PPM: Privacy Policy Manager for Personalized Services

Shinsaku Kiyomoto, Toru Nakamura, Haruo Takasaki, Ryu Watanabe, Yutaka Miyake

PPM: Privacy Policy Manager for Personalized Services

Shinsaku Kiyomoto[1], Toru Nakamura[1], Haruo Takasaki[2], Ryu Watanabe[1], and Yutaka Miyake[1]

[1] KDDI R & D Laboratories Inc.
2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502, Japan
kiyomoto@kddilabs.jp
[2] KDDI Research Institute Inc.
3-10-10 Iidabashi, Chiyoda-ku, Tokyo, 102-8460, Japan

Abstract. In this paper, we introduce a new architecture for personalized services. The architecture separates access control using a user own privacy policy from data storage for private information, and it supports privacy policy management by users. We design a core module, the Privacy Policy Manager (PPM). The module includes several functionalities: ID management, privacy policy management, control of information flows, and recording the flows.

1 Introduction

Personalized services have been successfully implemented in a variety of services such as targeted advertisements, personalized searches, and location-based services. Privacy breach has been a major concern for users of personalized services, not only online web services but also offline real services. O2O (Online to Offline) is a new direction for commercial services; however, privacy concerns have become serious due to the expansion of service collaborations. Users have been very concerned when diverted to services they were unaware of having any relationship with. In fact, some research results [26, 34] have suggested that Internet ads personalized with private data leak users' private information. On the other hand, it has been suggested that the creation of privacy awareness can assist users in dealing with context-aware services without harming their privacy unintentionally [15].

Another issue is the burden of checking on and maintaining privacy policies[33]. Users must check the privacy policies of a service that is presented by a service provider before using the service. Each service provider prepares a privacy policy for each service, so users must often check on many privacy policies. Furthermore, it is troublesome that users cannot determine or customize the privacy policies for themselves. If a user does not agree with the privacy policy of a service, the user cannot use the service.

Solove suggested that the privacy self-management model cannot achieve the goals demanded of it, and it has been pushed beyond its limits, while privacy

law has been relying too heavily upon the privacy self-management model [43]. In his paper, issues involved in giving consent to a privacy policy are clarified as: (1) developing a coherent approach to consent, one that accounts for social science's discoveries about how humans make decisions about personal data, and (2) developing more substantive privacy rules. An experimental result [1] by Acquisti and Grossklags shows a lack of knowledge about technological and legal forms of privacy protection when confirming privacy policy. Their observations suggest that several difficulties obstruct even concerned and motivated individuals in attempts to protect their own private information. One article [41] also suggested that users were not familiar with technical and legal terms related to privacy. Moreover, it was suggested that users' knowledge about privacy threats and technologies that help to protect their privacy is quite inadequate [30].

The Platform for Privacy Preferences Project (P3P) [44, 10] enables websites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. The project provides user agent modules that allow users to be informed of site practices and to automate decision-making based on these practices when appropriate. However, in practice, it is not used by online and offline services [40] due to complex policy definitions, even though some browsers have a module for privacy matching. Furthermore, it is only considered to implement the module on web browsers.

In this paper, we consider an architecture for personalized services, and solutions to privacy problems related to personalized services. The architecture separates data storage from access control based on a privacy policy, and it supports privacy policy management by users. We design a core module named Privacy Policy Manager (PPM) that provides two functionalities: ID management and privacy policy management.

2 Towards Privacy-Preserving Personalized Services

In this section, we introduce the background of our study and clarify issues that arise in designing the architecture.

2.1 Personal Data Service

A personal data vault has been presented as support for a user-transparent architecture that can control information flow [19]. It is a secure container to which only the individual has complete access. It decouples the capture and archiving of personal data streams from the function for sharing that information. The personal data vault would then facilitate the selective sharing of subsets of the information with various services. There are some platforms that manage a personal data vault. An individual can execute functions in the personal data vault: controlled push and informed pull. Each platform is managed by a company, so individuals must trust the service provider of the platform. To solve this problem, the concept of Personal Data Service (PDS) has been presented, and some

research projects have provided tools for realizing individual-based management of private information.

The PDS is a platform that allows users to control their own information by themselves. It is used for sharing personal data with friends and organizations that are trusted. The PDS holds an individual's sensitive data such as address, credit card, and employment and gives the user access control functionality. The concept of the PDS is an individual-centric model, meaning that centralized access control by each individual should be provided on their own terminal. Both an access control mechanism and data stage for the sensitive data are implemented in a program (such as a web browser) on the terminal. By using the PDS, users are allowed to securely manage their own information and control data flows of the information. Higgins [21] is a browser extension including modules for PDS, and it supports PDS for browser interactions and web client interactions. The project Danube [13] is another instance of PDS for web services. The VRM project [42] is a research project that aims to provide a platform and tools for realizing a personal data service. The project defines five principles for customers who use privacy preserving services.

- *Customers must enter relationships with vendors as independent actors.*
- *Customers must be the points of integration for their own data.*
- *Customers must have control of data they generate and gather. This means they must be able to share data selectively and voluntarily.*
- *Customers must be able to assert their own terms of engagement.*
- *Customers must be free to express their demands and intentions outside of any one company's control.*

On the other hand, there is a problem in that an individual must manage all functionalities for protecting and controlling his/her private information. Thus, a more user-friendly architecture is required. We will formalize issues for personalized services based on the above principles in the next subsection.

2.2 Issues for Personalized Services

There are some issues in existing services, handling of an individual's private information, when seen as a personal data service. The PDS solves some problems outlined in this subsection, but some issues remain for constructing user-friendly architecture. We should clarify the issues before designing an architecture for personalized services. Four issues are summarized the following:

- *Complexity.* Current service providers issue their own privacy policies for each service. Users must examine and accept a huge amount of information in multiple policies before even beginning to use the services.
- *Flexibility.* Privacy policies are determined at the initiative of service providers. Some conditions of privacy policies (including *opt-out*) may be selected, but there is no guarantee that the conditions fit the user's privacy needs.

- *Availability.* Distribution of private information is restricted. Privacy related information is useful for user-centric (personalized) services such as recommendation services and support services. However, the service provider has appropriate methods for information distribution without privacy breach. Each service provider presents its own privacy policy, and it only covers the service from the service provider. Users hope to apply a common privacy policy for all services.
- *Assurance.* Users are necessarily concerned about the management of private information by the service provider. How to ensure the integrity of operations and how to improve the credibility of service providers are important issues as services using private information expand.

In this paper, we present an architecture that deals with the above issues.

3 Architecture for Personalized Services

In this section, we introduce an architecture for personalized services under a new personal data service concept. To deal with the issues listed in the previous section, we design an architecture that supports users in their effort to enforce their common privacy policies and that reduces the complexity of operations on the user side. The main features of the architecture are as follows;

- **Separation of Access Control and Policy Management.** We separate the functionality of the personal data service into two parts: data storage and access control. A trusted entity manages the access control portion to support individuals in configuring appropriate privacy policies and controlling information flows based on those privacy policies. The construction of data storage is beyond the scope of this paper; it is assumed that this is managed by each individual or distributed into some domains. Privacy policies are managed in the trusted entity in order to apply common policies to several services.
- **Support for Policy Management.** The architecture provides a mechanism that supports management of user privacy policies. The mechanism helps to create a common privacy policy of each user and optimize it based on user suggestions.
- **Interoperable Architecture.** The architecture provides a function for ID federation, and users delegate ID management for accessing several service providers to the architecture side in conjunction with the common privacy policy management.
- **Log Management.** The architecture has a proxy between users and service providers. All communication is recorded into a trusted area of the architecture. Thus, users can verify flows of private information.

Figure 1 shows the architecture. The main component of the architecture is a Privacy Policy Manager (PPM). The PPM manages an individual's privacy policies and controls flows of private information according to those privacy policies.

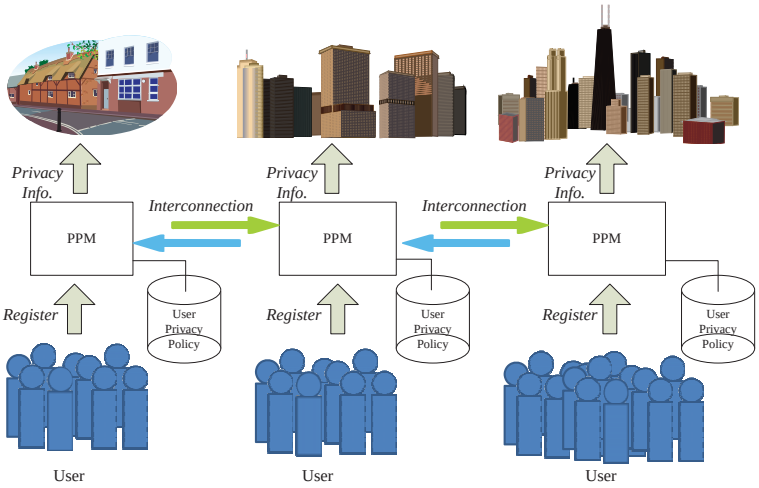

Fig. 1. Architecture for Privacy Policy Management

The PPM is built on a trusted entity in a domain, and each separate domain has at least one PPM. Individuals register their privacy policy with a PPM located in a domain to which the individual belongs, and configure the actions to be taken when a service provider requests private information whose delivery violates the privacy policies. For example, the PPM asks an individual whether the private information should be sent, when the act of sending the information is against the privacy policy of the individual. Inter-communication between PPMs is considered in the architecture. If an individual moves to another area, it is expected that the individual will access a PPM in the other area. In this situation, the new PPM requests the PPM that has the individual's privacy policy to transfer the privacy policy or a notice of a judgment on whether private information can be sent to a service provider.

Concept of Opt-In Domain. The "Opt-In Domain" is a concept for a comprehensive agreement on private information usage. Individuals generally have been concerned about privacy breaches in many situations, because they think that their private information may be used by unknown services or transferred to other service providers. On the other hand, availability is a problem on the service provider's side, as discussed in the previous section. The "Opt-In Domain" concept allows the use of private information not only by a service provider but also by other service providers who are located in a certain area (same domain), such as a local area, shopping mall, amusement park, small town, or university. The use of private information is restricted within a boundary defined in physical or virtual space[3]. In this concept, individuals define a privacy policy for a

[3] Note that a physical boundary is more acceptable to individuals, because it is an intuitive boundary, and individuals feel more confidence in this. Individual's acceptance of boundaries will be analyzed by an experiment in our future work.

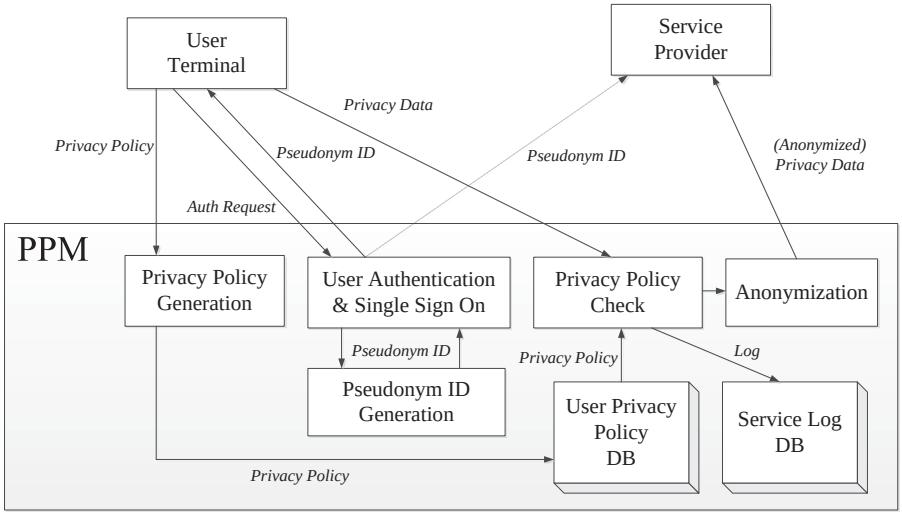

Fig. 2. Privacy Policy Manager

certain domain and give service providers in the domain-permission to access the individual's private information that is gathered in the domain. Our framework fits this concept because the PPM manages a comprehensive agreement on behalf of service providers.

4 Privacy Policy Manager

The Privacy Policy Manager (PPM) is the core model for the architecture. Individuals are users of a PPM on their domain, and an individual's privacy policy is managed in the database of the PPM, and information flow is controlled based on the privacy policies. The main role of the PPM is ID management, including user authentication and privacy policy management. Figure 2 shows an overview of the PPM. The PPM is similar to a proxy service including an access control mechanism, and has the following functions;

- User authentication and ID Federation.
  For user convenience, a single-sign-on scheme should be used. The PPM generates a pseudonym ID for each service provider and registers the pseudonym IDs to service providers. Once a user logs into the PPM, the user uses services provided by the service providers without additional login processes. The PPM automatically translates the user's original ID into the pseudonym ID and notifies the login condition to a service provider, when the user uses the service. The detailed mechanism of ID generation is explained in 4.2.
- Creation and Update of User Privacy Policy.
  The PPM should provide a user-friendly GUI for creating a user's privacy

policy. Furthermore, registered privacy policies should frequently be updated based on records of service use. We define two functions f_m and f_w for privacy policy management in later subsections.

- Privacy Policy Checking.
  When a service provider requests that a user send private information, the PPM should check the user's privacy policy and decide whether to send the private information.
- Storing Records of Service Use.
  Visualization of flows transferring private information is an important role of the PPM. All service access goes through the PPM. The PPM should hold all logs of the information flow, and provide them to users. We consider a concept called *user consent log search*, which is explained in 4.5
- Communication with other PPMs.
  To support a roaming user who belongs to another PPM, the PPM should have a communication function to ask for a privacy policy or a judgment about privacy control. The protocols are summarized in 4.6.
- Anonymization and Obfuscation (Optional).
  It is assumed that private information cannot be sent to a service provider in the original form, but that it is possible to send it after anonymization or obfuscation. For example, a user may allow the sending of approximate location information instead of precise location information such as GPS data. Thus, the PPM should have a function to modify private information in order to satisfy the privacy policy of a user. We can use existing techniques referred in section 5 for anonymization and obfuscation.

4.1 Procedure

The PPM has the role of a proxy that mediates communication between a user and a service provider. A pseudonym ID is also provided by the PPM in order to hide the user's identity and avoid a privacy breach that would make the user's actions traceable across several services. The procedure in a sample case of service use is as follows;

1. A user registers his/her privacy policy with the PPM before using services.
2. When a user registers with a service provider, the user first accesses the PPM and requests a pseudonym ID with an identification of the service provider. The PPM generates the pseudonym ID and sends it to the user.
3. When the user uses the service of the service provider, the user first logs in with the PPM. The user accesses to the service provider using the pseudonym ID. The service provider obtains authentication status from the PPM, then provides the service to the user.
4. During service provision, the service provider requests that the user send private information to the service provider via the PPM. The user sends private information to the service provider via the PPM. The PPM checks the privacy policy of the user and transfers the private information if allowed by the privacy policy.

5. The PPM stores the logs of the transfer of the private information.
6. The PPM updates the privacy policy of the user, if needed.

An offline batch operation to update of a privacy policy may be allowed, when updating would otherwise impose an excessive burden on the PPM.

4.2 Pseudonym ID Generation and ID Management

When the same IDs are used to identify users for all services, there is a privacy breach in that user activities on each service can be linked by their IDs. One simple solution to this problem is to use different IDs for each service. Users' activities then cannot be linked by their IDs even if service providers collude with each other. However, using different IDs on each service means that the PPM manages many IDs. We use a cryptographic technique for user ID generation on the PPM in order to reduce the cost of ID management by the PPM. The IDs that are used by service providers for user identification are generated from the user login ID plus a user secret like his/her password to the PPM, using a cryptographic technique.

A "pseudonym ID (pID)" is generated and used for each service in the architecture. The relationship between the newly generated ID and the user login ID (uID) is hidden in the pID itself. The PPM, which generated the ID, retains a master key K_m and does not have to maintain the relationship itself. An encryption key K_s is generated as $K_s = H(K_m||user\ secret)$, where $H(*)$ is a cryptographic hash function like SHA-256 and *user secret* is an input by the user during the user authentication process. The symbol $||$ denotes concatenation of data. Note that the PPM does not hold K_s itself as a security requirement. The ID generation scheme is shown below;

$$pID = E_{K_s}(uID||S_{info})$$

where S_{info} represents any bit string that is different for each service provider, such as the name of the service provider, and $E_{K_s}(*)$ is a symmetric key encryption algorithm with a secret key K_s. We use AES-256 (Advanced Encryption Standard with a 256-bit key) as the encryption algorithm for ID generation. The PPM dynamically generates the ID for each service provider and sends it to the user. We assume an *offline attacker* who can access local files of the PPM but cannot obtain any information from code that is executing on the PPM or a physical memory of an environment running on the PPM. That means that we assume an attack by a *curious* operator of the PPM. The PPM execution is protected by using software tamper-resistant techniques and memory-protection techniques; however, the *curious* operator (*offline attacker*) still has a chance to examine local files. The pseudonym ID is generated from a *user secret* and the master key K_m securely embedded in the PPM. Thus, the *offline attacker* cannot generate the pseudonym ID, and tracing the actions of a particular user is impossible.

4.3 Privacy Policy Creation and Modification

The PPM has the function of creating and updating user privacy policies. It is an essential task for a user to configure a precise privacy policy before service use. In our architecture, the PPM provides two steps: initial creation of a privacy policy and customization of the privacy policy.

A hierarchical structure is used to define a privacy policy $\mathcal{P}$ in the PPM. Let $P_i \in \mathcal{P}$ $(0 \leq i \leq l_i)$ be the ith item in the policy, and P_i has sub-items P_{ij} $(0 \leq j \leq l_{ij})$. If $P_0 = A$, then all items are allowed. In a similar fashion, P_{ij} has sub-items P_{ijk}. If the parent item is A, then all child items are A. In the initial policy creation, a user defines a policy for each top-level item such as $P_1 = A, P_2 = \neg A, P_3 = \neg A, ...$, where A denotes "allowed to send", and $\neg A$ denotes "not allowed to send". For example, if P_1 is a policy governing location information from the GPS of the user's terminal, the location information can be sent to all service providers. Thus, almost all items in the initial policy are defined as $\neg A$.

When a user uses a location-based service, the PPM receives a request with a permission description D_x (that is the same as a description of privacy policy items) from the service provider, and checks the privacy policy governing location information. For example, the permission description is denoted as "$D_x = P_{1433}$: Brief location information (town level) for a trust level 3 service provider". Let b_x be a feedback from the user for the permission description D_x. If the policy governing location information says that $P_1 = \neg A$, the PPM asks the user whether the permission described is allowed ($b_x = 1$) or not ($b_x = 0$). If the user grants permission, the location information is sent and the privacy policy is updated to add the item P_{1433}. Thus, the policy is modified as $f_m(\mathcal{P}, D_x, b_x) = \{P_1 = \neg A, P_{14} = \neg A, P_{143} = \neg A, P_{1433} = A\}$, where $f_m(*, *, *)$ is a modification function of the privacy policy. Then other items such as P_{1432} are implicitly configured as $\neg A$. For precise and usable privacy policy setting, we need to define groups of service providers. The above example case includes "trust level" as an index for grouping. The trust level should be defined by a trusted entity such as a rating agency selected by the users as trustworthy. Another case is to define two groups: a group of service providers in the same domain and a group of service providers out of the domain. This definition is reflected by "opt-in domain" explained in section 3.

We also consider a policy recommendation, which can be presented during the initial policy creation. The PPM shows an individual an example privacy policy of a user who has a similar profile as the individual, or a typical policy setting. Furthermore, it is possible to recommend modification of the privacy policy based on the privacy policies of similar users. Kelley *et al.* presented a user-controllable policy learning approach that involves neighborhood users' searches to explore incremental modifications of the user's current policy [31] and applied it to the people-finder [35]. We apply a similar technique to our policy generation and modification to reduce the complexity of operations required of users. Privacy policies should be encrypted by the master key K_m and stored in the user privacy policy database.

4.4 User-Friendly Interface for Privacy Policy

One problem with privacy policy management by the user is the complicated descriptions required in a privacy policy. To realize user-friendly privacy policy management, we should consider two technical issues: (1) the policy should be easy to configure, and (2) the policy should be easy to understand. Issue (1) was discussed in the previous section, so we mainly discuss issue (2) in this subsection.

To realize an easily-understood overview of a policy, a transforming function $f_w(x, y, z)$ from the machine-friendly format to a user-friendly format is needed. We consider the use of a layered view of a privacy policy. The initial view of the privacy policy is the top level, and the relevant part of the second level of the privacy policy is shown when a user clicks a certain top level item. The item that describes exceptions to items should be shown and other common items are displayed as one-paragraph descriptions. Let d be the level of description, and u be a user preference for a view of a privacy policy. The function $f_w(*, *, *)$ outputs a privacy view $\mathcal{P}_{view}$ as $f_w(\mathcal{P}, d, u) = \mathcal{P}_{view}$. For example, $f_w(\mathcal{P}, d_1, u_k) = \{P_0 = A, P_1 = \neg A, P_2 = \neg A, ..., P_{l_i} = \neg A\}$, where d_1 is the top-level of the privacy policy $\mathcal{P}$, and u_k is a preference of a user k. In the example case in the previous subsection, $f_w(\mathcal{P}, d_4 = 143*, u_k = all) = \{P_{1433} = A, others = \neg A\}$, because common items are merged. The policy view is optimized for each user, by using the user preference that is based on requirements and feedbacks input from users.

Another point is the description of each item; user-friendly description should be used to indicate the item. One useful technique is to highlight critical parts or unusual parts of the policy. For example, when items that many other users agree to is also agreed to by the user, the items are indicated with green color, and an item that includes critical private information or where many other users disagree is indicated red when the user agrees to the unusual item. A detailed design for privacy policy visualization is an open issue for our future research.

4.5 Log Management

The PPM stores records of private information flows that include pseudonym ID, date and time, service provider name, and types of private information that have a structure similar to those mentioned in the privacy policy definition, but not include private information itself. The records are written into the service log DB of the PPM. Users can search their own records using retrieval keys: user ID and service provider name, user ID and type of private information, and three keys of user ID, service provider name, and type of private information within a given range of dates and times. The database should be encrypted and protected against external attackers.

User Consent Log Search. The PPM cannot trace or search user logs without the consent of the user. User authentication is required to search the database and a user provides *user secret* to the PPM during the authentication process. The PPM cannot compute a secret key K_s to generate the pseudonym ID without

user consent, because the secret key K_s is needed to generate a pseudonym ID. Thus, it is impossible for an *offline attacker* such as a curious operator of the PPM to search a particular user's records. There are some cryptographic techniques for private search, but those schemes requires heavy computational costs; thus ,we design a lightweight user consent log search scheme.

4.6 Interoperability Between PPMs

A user accesses a PPM in a domain that the user belongs to, and uses several services in the same domain. For more general use cases, we should consider the case in which the user may access other PPMs in different domains. To build PPMs in different domains, we realize a distributed architecture of PPMs, and a concentration of transaction to a PPM is avoided. The PPM has connections with service providers in the same domain, and privacy policy formats for the service providers, but the PPM may not have access to service providers in different domains and appropriate privacy policy information about the service provider. Therefore, users have to use a separate PPM in each domain. When the user accesses a PPM in a different domain, that PPM has to transfer the user's request to the PPM in the user's home domain. This situation is similar to roaming schemes for user authentication. Protocols should be designed for the interoperability of PPMs. There are four protocols for realizing interoperability.

- A protocol for requesting user authentication from the home PPM.
- A protocol for downloading a user privacy policy from the home PPM.
- A protocol for uploading a modified user privacy policy to the home PPM.
- A protocol for sending logs of service use to the home PPM.

After user authentication is successful, the PPM downloads the user privacy policy from the home PPM of the user. The user privacy policy is modified, where the user privacy policy does not include permission items needed for service use. The PPM asks the user whether the permission items are allowed, and if so, adds the permission items to the user's privacy policy. The modified user privacy policy should be uploaded automatically to the home PPM, and the old user privacy policy is then replaced by the uploaded privacy policy. A public key infrastructure is needed for mutual authentication between PPMs. A private key is securely embedded in the PPM. The general steps of the protocols are as follows;

1. PPMs establish a secure channel to execute an authenticated key exchange protocol including public-key-certificate-based mutual authentication such as the ephemeral DH with RSA certificates mode (DHE-RSA) in TLS 1.2 [16]. Each PPM is endorsed by a trusted entity and holds a valid certificate, so a PPM can authenticate other PPMs to execute the authenticated key exchange protocol.
2. PPMs communicate with other PPMs using the protocols. All transaction data is sent via a secure channel, so all transaction data is securely protected.

5 Related Work

In this section, we introduce related work regarding an architecture for privacy preserving services.

5.1 ID management for Privacy Protection

The Identity management (IdM) [9] technique is a method to control user information, which was originally developed for intra-net use. The concept of a user-centric IdM [18, 2] is also one of the most important features for privacy protection on IdM. Under this concept, users have the right to control their identities, which are shared among ID providers (IDPs) and service providers (SPs). Therefore, IDP requires user permission, if it is to provide user information to SPs. J. Altmann *et al.* have proposed a user centric framework for IdM [2]. The framework provides comprehensive IdM for users and protects user concerns without revealing business interests.

5.2 Privacy Policy Management

The Privacy Bird [11, 12]is an extension of a web browser and automatically retrieves the P3P policies of a web site. However, Kolter and Pernul[33] suggested that the available privacy preference settings of the Privacy Bird result in inadequate user acceptance, putting the ultimate goal of real-world use at risk. Thus, they proposed a user-friendly, P3P-based privacy preference generator [33] for service providers, including a configuration wizard and a preference summary.

Yee presented a privacy policy checker [47] for online services. The checker compares user privacy policy with provider privacy policy and then automatically determines whether the service can be used. Biswas presented an algorithm [7] that detects conflicts of privacy settings between user preference and the requirements of an application on a smart phone. Privacy Butler [46] is a personal privacy manager that can monitor a person's online presence and attempt to make corrections based on a privacy policy for user's online presence in a social network. The concept of the Privacy Butler is similar to the concept of our project, but it focuses on modifications to content hosted by social networking services; it monitors whether the modification is a satisfactory match for the privacy policy. Privacy Mirror [8] is a tool that is intended to show users what information about them is available online.

Some languages to describe privacy policies have been presented in [10, 14, 6]. Backes *et al.* examined some comparisons of enterprise privacy policies using formal abstract syntax and semantics to express the policy contents [4].

The objectives of the VRM project [42] are making individuals the collection centers for their own data and giving them the ability to share data selectively, controlling how their data is used by others, and asserting their own terms of service. The concept is based on Personal Data Service (PDS) and it is essentially similar to our proposal, but their approach is a combined model of access control and data storage.

5.3 Privacy Preserving Techniques

Adnostic [39] is a privacy-aware accounting tool to correctly bill advertisers without leaking the private information identifying which user clicks on what ads. RePriv [22] provides a verified miner tool through a browser plug-in that allows a user to control how much private information leaves through a browser and to which web site. Guha *et al.* presented a way of disguising the user's identity and obfuscating private information before releasing it [27]. Hardt and Nath proposed a flexible framework [28] for personalizing ad delivery to smart phones. They proposed a differentially-private distributed protocol to compute various statistics required for their framework. Kido *et. al.* proposed a *false dummy method* [32], where a user sends n different locations to a location database server, with only one of them being correct (the rest are "dummies" that mask the true location). Hong and Landay introduced an architecture based on *landmark objects* [29], in which users refer to the location of a significant object (landmark) in their vicinity, rather than sending an exact location. This scheme makes it difficult to control the granularity of location information and thus may not be suitable for some types of location-based services. Recent research [37] has focused on establishing location anonymity in the spatial domain. Gruteser and Grunwald [25] suggested "blurring" the user's location by subdividing space in such a way that each subdivision contains at least $k-1$ other users. Gedik and Liu [23] adapted this to allow users to be assigned personalized values of the masking parameter k. Mokbel *et. al.* presented a hierarchical partitioning method to improve the efficiency of location perturbation [38]. Selection of optimal subdivision spaces was investigated in [36, 5]. In [24] a decentralized approach without an anonymizer was considered in order to realize good load balancing. However, communication between users is required to calculate the anonymized location information. Ardagna *et al.* presented a location obfuscation [3] that provides privacy-preserved location information without relying on trusted entities. Perturbation methods[20, 45, 17] iare used for adding a random noise as chaff in an interactive setting.

6 Discussion and Conclusion

We have presented an architecture for privacy-preserving services and designed a core module PPM. The PPM supports privacy management by users and acts a proxy that checks flows of private information and records them. Our concept is a delegation of access control and policy management that are inconveniently complex for users to a trusted third party, even though our architecture is based on the concept of PDS. In the architecture, users can verify that private information flows use the service log database. Furthermore, our design of the PPM includes consideration of a potential *offline attacker* in order to ensure the security of the PPM. Availability and flexibility are achieved by the PPM which provides centralized management of common user privacy policies and a policy checking mechanism that refers to the common policies. As suggested in published studies [15], it is expected that users should be able to easily use

user-centric services to provide a clear view of information flows and to ensure access control based on their own privacy policies.

We are implementing a prototype system using the PPM and plan to conduct a demonstration experiment using the prototype system in a large shopping area.

Acknowledgment. This work has been supported by the New Energy and Industrial Technology Development Organization (NEDO) funded project, "Development and demonstration of life support services that facilitate movement around stations by utilization urban special information."

References

1. A. Acquisti and J. Grossklags. Privacy and rationality in individual decision making. *Security Privacy, IEEE*, 3(1):26 –33, 2005.
2. J. Altmann and R. Sampath. Unique: A user-centric framework for network identity management. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 495–506, 2006.
3. C.A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. An obfuscation-based approach for protecting location privacy. *Dependable and Secure Computing, IEEE Transactions on*, 8(1):13 –27, 2011.
4. Michael Backes, Günter Karjoth, Walid Bagga, and Matthias Schunter. Efficient comparison of enterprise privacy policies. In *Proceedings of the 2004 ACM symposium on Applied computing*, SAC '04, pages 375–382, 2004.
5. B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proc. of 17th International World Wide Web Conference (WWW 2008)*, pages 237–246, 2008.
6. K. Bekara, Y. Ben Mustapha, and M. Laurent. Xpacml extensible privacy access control markup langua. In *Communications and Networking (ComNet), 2010 Second International Conference on*, pages 1 –5, 2010.
7. D. Biswas. Privacy policies change management for smartphones. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*, pages 70 –75, 2012.
8. Markus Bylund, Jussi Karlgren, Fredrik Olsson, Pedro Sanches, and Carl-Henrik Arvidsson. Mirroring your web presence. In *Proceedings of the 2008 ACM workshop on Search in social media*, SSM '08, pages 87–90, 2008.
9. David W. Chadwick. Federated identity management. *Foundations of Security Analysis and Design V*, pages 96–120, 2009.
10. L.F. Cranor. P3p: making privacy policies more useful. *Security Privacy, IEEE*, 1(6):50 – 55, 2003.
11. Lorrie Faith Cranor, Manjula Arjula, and Praveen Guduru. Use of a p3p user agent by early adopters. In *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, WPES '02, pages 1–10, 2002.
12. Lorrie Faith Cranor, Praveen Guduru, and Manjula Arjula. User interfaces for privacy agents. *ACM Trans. Comput.-Hum. Interact.*, 13(2):135–178, 2006.
13. Project Danube. Danube, identity and communication for political and social innovation. *Project Danube Web Page, http://projectdanube.org/*, 2010.
14. A. Dehghantanha, N.I. Udzir, and R. Mahmod. Towards a pervasive formal privacy language. In *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on*, pages 1085 –1091, 2010.

15. Andrè Deuker. Addressing the privacy paradox by expanded privacy awareness - the example of context-aware services. *Privacy and Identity Management for Life*, 320:275–283, 2010.
16. T. Dierks and E. Rescorla. The transport layer security (TLS) protocol version 1.2. *Internet Engineering Task Force (IETF), RFC5246*, 2008.
17. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of TCC 2006, LNCS*, volume 3876, pages 265–284, 2006.
18. T. Eap, M. Hatala, and D. Gasevic. Enabling user control with personal identity management. In *Services Computing, 2007. SCC 2007. IEEE International Conference on*, pages 60–67, 2007.
19. D. Estrin. Participatory sensing: applications and architecture [internet predictions]. *Internet Computing, IEEE*, 14(1):12 –42, 2010.
20. S. E. Fienberg and J. McIntyre. Data swapping: Variations on a theme by dalenius and reiss. In *Proc. of PSD 2004, LNCS*, volume 3050, pages 14–29, 2004.
21. The Eclipse Foundation. Higgins, personal data service. *Higgins Home, `http://www.eclipse.org/higgins/`*, 2009.
22. Matthew Fredrikson and Benjamin Livshits. RePriv - re-envisioning in-browser privacy. In *Microsoft Research Technical Report, MSR-TR-2010-116*, 2010.
23. M. Gedik and L. Liu. A customizable k-anonymity model for protecting location privacy. In *Proc. of the 25th International Conference on Distributed Computing Systems (ICDCS 2005)*, pages 620–629, 2005.
24. G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVÉ: Anonymous location-based queries in distributed mobile systems. In *Proc. of 16th International World Wide Web Conference (WWW 2007)*, pages 371–380, 2007.
25. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, pages 163–168, 2003.
26. Saikat Guha, Bin Cheng, and Paul Francis. Challenges in measuring online advertising systems. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, IMC '10, pages 81–87, 2010.
27. Saikat Guha, Alexey Reznichenko, Kevin Tang, Hamed Haddadi, and Paul Francis. Serving ads from localhost for performance, privacy, and profit. In *Proc. fo the 8th ACM Workshop on Hot Topics in Networks (HotNets-VIII), HOTNETS '09*, 2009.
28. Michaela Hardt and Suman Nath. Privacy-aware personalization for mobile advertising. In *Proceedings of the 2012 ACM conference on Computer and communications security*, CCS '12, pages 662–673, 2012.
29. J. I. Hong and J. A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proc. of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys 2004)*, pages 177–189, 2004.
30. Carlos Jensen, Colin Potts, and Christian Jensen. Privacy practices of internet users: self-reports versus observed behavior. *Int. J. Hum.-Comput. Stud.*, 63(1-2):203–227, 2005.
31. Patrick Gage Kelley, Paul Hankes Drielsma, Norman Sadeh, and Lorrie Faith Cranor. User-controllable learning of security and privacy policies. In *Proc. of the 1st ACM workshop on Workshop on AISec*, AISec '08, pages 11–18, 2008.
32. H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Proc. of IEEE International Conference on Pervasive Services 2005 (ICPS 2005)*, pages 88–97, 2005.

33. J. Kolter and G. Pernul. Generating user-understandable privacy preferences. In *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, pages 299 –306, 2009.
34. Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops*, ICDMW '10, pages 474–482, 2010.
35. Jialiu Lin, Guang Xiang, Jason I. Hong, and Norman Sadeh. Modeling people's place naming preferences in location sharing. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, Ubicomp '10, pages 75–84, 2010.
36. S. Mascetti and C. Bettini. A comparison of spatial generalization algorithms for lbs privacy preservation. In *Proc. of the 1st International Workshop on Privacy-Aware Location-Based Mobile Services (PALMS 2007)*, pages 258–262, 2007.
37. M. F. Mokbel. Towards privacy-aware location-based database servers. In *Proc. of the 22nd Internationl Conference on Sata Engineering Workshops (ICDEW 2006)*, pages 93–102, 2006.
38. M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In *Proc. of the 32nd International Conference on Very Large Data Bases (VLDB 2006)*, pages 763–774, 2006.
39. Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, and Dan Boneh. Location privacy via private proximity testing. In *Proc. of the Network and Distributed System Security Symposium, NDSS 2011*, 2011.
40. A. Pedersen. P3 - problems, progress, potential. *Privacy Laws & Business International Newsletter*, 2:20–21, 2003.
41. Irene Pollach. What's wrong with online privacy policies? *Commun. ACM*, 50(9):103–108, 2007.
42. Doc Searls. Project vrm - vendor relationship management -. *Project of the Berkman Center for Internet Society at Harvard University*, 2013.
43. Daniel J. Solove. Privacy self-management and the consent paradox. *Harvard Law Review*, 126, 2013.
44. W3C. The platform for privacy preferences 1.0 (P3P1.0) specificati. In *Platform for Privacy Preferences (P3P) Project*, 2002.
45. W. E. Winkler. Masking and re-identification methods for public-use microdata: Overview and research problems. In *Proc. of PSD 2004, LNCS*, volume 3050, pages 231–246, 2004.
46. R. Wishart, D. Corapi, A. Madhavapeddy, and M. Sloman. Privacy butler: A personal privacy rights manager for online presence. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pages 672 –677, 2010.
47. G.O.M. Yee. An automatic privacy policy agreement checker for e-services. In *Availability, Reliability and Security, 2009. ARES '09. International Conference on*, pages 307 –315, 2009.