

A Framework for the User-Oriented Personal Information Protection

Ken'ichi Takahashi

Institute of Systems & Information
Technologies/KYUSHU
2-1-22 Momochihama, Sawara-ku, Fukuoka,
814-0001, Japan

Kouichi Sakurai

Faculty of Information Science and Electrical
Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka,,
812-8581, Japan

Abstract - *Thanks to the spread of mobile technologies, we can access to the network anytime and from anywhere. In the near future, we will enable to realize the ubiquitous computing environment, in which user's mobile terminal assists in their activity. In the ubiquitous computing environment, user's mobile terminal manages personal information, which is used for negotiations with a service provider. However, various privacy problems, such as information leakage cases, have happened, so that we desire to protect our personal information.*

*We propose a framework for protecting personal information. A basic idea of the framework is to make use of personal information through a program which the owner of personal information knows the behaviour of. We call this program *trusted program*, because the owner of personal information can trust a way of personal information use. Then, a user offers his/her trusted program to a service provider and compels the service provider to make use of his/her personal information. In this paper, we introduce the framework under the assumption of using an anti-tampering device and propose the method for creating trusted-program.*

Keywords: personal information protection, privacy, user-oriented, ubiquitous computing environment

1 Introduction

Thanks to the spread of mobile technologies, we can access to the network anytime and from anywhere. In the near future, a lot of equipments, not only cellular phones and personal digital assistants, but also refrigerators and microwave ovens, etc, will be connected to the Internet. This will enable to realize the ubiquitous computing environment [10], in which equipments connected to the Internet assists in users' activities without special care on their part. In the ubiquitous computing environment, a user brings the mobile terminal, which can communicate with service provider around him/her. As the result of communications, the mobile terminal assists in his/her activities. Here, the mobile terminal will need user's personal information for doing effective assists of users'

activities. Therefore, the mobile terminal will manage user's personal information and uses them in communications with service providers. However, various information leakage cases, such as customer information leakage of Yahoo! BB and credit card information leakage of CardSystems Solutions, happen and we fear that our personal information is leaked. Therefore, we desire a method for protecting our personal information.

Here, we look at scenes when we buy products at an online store and we sign up for a profitable information distribution site and so on. Their sites require personal information such as name, address, telephone number, e-mail address and credit card number and so on. A user fills in his/her information and pushes the confirmation button. Then, their information is sent out to the site. Then, the protocol to send personal information and the way of information use are decided by that site¹. In other words, the user cannot decide the protocol and the way of information use, although information is his/her property. A user should be able to decide them, since personal information is each user's property. Moreover, a user decides the manner of his/her information use and should be able to prevent an illegal use of his/her personal information.

Therefore, we propose a framework for protecting personal information. A basic idea of the framework [8] is to make use of personal information through a program which the owner of personal information knows the behaviour of. We call this program *trusted program*, because the owner of personal information can trust a way of personal information use. Then, a user offers a trusted program to a service provider and compels the service provider to make use of his/her personal information. However, there are some challenges to actualize the framework. Therefore, we put the assumption of using an anti-tampering device. In this paper, we introduce the framework under the assumption of using an anti-

¹ Some pages allow users to select *http* or *https*. But it is just a right of selection, users can never select other protocol.

tampering device and propose the method for creating trusted-program.

The remainder of the paper is structured as follows. Section 2 describes related work. Section 3 shows the framework and proposes a method for creating a trusted program. And then section 4 concludes.

2 Related Works

Cryptographic algorithms, such as symmetric algorithms and public-key algorithms, and technologies based on them, such as digital signatures and Public Key Infrastructure (PKI), have been proposed [7]. These algorithms and technologies aim at the prevention of message interception or the identification of communication partners. Therefore we can ensure message confidentiality, integrity and availability against malicious exploitation by third parties. However, they are difficult to prevent the communication partner using information illegally.

At the bottom of homepages, we often find a link concerning privacy, indicated as *Privacy Policy* at Yahoo!, *Privacy* at IBM and so on. These pages show how a company treats users' personal information that the company collects. However, here are two problems. One is that users must read their page carefully. Most people do not read their pages, even when they provide their personal information. Another one is that users cannot confirm whether the company actually keeps the promises made on the privacy page or not. Consequently, we have no choice but to trust that the company keeps the promises made on the privacy page.

The Platform for Private Preferences (P3P) [5] is developed by the World Wide Web Consortium. P3P enables web sites to express their privacy policies in a standard format that can be interpreted automatically by user agents. Thus, a user agent can automate decision-making by the comparison between a privacy policy and user-specified privacy preferences. Therefore, a user does not need to read a privacy policy, because the user agent does it instead of the user. P3P, however, does not provide technical assurance that the site keeps the promise made on their privacy policies.

The Enterprise Privacy Authorization Language (EPAL) [2] is a formal language to specify fine-grained enterprise privacy policies. The EPAL policy defines formal privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations. Employees within the organization are compelled to keep EPAL policies. Thus EPAL prevents employees within the organization using information illegally. But EPAL does not aim to assure users of how

the organization protects users' personal information. Consequently, users cannot know whether the organization manages personal information securely or not.

There are various researches for providing a right of information access or service use based on trustworthiness [1][9][11]. These researches aim to build up trust relationships among users. But it is difficult to define a general method for building trust relationships among users, because trustworthiness depends on users' conditions and/or situations. Moreover, a user just only believes the communication partner not to use his/her personal information illegally. They do not provide any restriction of information use. Therefore, the partner can use information anyway, even if an illegal purpose.

Some researchers try to use information securely by a support of a mediator [4][6]. Here, we must prepare a mediator(s) which both sides of communications can trust. It is difficult in the ubiquitous computing environment, since there are a great many users and services. Otherwise, a computational load is centralized in the mediator.

3 A Framework for Personal Information Protection

We introduce a framework for protecting personal information by compelling a service provider to use user's personal information through a program which the user knows the behaviour of. The framework is based on two agent systems, named KODAMA [12] and VPC [3].

In the framework, both a user and a service provider are represented as an agent. An agent has a *public zone* and a *private zone*. The public zone is a freely accessible space to realize flexible service use. And we assume the public zone has an anti-tampering device in which a trusted program runs. The anti-tampering device has a *public key* and a *private key* of public-key cryptography, and a certificate for their key given by its manufacturer. The private zone manages and protects personal information. And there is a *security barrier* between the public zone and the private zone. The security barrier has a function for restricting an access to personal information, and a function for controlling communications of the program which accessed personal information.

3.1 Public Zone

In the ubiquitous computing environment, there are a great many services, which have a different method for its use. So, it is difficult to implement an agent which is able ab initio to use various services. Therefore, we define a *service program* and a *client program* as a pair. The service program defines a process for providing a service. The client program defines a process for using a service. A service provider discloses a *public policy* which consists of

their programs and some service attributes in the public zone.

When a user wishes to use a service, the user acquires a public policy for the service from the service provider and invokes the client program included in the public policy. Then the service is actualized through communications, guided by the client program, between the user and the service provider. In this way, a user can make use of various services without implementing explicit methods for the use of various services in advance (Figure 1).

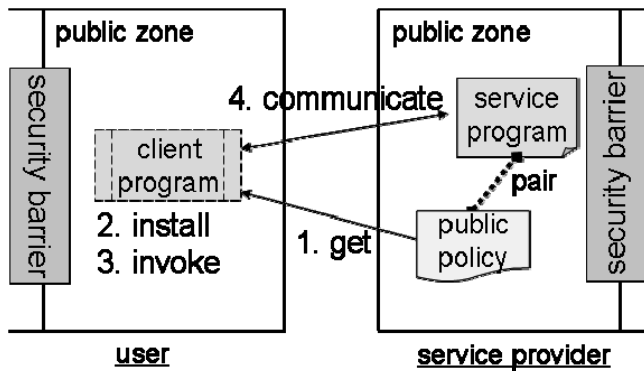


Figure 1. A Method for Flexible Service Use

In here, some services require user's personal information, so that the client program tries to access personal information. Then the security barrier checks whether its access allows or not. After it is accepted, the client program may send it to the service provider. Then, the security barrier creates a trusted program which guides a process of information use. After that, the user gets a public key and its certificate of an anti-tampering device of the service provider. The user verifies the certificate and confirms the public key is owned by the anti-tampering device made by a legitimate manufacturer. The user encrypts the trusted program and personal information using the public key and sends it to the service provider. The anti-tampering device of the service provider decrypts it using the private key and invokes the trusted program. Then the service provider uses user's personal information in the anti-tampering device through the trusted program. In this manner, a user protects his/her personal information by his/her trusted program (Figure 2).

3.2 Private Zone

The private zone manages personal information. Personal information has a *privacy policy* which consists of some attributes for protecting personal information. Privacy policies are registered with the security barrier.

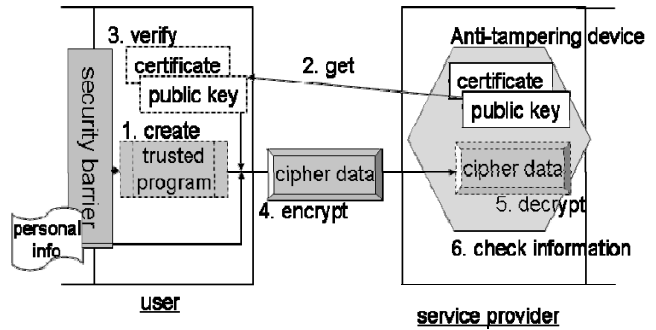


Figure 2. A Method for sending a Trusted Program

Behaviour of the private zone is illustrated in Figure 3. When an access to personal information from a client program happens, the security barrier compares the public policy of the client program and the privacy policy of the personal information, and then decides whether the access is accepted or not. If the access is accepted, the security barrier registers the client program into an *access-table* and returns its value; if it is refused, an *IllegalAccessException* happens. After that, the security barrier monitors communications of the client program registered into the access-table. When the client program tries to communicate with an agent, the security barrier creates a trusted program from attributes of the public policy and privacy policy. After that, the user sends the trusted program to the communication partner as mentioned at Section 3.1.

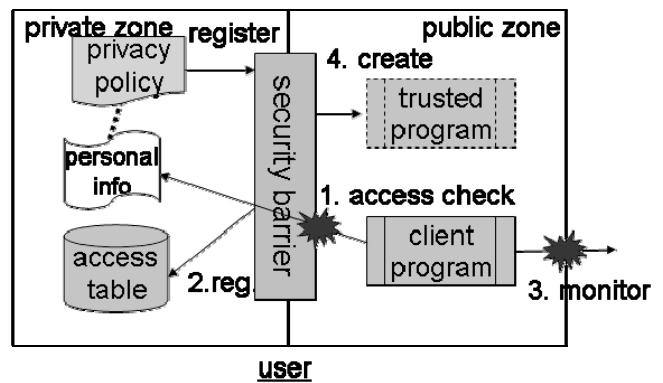


Figure 3. Behaviour of the Private Zone

3.3 The Public Policy and Privacy Policy for Creating a Trusted Program

A user protects his/her personal information by compelling a service provider to use personal information through his/her trusted program. In other words, the service provider confirms a qualification of the user for a service use through the trusted program. For example, the service provider tries to do a verification of a password, a

```

<public policy> ::= <access-infos><client-program><service-program>
<access-infos> ::= *<access-info>
<access-info>  ::= <info><purpose><partner><use>
<partner>     ::= "owner" | *<name> | "no"
<use>         ::= <where><operation><reference>
<where>       ::= "client" | "service"
<operation>   ::= "equal" | "greaterthan" | "lessthan" | "concatinate" | ...
<reference>   ::= places where uses <info> on the programs

```

Figure 4. The Public Policy

```

<privacy policy> ::= <info><allowable-purpose><allowable-operations>
                  <partner><protocol>
<allowable-operations> ::= *<allowable-operation>
<allowable-operation> ::= <operation><conversion-rule>
<operation>           ::= "equal" | "greaterthan" | "lessthan" | "concatinate" | ...
<conversion-rule>    ::= conversion rule of <operation> for creating a trusted
program
<partner>            ::= "owner" | *<name> | "no"
<protocol>           ::= <protocol-name><program>

```

Figure5. The Privacy Policy

confirmation of an age limit and so on through the trusted program. Therefore, the trusted program must be a program which enables to confirm that the user is qualified for the service use. So, we define the public policy and the privacy policy shown in Figure 4 and Figure 5.

The public policy consists of access-info(s), client-program and service-program. Client-program is a reference to a client program; service-program is a reference to a service program. Access-info shows that a service provider requires info; info is used for purpose; then a user must communicate with partner for achievement of purpose; partner uses info for use. And info is used for operation at places of reference on where (client or service program).

The privacy policy shows to allow info to be used for allowable-purpose; to be used only by computations shown in allowable-operations; to be sent only to partner; then info is sent out by protocol. And allowable-operation shows when a program uses info, the program has to be converted according to conversion-rule.

Then a trusted program is created by a security barrier. First, a security barrier looks over operation of a public policy and allowable-operation of a privacy policy. Then if allowable-operation of the privacy policy includes operation of the public policy, the security barrier extracts parts which are indicated by reference of the public policy. After that, the security barrier converts their parts

according to conversion-rule. The converted program is a trusted program, because the user knows a way of his/her personal information use. The trusted program prevents a service provider from illegal use of his/her personal information.

3.4 An Example of Trusted Program Creation

We show an example of creation of a trusted program. In this example, a service provider provides a service after confirmation of user's ID and password pair. Then a user has to send his ID and password pair; the service provider receives and tries to confirm them. Therefore a service program and client program is defined as the left side of Figure 8. Here, a service provider tries to confirm an ID and password pair by an equal operation at line s4 of the service program. Arguments of equal operation are a password and a value from database of the service provider. The password is received at line s1 and the value appears at line s3. Therefore a part of the password of the public policy is defined as Figure 6. We omit a part for the ID.

A user has an ID and a password. Especially, the password is so important personal information. So the user hopes the password to be used only by a hashed form. Moreover, ID and password are usually used only for evaluating equivalence of values. Therefore a privacy policy for the password is defined as Figure 7. We omit a privacy policy for the ID.

```

<info>="password",      <purpose>="login",      <partner>="owner",      <where>="service",
<operation>="equal",
<reference>={
  method : eq(a, c) at line s4
  where a is "password" received at line s1 from line c2 of client,
        c is "localvalue" at line s3
}

```

Figure 6. An Example of a Public Policy for the Password Confirmation

```

<info>="password", <allowable-purpose>="login", <operation>="equal",
<partner>="owner", <protocol-name>="https", <program>=http program,
<conversion-rule>={
  convert value : password -> hash-pass by hash(password) at client
  convert method: method(password, b) -> method(hash-pass, hash-b)
  where hash-b is hash(b)
}

```

Figure 7. An Example of a Privacy Policy for the Password

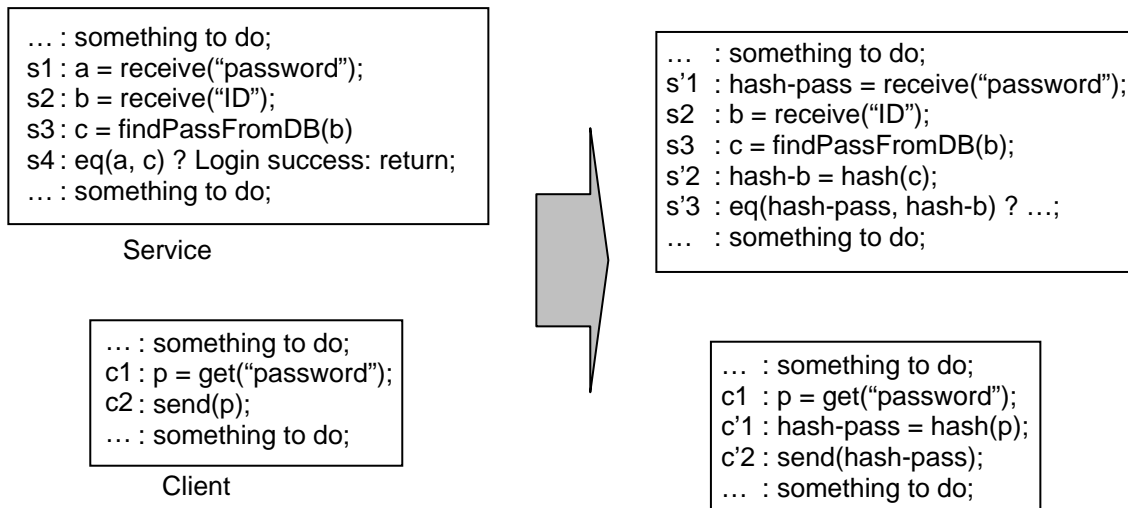


Figure 8. An Example of Creation of a Trusted Program

Then a trusted program is produced as follows. First conversion-rule of privacy policy converts password p of line $c2$ in the client program to $hash-pass$ by function of $hash(password)$ (see line $c'1$ and $c'2$ in Figure 8); value a of the service program is also converted to $hash-pass$ (see line $s'1$). Second conversion-rule converts $eq(a, c)$ of line $s4$ to $eq(hash-pass, hash-b)$, where value c of line $s3$ is converted to $hash-b$ (see line $s'3$ and $s'2$). As the result, a trusted program of the right side in figure 8 is produced. Therefore the trusted program is a program which the user trusts a way for dealing with a password.

3.5. The Overall Process for Personal Information Protection

We summarize the overall process for the personal information protection. When an access to personal information from a client program happens, a user behaves as follows:

1. A security barrier finds a privacy policy of `info` which a client program tries to access.

2. The security barrier confirms whether **purpose** of the public policy is accepted or not by **allowable-purpose** of the privacy policy.
3. If it is accepted, the security barrier registers the client program into an **access-table** and returns its value; if it is refused, an `IllegalAccessException` happens.
4. The security barrier returns a value of `info` to the client program.
5. When the client program tries to communicate to other agent, the security barrier confirms that the agent is specified into both **partner** of the public policy and **allowable-partner** of the privacy policy.
6. If it is specified into both of them, the security barrier confirms whether **allowable-operation** of the privacy policy includes **operation** of the public policy or not; if it is not specified, an `IllegalCommunication-Exception` happens.
7. The security barrier creates a trusted program from the client program and the service program, by converting places of their program specified in **reference** of the public policy according to **conversion-rule** of the privacy policy.
8. The security barrier obtains a public key of the anti-tampering device which the service provider has, and encrypts the trusted program and personal information with the public key. Then, the security barrier sends encrypted one to the service provider.
9. The service provider decrypts encrypted one with a private key in the anti-tampering device.
10. The service provider confirms user's personal information through the trusted program in the anti-tampering device.

A security barrier confirms whether **purpose** of a personal information use is accepted or not at step 2. Then the security barrier creates a trusted program at step 7 and sends it to a service provider at step 8. Then the service provider confirms user's personal information indirectly through the trusted program in the anti-tampering device. In this manner, a user prevents a service provider from illegal use of his/her personal information.

4 Conclusions

We introduced the framework for protecting personal information under the assumption of using an anti-tampering device. In the framework, a user protects his/her personal information by compelling a service provider to use personal information through a trusted program. In this

manner, a user prevents a service provider from illegal personal information use. To actualize the framework, we defined the public policy and the privacy policy. The public policy specifies what information is required as conditions of a service use. The privacy policy specifies qualifications of personal information use. And we showed the method for creating trusted program from a public policy and privacy policy. A future work is to provide the detailed definition of the public policy and the privacy policies toward practical applications.

Acknowledgment

This research was supported by Strategic International Cooperative Program, Japan Science and Technology Agency (JST), and by Strategic Information and Communications R&D Promotion Programme under grant 052310008.

References

- [1] C. English, P. Nixon, S. Terzis, A. McGettrick, H. Lowe. Dynamic Trust Models for Ubiquitous Computing Environments. *UBICOMP 2002*, 2002.
- [2] The EPAL 1.1. <http://www.zurich.ibm.com/security/enterpriseprivacy/epal/>.
- [3] T. Iwao, Y. Wada, M. Okada, and M. Amamiya. A Framework for the Exchange and Installation of Protocols in a Multi-Agent System. *Proc. of Cooperative Information Agents 2001*, LNCS 2182, pp. 211-222, 2001.
- [4] Y. Karabulut. Towards a Next-Generation Trust Management Infrastructure for Open Computing Systems. *SPPC04*, 2004.
- [5] P3P project. <http://www.w3.org/P3P>.
- [6] C. Pearce, P. Bertok, R. Schyndel. Protecting Consumer Data in Composite Web Services. *IFIP/SEC 2005*, 2005.
- [7] D.R. Stinson, editor. *Cryptography: Theory and Practice*. Crc Pr I Llc, 1995.
- [8] K. Takahashi, K. Sakurai, M. Amamiya. A Framework for Protecting Private Information through User-Trusted-Program and its Realizability, *UISW 2005*, LNCS 3823, pp. 433-442, 2005.
- [9] G. Theodorakopoulos, J. Baras. Trust Evaluation in Ad-Hoc Networks. *WiSe04*, pp. 1-10, 2004.
- [10] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, pp. 94-104, 1991.

[11] D. Xiu, Z. Liu. A Dynamic Trust Model for Pervasive Computing Environments. *FTDCS 2004*, pp. 80-85, 2004.

[12] G. Zhong, S. Amamiya, K. Takahashi, T. Mine, and M. Amamiya. The Design and Implementation of KODAMA System. *IEICE Transactions INF. & SYST.*, Vol. E85-D, No. 4, pp. 637-646, 2002.