

[1] Graduate School of Media and Governance, Keio University

[2] Keio Research Institute at SFC

[3] Faculty of Environmental Information, Keio University

5322 Endo, Fujisawa, Kanagawa 252-8520,Japan
+81 466 47 0835
{syu-he-, jin, kaz, hxt}@ht.sfc.keio.ac.jp

ABSTRACT

In this paper, we propose Privacy Profile Negotiation Protocol (PPNP) that is a protocol for applying user's privacy profile to services in public spaces to protect users from theft and leakage of privacy profile. In public spaces, third parties can provide services. This may cause malicious services to exist, which handle privacy profiles illegally. Thus, unintended third parties know privacy profiles. Basically, it is difficult for users to distinguish whether a service is malicious or not. In addition, a method aiming at justifying a service is not realistic, since the justifier must also be justified. Therefore a new mechanism that does not quite trust services is needed. To cope with this problem, PPNP incorporates granularity for representing privacy profiles. Users can mosaic their privacy profile offered to services coarse by changing the granularity. In this paper, we also describe PEGASUS that is a middleware designed for services in public spaces, which implements PPNP. We also mention PPM, which is a sample application of PEGASUS with PPNP.

Keywords

privacy profile, malicious services in public spaces, privacy protection, Privacy Profile Negotiation Protocol (PPNP)

INTRODUCTION

In Ubiquitous Computing [3][4], users utilize services not only in their private spaces such as their rooms and their offices but also in public spaces like towns, train stations and schools. In public spaces, a number of unspecified users, varying in terms of sex, age, origin, educational background, and other attributes, utilize services. Therefore, services in public spaces should behave adaptive to each user considering his or her attributes.

In order to make the services behave adaptively, users are required to describe their attributes. The attributes will be stored in the user's mobile devices, which will contain higher-performance computation capability and wider-bandwidth network connectivity in the future. So users move with their own attributes kept in mobile devices. Such attributes may include mail address, phone-number, personal affiliation, and history of disease in addition to those previously listed. In this paper, we simply call the set of attributes described in users' mobile devices *"privacy profile."*

In public spaces, it is difficult for users to judge whether a service is malicious or not. The malicious service deals with users' privacy profile differently from the users' intention. For example, it may take a part of a user's privacy profile that he or she does not expect to be used for an adaptation-purpose. It may also transfer the privacy profile to other services without the user's permission. So a protocol for applying privacy profile to services should protect users' privacy profile from the malicious services in order to provide users with *"privacy-safety."*

To cope with the problem, we developed a privacy-safe protocol called "Privacy Profile Negotiation Protocol: PPNP." PPNP treats privacy profile with variable granularity, and enables users to decide disclosure level of their privacy profile. The granularity reflect that negotiated levels of the details.

The remainder of this paper is divided as follows. The next section discusses the services in public spaces and defines problems of privacy. Third section describes PPNP. Forth section explains the design of PEGASUS, which is an implementation of PPNP. Fifth section illustrates a scenario and implementation of PPNP and PEGASUS. Sixth section looks into related works. Final section concludes the paper and remarks future works.

APPLYING PRIVACY PROFILE IN PUBLIC SPACES

In this section, we describe services in public spaces, and define problems of privacy.

Applying privacy profile to services

We assume, in public spaces, services run on multimedia kiosks or Smart-Furniture [5], and assist user activities. For example, an advertisement viewer on a kiosk shows different advertisements for each user considering his or her privacy profile. It shows, in a department store, advertisements of men' or women's clothes for men or women, respectively. During the Christmas season, it shows toys if he or she has children. Similar services likely exist in other public spaces such as train stations, airports, and university campuses. These services, thus, benefit both users and service providers since the users can acquire various useful information without controlling the services each time, while the providers can deliver the information precisely to targeted users.

Figure 1. Smart-Furniture

We assume that users and service providers declare their intention to be benefited from through the privacy profile. Users set their privacy profile in their mobile devices like PDA, or mobile phones. This initial step occurs at the first time a user meets a service. This is like registration of rental video shops. When the users are about to use the service, the user's privacy profile is applied to the service. The service, then, behaves in a different way for each user according to the privacy profile.

Problems of privacy

In public spaces, there might be malicious services since third parties provide services. The malicious services handle privacy profile illegally, and consequently disclose privacy profile known to unintended third parties. Here we categorize behaviors of malicious services into two classes: *theft* and *leakage*.

Theft of privacy profile

This is the case where a malicious service takes a part of a user's privacy profile, which the user does not permit the service to use for adaptation, from his or her mobile terminal. For example, suppose a user expecting that "This service will use my birthday for adapting to my age." If the service takes his address and phone number instead, it is malicious.

Leakage of privacy profile

This is the case where a malicious service transfers privacy profile without authorization. For example, a malicious service operated by a certain shop leaks user's private information to other shops that do not have relationship with the user. Therefore, it is undesirable to offer his or her privacy profile to unspecified services.

PPNP: PRIVACY PROFILE NEGOTIATION PROTOCOL

To cope with these problems, we propose Privacy Profile Negotiation Protocol (PPNP). PPNP is a negotiation protocol for protecting users' privacy. Here, we assume that the services might be malicious.

Types and Granularity of privacy profile

PPNP assumes that privacy profile has granularity. A classification of privacy profile is shown below. Each of them has features and types, so appropriate divisions are needed.

Feature and type	Exapmles
Atomic	Name
Time	Birthday, Wedding day
Hierarchical strings	Address, Affilication
Hierarchical number	Phone number
Atomic limited	Sex, Type of blood

When users apply their privacy profile to services, users can change their granularity to be offered to services. It is a method we regard as protecting users' privacy. A user can offer abstract privacy profile to a service that the user does not trust well, while concrete privacy profile to a close-to-home services like those of a favorite shop, or a service whose service provider knows much of the privacy profile in the concrete privacy profile like a school the user attends. For example, time can be divided into six or more elements: years, months, days, hours, minutes, seconds, and so on. Similarly, an address can be divided into several levels of details such as country, city, town, street name, and so on. Even if the user's country is exposed, it is difficult to know user's real address. In other words, whether the theft and leakage of the privacy profile leads to probles of privacy depends on the granularity of profile.

On the other hand, services need to adapt for change of privacy profile. They need to prepare behaviors for diverse granularity of privacy profiles. For example, a service that changes its behavior according to user's addresses, needs to behave differently for a privacy profile exposing, "a user living in Japan", and another expressing "a user in Tokyo".

Data elements of PPNP

PPNP uses the following four data elements: *privacy profile*, *service rule*, *user rule*, and *control command*.

Privacy profile

Privacy profile contains all kinds of attributes of a user. It consists of address, birthday, phone-number, history of affiliations, and other important attributes. As mentioned

above, PPNP assumes that privacy profile has granularity. Each profile should be able to change its granularity. The granularity of hierarchical data and time are changed as mentioned above. However the granularity of atomic type attributes like name, sex, type of blood, and so on, is not changed in the same way. The granularity is either the only one granularity or nothing as data.

Service rule

A service rule is a formula or a set of formulas described by a service provider. When a service starts, the service sends a service rule to a user's host. Next, the user's host generates a control command from the user's privacy profile and the service rule, and sends it back to the services.

Service rules need to be defined considering a diversity of granularity, since users can control granularity for each portion of their privacy profile. A diversity of granularity indicates that diverse control commands exist for each privacy profile. For example, suppose two users, one permits a service to fully utilize his birthday profile, and the other only her birth century. In this case, a service that utilizes their ages should be able to accommodate these two different granularities.

User rule

A user rule decides the granularity of his or her privacy profile being disclosed. When a service rule arrives at a user's host, requested privacy profiles are listed. The granularity of requested privacy profile offered to a service rule is changed according to the user rule.

A user rule is a formula described by users. As mentioned above, users can configure it according to the confidence of services. For example, a user offers privacy profile in a large granularity to unknown services, while in a small granularity to well-known services. In other words, user rules should exist for each unit of privacy profile and each service. For example, a user rule for a user's birthday decides the granularity of its disclosure to an advertisement service, and another one in the same way.

Control command

A control command is a data element on which the behavior of a service is decided. The control command is only data element offered to services. And services behave according to the command. Therefore, the service provider should be able to decide the description. Furthermore, the service provider should not be able to derive back user's privacy profile from the service rule and the control command. Because if a malicious service provider can derive back concrete privacy profile from a control command and a service rule that he or she has described, problems of privacy may occur. Therefore, granularity of privacy profile stands up.

Behavior of PPNP

In this subsection, we describe the behavior of PPNP. Figure 2 indicates it by a simple time chart. The vertical axis indicates timeline. Arrows indicate some actions. Dotted arrows indicate the other pattern of sequence.

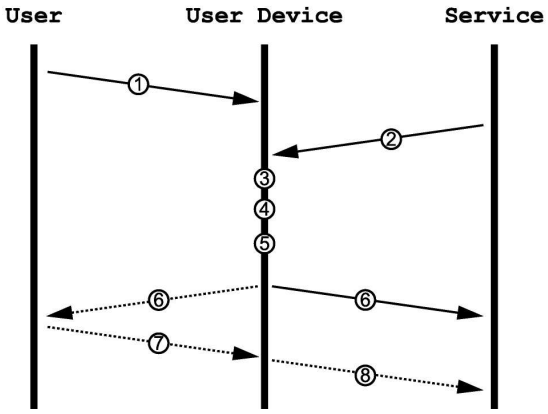

Figure 2. Behavior of PPNP

1. A user sets his or her privacy profile and user rules on a user device.
2. A service starts with some events or a request as triggers. For example, a service starts if a user pushes a start-button, or if the user enters a certain area. The host that runs the service immediately sends a service rule to the user's mobile device.
3. According to the service rule, requested privacy profiles are listed on the user's device.
4. According to a user rule, privacy profiles are changed its granularity and offered to the service rule.
5. From the service rule and the privacy profiles offered, a control command is generated.
6. A control command is returned.

6'. If PPNP is implemented for enabling users to confirm the privacy profile that is going to be used, the user is notified.

7. The user confirms.
8. Then, the control command is returned.

Basically, a control command indicates a part of privacy profile because it is generated from the service rule. For example, suppose a service provider has described a service rule whose relationship between privacy profile and control command is one-to-one, as following.

if (profile=A) then command=a.

else if (profile=B) then command=b.

This description means that in the case the control command is "a", the user's profile is "A", and the same goes for "b." In the case where the service is malicious,

problems of privacy occur, although in the case where the service is not malicious, do not.

Therefore, it should be guaranteed that a service rule cannot derive back the concrete privacy profile from itself. PPNP satisfies this since it introduces the notion of granularity for privacy profile.

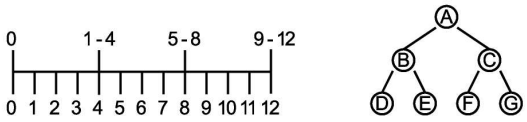

Figure 3. Notion of granularity

The left side of Figure 3 illustrates the case of continuous data, like some number of times or some anniversary such as birthday, wedding-day, and so on. If a user's privacy profile is "3", it is changeable to "1-4", or "0-12." Thus, a malicious service rule cannot pinpoint a concrete profile. For example, in the case where the user is "24 years old", the profile can be changed to "twenties."

The right side of Figure 3 illustrates the case of hierarchical data, like an address, and an affiliation. If a user's privacy profile is "A-B-D", it is changeable to "A-B", or "A." For example, in the case where the user's address is "XXX N. Mathews Ave. Urbana, IL 61801 USA", it can be changed to "Urbana, IL 61801 USA", "IL 61801 USA", or "USA."

Thus, a malicious service rule cannot construct one-to-one relationship between user's privacy profile and control commands. In this manner, PPNP enable both protection of privacy and application of privacy profile.

Representation of Service Rules

We have two ideas to implement PPNP. One is describing service rule as a symbolic description, and the other is exploiting mobile code technologies.

Symbolic description

In this method, a service rule is implemented as a symbolic description, which maps rules and control commands. For example, we can define:

Birth {1970:1979=A}{1980:1989=B}

This means that "In the case of '70s age, control command is A, and in the case of '80s age, control command is B."

An advantage of this method is that it is easier to check whether the service rule can derive back privacy profile than the latter method. A disadvantage of this method is that massive description of a service rule is needed and that the flexibility of the application is poor.

Mobile code

In this method, a service rule is implemented in an executable code, and transmitted to the user-side terminal. In this case, each service rule is defined as a conditional expression. For example, we can define:

```
birthday = getBirth(profile);
if(birthday.month == getCurrentMonth())
        SetControlCommand(birthday.month);
else if(birthday.getAge()>15 && <21)
        SetControlCommand("young");
```

This means that "In the case where today is the user's birthday month", control command is "for birthday month", and in the case where a user is 16-20, control command is "for young."

An advantage of this method is that it can be more powerful to apply privacy profile to services than the former way. A disadvantage of this method is that it can be less portable because of dependence on the program language. Moreover, in order to check whether the service rule can derive privacy profile or not, a library described in the specific language is needed. Therefore, it is limited for a programmer of services to make services with the specific language.

DESIGN

In this section, we describe the design of PEGASUS, which is an implementation of PPNP.

PEGASUS overview

Figure 4 illustrates an overview of PEGASUS.

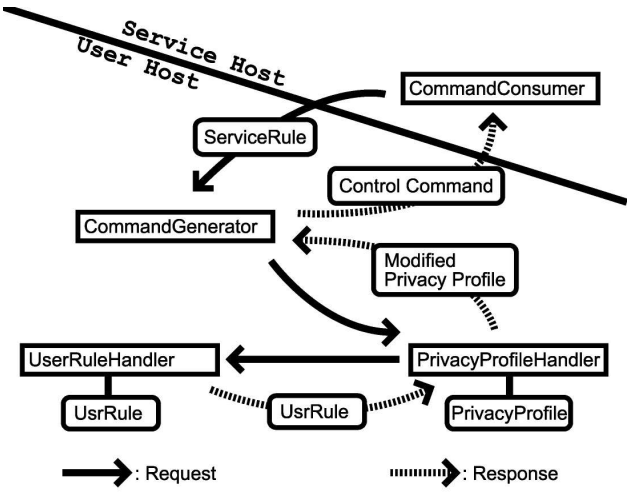

Figure 4. PEGASUS overview

PEGASUS runs on both the host that stores a user's privacy profile and the host that runs a service. We assume that the host where the user's privacy profile is stored is a mobile device that the user possesses, or a personal server that he or she administrates.

PPNP in PEGASUS

In PEGASUS, PPNP is implemented as following.

Privacy profile

A description of privacy profile should have a changeable granularity for safety of privacy. Moreover, the description

should have portability for adaptation to various services. So it should be described in a portable language.

Service rule and Control command

In PEGASUS, we have employed mobile code for designing a service rule. A service rule in mobile code should guarantee safety and portability. Safety means that the code must not touch and change resources not permitted, and must not destroy a system that it has been sent to. Portability means that the code should not only have a template for transparency but also be adaptive for diverse services.

A service rule consists of both conditional formulas for generating control commands and needed privacy profile.

User rule

The description of a user rule should correspond to a privacy profile. In other words, it exists for each privacy profile. The problem is that it is not scalable for a user rule to exist on each service. Therefore, some classification according to services' confidence level is needed.

Components

PEGASUS consists of four components: (1) *Command consumer*, (2) *Command generator*, (3) *Privacy profile handler*, and (4) *User rule handler*. Command consumer runs on a host where the service runs, and the other components run on a host that has the privacy profile.

Command consumer

Command consumer provides an application interface. The command consumer needs to run on a host that the service behaves. A service rule is sent via the command consumer. Services get commands via the component.

Command generator

Command generator runs on a host that keeps user's privacy profile, and offers run-time environment for service rules. Getting a service rule, the command generator invokes other components in order to generate control commands. Command generator sends a generated control command to the command consumer, running on a host that the service behaves.

Privacy profile handler

Privacy profile handler has a pointer for the privacy profile. When the privacy profile handler is called from the command generator, this component obtains the required privacy profile and invokes the user rule handler for obtaining user rules for changing the granularity of the privacy profile. Next, the component changes the granularity of the privacy profile and offers it to the command generator that is running the service rule.

Furthermore, this component offers a facility for editing the privacy profile. For example, an application programmer can construct a GUI through which end-users can edit their user rules easily.

User rule Handler

User rule handler has pointers for user rules. When the user rule handler is called from the privacy profile handler, this component obtains required user rules and passes it to the privacy profile handler.

Furthermore, this component offers a facility for editing user rules, similar to privacy profile handler.

APPLICATION EXAMPLE

In this section, we introduce our implementation of an application on Smart-Furniture. The service we introduce here is Personalized Public Message-Board (PPM). This is an electronic advertisement display that can change its contents of advertisement according to user's privacy profile. PPM can be installed in public spaces, like entrances of departments, train stations, campuses, and so on. Figure 5 depicts the screen dump.

In the current version, PPM changes its contents when a user enters the sensing area of RF-ID reader. And a host which keeps users' privacy profile is an IPAQ with familiar Linux v0.6.

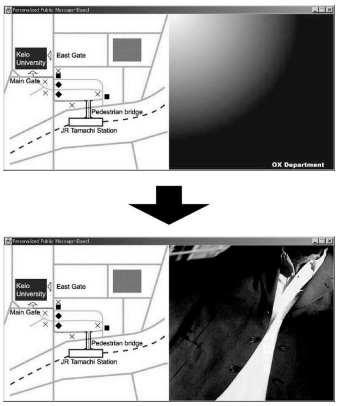

Figure 5. PPM

PPM is constructed with PEGASUS. We employed ContactXML [1] on a description of privacy profile, which is XML-based. PPM is a realization of a service applying users' privacy profile to protect privacy in public spaces.

RELATED WORK

Foregoing researches principally focus on location privacy. LOCATIONWARE [7] challenged the problem of privacy of location information in public spaces. Their approach is a method of negotiation between users and service providers on disclosure of users' location based on users' and services' policies. This approach realizes introduction of the system to accounting. However, their approach does not consider the confidence on the policy of services. Our approach can realize application of users' privacy profile to services to protect user's privacy even when the service is malicious. Cricket [6] enables users to control their location information that is offered to services. However, if a user does not offer any information, Cricket cannot provide anything. Our approach can realize a point of

compromise between the user and the service. Mist [2] realizes anonymousness, so the system cannot know user's location information, while the user can receive the full benefit of location information. However, their assumption does not consider existence of malicious third parties, and may not conform to public spaces. Our assumption is that malicious services exist. It is realistic especially for public spaces.

CONCLUSION

In this paper, we have proposed Privacy Profile Negotiation Protocol (PPNP) that is a protocol for protecting user's privacy profile from services in public spaces. PPNP enables a user to decide the granularity of disclosure of his/her privacy profile and the condition of disclosure. In consequence, the user's privacy is protected. In public spaces, there will be some malicious services. PPNP has a beneficial effect on services in public spaces where it is not able to identify whether a service can be trusted or not.

We have also proposed Personalized Public Message-Board (PPM) with PEGASUS that is a middleware implementing PPNP. However, PPM is very primitive as a service, since it only applies user's age and sex. PEGASUS should offer more expressive description of the privacy profile.

We are now extending this work to apply to diverse services, and to increase the portability of user rule description. We intend to provide a facility in PEGASUS for assisting users to separate a malicious service from regular services in public spaces.

ACKNOWLEDGMENTS

We thank all members of Tokuda Lab who wrote and provided helpful comments on previous versions of this document.

REFERENCES

1. ContactXML.org. http://www.contactxml.org/.
2. Jalal Al-Muhtadi, Roy Campbell, Apu Kapadia, Dennis Mickunas, and Seung Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environment", in *the International Conference of Distributed Computing Systems (ICDCS 2002)*, pp. 65-74, Vienna, Austria, July 3, 2002.
3. M. Weiser, "Some Computer Science Issues in Ubiquitous Computing", *Communications of the ACM, pp. 74-83(*1993).
4. M. Weiser, "The Computer for the Twenty-First Century", *Scientific American*, vol. 265, 1991, pp. 94-104.
5. Masaki Ito, Akiko Iwaya, Masato Saito, Kenichi Nakanishi, Kenta Matsumiya, Jin Nakazawa, Nobuhiko Nishio, Kazunori Takashio and Hideyuki Tokuda, "Smart Furniture: Improvising Ubiquitous Hot-spot Environment", in *Proceedings of IEEE 3rd International Workshop on Smart Appliances and Wearable Computing*, 2003, pp. 248-253.
6. N. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The Cricket Location-Support System", *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (ACM MOBICOM)*, Boston, MA, August 2000.
7. NEC corporation, LOCATIONWARE: policy-based location information platform for mobile phones. http://www.sw.nec.co.jp/nwk/products/application/locationware/pdf/locationware.pdf.

Shuhei Tamaru received Bachelors degree in Environmental Information from Keio University in 2003. He is currently a master's course student in graduate school of media and governance Keio University Japan. His research interests include ubiquitous computing systems, privacy protecting, and providing services in public spaces.

Jin Nakazawa received Ph.D. in Media and Governance from the Keio University in 2003. He is currently a researcher of Keio Research Institute at SFC. His research includes mobile code architecture, distributed component software architecture, and home area networks. He is a member of ACM, and IPSJ.

Kazunori Takashio received Ph.D in Computer Science from Keio University in 1995. He is Associate Professor in the Graduate School of Media and Governance at Keio University. His research interests include ubiquitous computing systems, mobile agent systems and programming frameworks for distributed real-time systems. He is a member of ACM, IEEE, IPSJ and JSSST.

Hideyuki Tokuda received Ph.D. in Computer Science from the University of Waterloo in 1983. He is currently a Professor in the Faculty of Environ mental Information, Keio University. His research interests include ubiquitous computing systems, distributed real-time systems, multimedia systems, mobile systems, networked appliances, and wearable systems. He is a member of ACM, IEEE and IPSJ.