A Conceptual Framework for Privacy Policy Negotiation in Web Services

C.Wu, V.Potdar and E.Chang

Centre for Extended Enterprises and Business Intelligence, Curtin Business School,
Curtin University of Technology, Western Australia
e-mail: {Chen.Wu, Vidyasagar.Potdar, Elizabeth.Chang}@cbs.curtin.edu.au

Abstract

Research into privacy in web services based service-oriented environment gained attention in recent years. Business Transaction Level Data (TLD) privacy is important because in web services the interaction between the Service Provider and Service consumer is far more complicated than in the browser-server environment. This results in an enormous amount of data, and complex data, which raises many transaction level data privacy issues. In web services we can define arbitrary transaction inter-faces and hence, the privacy concerns and associated complexity increases. The existing privacy solutions only offer session level data privacy; therefore, we extend this solution by adding transaction level data privacy. This would offer the service provider and consumer more control over their privacy data, and so that is the difference between existing privacy negotiation protocols and new generation service oriented based privacy protocols. In this paper we tackle this issue of privacy policy negotiation in the distributed service-oriented computing environment. To solve this privacy issue we propose a framework that would negotiate and generate dynamic transaction-based privacy policies based on transaction-related confidential data and its associated privacy preferences. A detailed protocol and supporting context is provided to illustrate the applicability of our proposed framework.

Keywords

Privacy Policy, Web Services, Privacy Policy Negotiation, Transaction Level Data Privacy.

1. Introduction

The privacy issue has been extensively studied in the literature. An early definition of privacy can be found in Westin (1967), who defines privacy as “the claim of individuals, groups, or institutions to determine for themselves when, how and to what extent information about them is communicated to others”. Clarke (2000) defines “personal privacy” as the interest that individuals have in sustaining a ‘personal space’, free from interference by other people and organizations. While these definitions give broad explanation for general privacy, data privacy, in particular, is considered an exceedingly critical ethical issue raised by the developers of modern computing technology during the information age (Ashrafi & Kuilboer, 2005; Moor, 1997).

The notion of using policy to address privacy issue is not new. In general, a privacy policy defines how an individual’s personal information may be used. A *technical privacy policy* is essentially a formal specification composed of expressive statement and rules that indicate privacy preferences for both the information gatherer and the information provider. For instance, one of the most well-known technical privacy policy specifications – the *P3P*

(Platform for Privacy Preferences (P3P) Project, 2004) – utilizes the XML representation to describe what information is with which data, how long the data will be kept, how the data will be shared and the measures taken to protect it, etc (Ashrafi & Kuilboer, 2005). In general, P3P is a simple Internet standard aiming at facilitating the exchange of information about website privacy politics. It is comprised of a set of standardized multiple-choice questions, covering all the major aspects of a Web site's privacy policies in the XML representation. On the server side, a P3P specification provides to its user clear information about how a website treats personal data gathered during the user experience. The privacy policy negotiation occurs on the client's side where P3P compliant browsers (e.g. Microsoft IE 6) parse such policy information and compare it to the consumer's own set of privacy preferences described in APPEL (Cranor et al, 2002) specification language.

When thinking of leveraging P3P approach to address the privacy issue in web services environment, we can find one issue: the current P3P specification solely addresses those service-independent built-in data, for example, the user name, user gender and click stream etc. This kind of data is applicable through multiple interactions on the same connection between client and server, and is orthogonal to the actual business data concerned by service providers and consumers. Since the term 'session' is often used to represent an enduring connection between a user (agent) and a server, we call such a data Session Level Data (SLD). We further argue that SLD is very different from the actual business data transferred during business transactions in terms of the privacy issue. Consequently, such a difference motivates us to propose a privacy solution addressing the privacy issue for business data, which is termed as Transaction Level Data (TLD) in this paper. Moreover, in a Service-Oriented Environment (SOE), each service is explicitly described by publicly accessible interfaces (e.g. WSDL) that enclose arbitrary user-defined operations with various complex data types. This not only complicates the TLD being exchanged but makes it too easy to disclose sensitive data transferred between providers and consumers. To our best knowledge, such a TLD privacy issue is not adequately tackled by P3P specifications. As indicated in (Thibadeau, 2000), P3P focuses only on online websites; it does not support negotiation of privacy policies between the service provider and consumer in the SOE. Hence in this paper, we mainly deal with business data privacy issues in a service-oriented environment. We assume that any operation (regardless of its success or failure) constitutes a privacy problem if it attempts to facilitate misuse or unauthorized leakages of business-sensitive information including profile information among multi-party transactions throughout the entire interactions between service providers and consumers.

2. Related Work

Most of the research on privacy policy points to the issue of policy negotiation, a topic that has been comprehensively investigated in distributed systems. Anderson (2004) introduced web services policy language (WSPL), which supports policy negotiation by merging policies from two sources. In particular, the simple negotiation steps are also given. One of the important issues that the author presents is the dynamic policy negotiation, which is performed at runtime based on dynamic constraints per service request. In WSPL initial policies contain all possible combinations of parameters, and hence, support the determination of all mutually acceptable policy parameters on the first round with no further incremental negotiation. Such a general policy negotiation process is very different from the negotiation proposal presented in our research, in which incremental parameters are negotiated when initial policies fail to match exactly. This is due to our consideration that in the highly distributed SOE, negotiation shall be carried out in an extended manner between heterogeneous service providers and consumers. Next, Barrere et al. (2003) proposed a solution to make an administrator located in different domains agree on a common dynamic security policy. Furthermore, inter-domain policy dynamic negotiation is fostered in architecture consisting of distributed domains rather than relying exclusively on the centralized global repository. Lastly, Chang et al. (2003) describe a solution for managing security policies in a large distributed web services environment. It allows the collaboration parties to negotiate and establish security policy dynamically for each individual interoperation. However, they did not give detailed elaboration on the policy integration protocol and algorithm.

On the other hand, research on privacy issues in service-oriented environment still lies in its initial stage. A recent survey study of privacy issue in web services environment was presented by Hung et al. (2004). The authors suggested several research directions in this area. One of the most promising directions is the privacy negotiation. In particular, Korba (2002) described privacy in a distributed electronic commerce environment. The author proposed an agent-based negotiation approach to integrate the privacy policy from disparate organizations from different countries, where different privacy laws are enacted. However, the author does not consider contextual relation when privacy policy is created, namely the dynamic negotiation, but our model is dynamic. We add a new dimension to the static aspect and consider the requirement for a context to design our privacy template policy. Moreover, El-Khatib (2003) proposed a new privacy negotiation protocol for web services. The proposed protocol enables the generation and negotiation of a bilateral privacy between consumers and service providers using an extended version of P3P privacy policy description language. This research is the closest work to our research. However, the author has not discussed how the service consumer and provider can reach an agreement in terms of technical detail and only gives a conceptual high abstract description. We provide a detailed description on how they agree on the shared understanding of privacy policy.

3. Proposed Framework

In this section we propose our conceptual framework. The aim of this framework is to generate transaction-based privacy policy based on the provider's and consumer's privacy preference in the Service-Oriented Environments – the underlying underpinning and the privacy research context for our proposal. The framework is composed of four major layers, Privacy Platform, Policy Specification, Privacy Policy Negotiation, and Policy Generation, which are shown in Figure 1. Before discussing this framework in detail, we would like to introduce the preliminary concept of transaction and Transaction Level Data (TLD) privacy followed by a thorough description of these components.

3.1 Term Definition

Transaction in our research is defined as a one-to-one interaction between the service provider and service consumer, which involves multiple operations selected from WSDL in an order that fulfills business requirements.

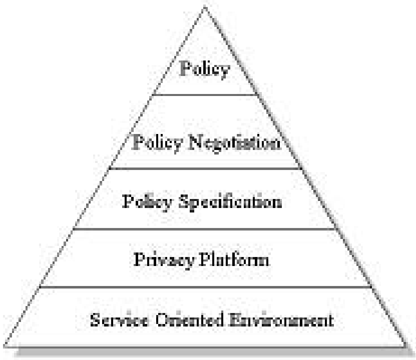

Figure 1: Proposed Conceptual Framework

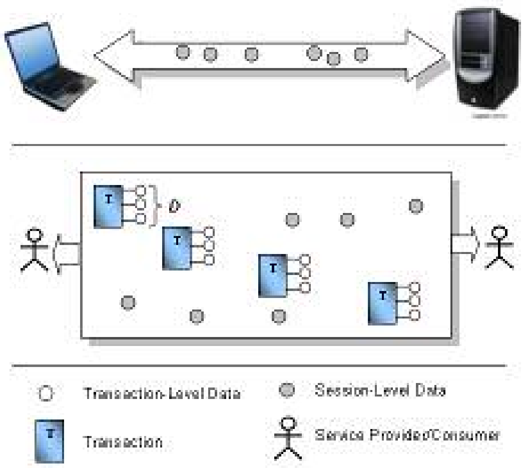

Figure 2: SLD Privacy vs. TLD Privacy

As illustrated in the middle part of the Figure 2, each transaction has corresponding TLD tightly associated with it. The TLD is in stark contrast with the SLD, which scatters within the whole session and does not belong to any particular transaction. SLD is shared by all the transactions (or even by all the sessions between one pair of service provider and consumer). They are relatively predetermined, and hence, can be stated in a general specification such as P3P. The TLD can be further formulated as a set of all input and output parameters that are required to complete one transaction when it is invoked by a service consumer. The privacy of this TLD is termed as TLD Privacy. For the service consumer, this indicates the way in which private information will be handled; while for the provider, this indicates preferences for the way in which information will be handled. An example of TLD Privacy is illustrated in Table 1. In transaction level data privacy, the consumer and provider are aware of the data set that are at risk, e.g. In the case of “Querying the rent of a Warehouse”, the consumer has to disclose the quantity of shipment while the provider has to disclose the warehouse rent, location and size.

Transaction	Service Provider	Service Consumer
Querying the Rent of a Warehouse	Rent Size Address	Type of Goods Volume of Goods

Table 1: Example of Transaction Level Data in Service Oriented Environment

3.2 Privacy Platform

The privacy platform acts as a bottom layer on which the privacy policy and policy negotiation works. In this framework we propose a solution to the problem of how privacy policy can be negotiated (or integrated) in a real time environment so that they can be in compliance with cross domain privacy regulations. For example, in some domains (or different countries legal frameworks) *analyzing consumer behavior* wouldn’t be a breach of privacy regulations, but for other domains it may cause serious concerns. This issue could be a major concern for logistics companies with global presence. To solve this privacy issue we propose this framework which would generate a dynamic privacy policy based on domain specific privacy regulations.

3.3 Privacy Policy Specification

The privacy policy specification would comply with the P3P standard. However, due to the reason we have mentioned earlier, the existing P3P file lacks the ability to thoroughly tackle the transactional-based data privacy issue. Hence, in this research we extend P3P specification in a way that application-level data privacy is well fostered. For instance, each datum exchanged during the transaction can have an associated privacy policy. We also note that each transaction would have at least two privacy polices, one from the consumer domain and the other from the provider domain. Whenever a transaction is invoked, the consumer first checks the privacy policy of the provider for that transaction. We assume that the provider posts its transaction privacy policies. The consumer would parse the privacy policy of the provider to compare it with its own privacy policies. If the policies of the consumer and the provider match for the transaction, the transaction can further proceed. If, however, the policies fail to match in the first round, then the Privacy Policy Negotiation is needed as described in the section 3.4.

3.4 Privacy Policy Negotiation

As discussed earlier, a transaction can be seen as a functional graph that includes all the involved operations provided by the service providers. Note that to consider privacy issues in the service-oriented environment, we make the following assumption: service provider will not publicly expose its operation through WSDL unless it considers no adversary effect on its own data privacy issue after 'opening' that operation. Hence, we conclude that during any interactions between service consumer and provider, it is the service consumer whose data privacy will always be jeopardized rather than the service provider. However, the service provider publishes its privacy policy regarding the privacy data sent from and to the service consumer. Meanwhile, data privacy on the consumer side always lies at the transaction level, thus for the same consumer's private data it may present different preferences. Such dynamic transaction-level privacy policy is suitable for service-oriented computing and the distributed business environment. The transaction protocol is classified into five stages, which are, *firstly* Generating Consumer's Privacy Preferences, *secondly* Searching Service Providers, followed by Selecting Service Providers, then Negotiating Privacy Preferences, and finally Invoking Privacy Preference Compliant Service Providers.

Step 1: Assigning Consumer's Privacy Preferences

For a transaction T the service consumer identifies a list of parameters which would affect the privacy of its own data. This collection is termed as fixed data set D where $D = \{D_1,.......D_n\}$, which is regarded as risky in terms of data privacy. D_i is a set of S_m where S describes the properties which would contribute to the data privacy issues $D_i = \{S_1,...., S_m\}$. S is a set of V, which encompasses a list of probable values that S_m can have. For example if D = {Rent, Size, Address}, then D_i = *Rent i.e.* D_1*={Purpose, Recipient} then* S_2*={"Ours", "*3^{rd} *Party"}*. We now construct n level hash table $1<n<3$ (as shown in Table 2), for the first level hash table H_h^1 the key is D_i and the value is a nested hash table H_h^2 whose key is S_i and the value is V_i.

This can be represented using XML which can be stipulated by XML schema. This is the structure of the policy preference from the consumers service request subscription based on one transaction. A similar structure can be found for the providers published policy.

<table>
<tr><td colspan="3">Hash Table 1 H_h^1</td></tr>
<tr><td><i>Key</i></td><td colspan="2"><i>Values</i></td></tr>
<tr><td rowspan="4"><b>Rent</b></td><td colspan="2">Hash Table 2 H_h^2</td></tr>
<tr><td><i>Key</i></td><td><i>Values</i></td></tr>
<tr><td><b>Recipient</b></td><td><b>Ours</b></td></tr>
<tr><td></td><td><b>3rd Party</b></td></tr>
</table>

Table 2: Consumers Policy Preference Structure

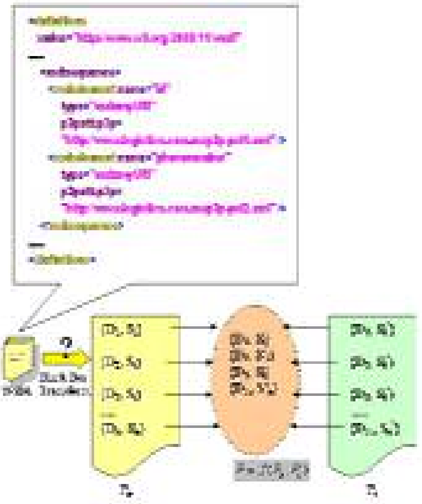

Figure 3: Transaction Data Level Privacy Policy Generation

Step 2: Searching Service Providers

In this phase the functional requirements are gathered and the appropriate service provider is searched. This is where the service providers are selected based on the functional requirements of the desired transactions, e.g. the consumer only looks for those providers who can fulfill the transaction without considering other factors like performance, scalability, cost and privacy. Since each transaction is composed of several operations, the service consumer finds a list of all possible candidate service providers who would fulfill the functional requirements listed in the transaction

Step 3: Selecting Service Provider

For each candidate provider, the service consumer retrieves its privacy profile P_p, which is a P3P-compliant WSDL file, and converts it back to D and S so that the comparison with P_c can be carried out. For this paper we consider such a converting as a black box. (This discussion is out of the scope of this paper). However, it can be achieved by scanning the in and out message types for each WSDL operation by existing WSDL parsers, e.g. WSDL4J The algorithm in this black box is currently being conducted in the solution proof-of-concept implementation. Suppose P_c is given by the following relation $(D_1 \rightarrow S_1, D_2 \rightarrow S_5, D_3 \rightarrow S_1)$ and P_p is given by the following relation $(D_1 \rightarrow S_1, D_2 \rightarrow S_2, D_3 \rightarrow S_1)$. If we compare the privacy preference associated to D_1 and D_3 we identify that the privacy preferences matches; however, for D_2 the privacy preference doesn't match. If this happens then the consumer has one of three options: 1) changing the provider, 2) giving up the transaction, or 3) negotiating with the provider. If the consumer chooses the first option then the consumer can find another provider, and this would enable a return to step 3. However, if the consumer opts for the second option, then the whole transaction ceases immediately. Nevertheless, if the consumer chooses the last option then it has to begin a negotiation with the current provider to arrive at a mutual agreeable policy P. This negotiation step is discussed next.

Step 4: Privacy Preference Negotiation

In this step the consumer and provider negotiate to arrive at a consensus. The consumer sends a request to the provider to modify its existing privacy policy for the transaction T. The provider can either accept or reject a consumer's request based on its own privacy concerns and domain specific legal regulations. If the provider rejects the offer altogether then the consumer has to opt for another service provider. However, if the provider wants to accept these modifications completely, then the provider would send an acceptance response. Although, in the case of a provider not being ready to accept all the suggested modifications, then it would rather send a request to the consumer to accept the partially modified privacy profile $P_p^{'}$. This process continues until both consumer and provider reach an agreement. When they reach consensus, a mutually agreeable session based privacy policy P is generated. It is termed as session based because the same transaction can be carried out at two different sessions into which the privacy preferences may have changed. This policy P is now used to manage the privacy concerns of both the consumer and producer. This process is shown in Figure 3.

Step 5: Invocating Privacy Preference Compliant Service Provider

This is the last step of the protocol. The main input to this step is the mutually agreed privacy policy P. Based on this privacy policy, the consumer can now invoke the service provider to initiate a transaction. Here both the parties agree to follow this privacy policy. Once the transaction is completed, this session privacy policy will be destroyed or it would be stored in the producer's and consumer's database for any future transactions.

4. Application

In this section we discuss how the proposed framework can be applied to distributed logistics network. Suppose we have one logistics service provider (called NSW) in the state of New Southern Wales, and a logistics service consumer in a Western Australia (called WA). WA wants to initiate a transaction and is looking for a service provider who offers this service. Suppose WA finds a list of candidate service providers, which includes NSW. Let us consider that WA is looking for a service provider to offer warehouse space for a shipment it has to deliver. This simple transaction can involve many privacy issues. These issues are listed in Table 1 Section 3. We can find that WA will disclose shipment's size, whereas NSW needs to disclose the size, location and rent fee of the warehouse. Both entities have a sense of privacy concern with this transaction and need to be aware of how their private data would be handled in this transaction. This is where the protocol's first step begins. The consumer i.e. WA identifies its privacy preferences for this transaction and assigns privacy preference values to it. This is the privacy preference profile of WA. Now WA starts searching for service providers who can match these privacy preference requirements and gets a list of providers who match most of the requirements. The candidate provider (e.g. NSW) who matches most of the requirements is selected in the first round. WA then asks for the privacy policy of NSW and compares it with its own privacy preferences. If the preference doesn't match, then WA enters into a negotiation with NSW to modify its policy to cater for its needs.

This negotiation step is the Step 4 in the protocol. WA and NSW negotiate and eventually generate the session based privacy policy which would then be used for the transaction. Once the negotiation is successful WA can invoke the NSW's service to complete the transaction.

5. Conclusions

In this paper we described a framework and presented a protocol to generate session privacy policy based on the provider's and consumer's privacy preferences in distributed environments. We showed how two entities, i.e. a service provider and service consumer, get acquainted with each other's privacy policy and how they make a decision (or a compromise on some privacy aspects) to proceed with a transaction. The transactions level data would determine whose privacy is at a risk. We discussed the detail negotiation protocol and described how the privacy policy is accessed, modified and mutually agreed for an individual transaction.

6. References

Anderson, AH, (2004), 'An Introduction to the Web Services Policy Language (WSPL)', in *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*.

Ashrafi, N & Kuilboer, JP, (2005), 'Privacy Protection via Technology: Platform for Privacy Preferences (P3P)', *International Journal of E-Business Research,* vol. 1, no. 2, pp. 56 - 69. Retrieved: April-June 2005, from

Barrere, F, Benzekri, A, Grasset, F, Laborde, R & Nasser, B, (2003), 'Inter-Domains policy negotiation', in *the 4th International Workshop on Policies for Distributed Systems and Networks*.

Chang, S, Chen, Q & M., H, (2003), 'Managing Security Policy in a Large Distributed Web Services Environment', in *the 27th Annual International Computer Software and Applications Conference (COMPSAC'03)*.

Clarke, R, (2000), 'Beyond the OECD Guidelines: Privacy Protection for the 21st Century', http://www.anu.edu.au/people/Roger.Clarke/DV/PP21C.html visited 28/8/2005.

Cranor, L, Langheinrich, M & Marchiorit, M, (2002), *A P3P Preference Exchange Language 1.0 (APPEL1.0)*. Retrieved: 7/04/2006, from http://www.w3.org/TR/P3P-preferences/.

El-Khatib, K, (2003), 'A Privacy Negotiation Protocol for Web Services', in *Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments*, Halifax, Nova Scotia, Canada.

Hung, PCK, Ferrari, E & Carminati, B, (2004), 'Towards standardized Web services privacy technologies', in *The 2nd IEEE International Conference on Web Services*, U.S.A, pp. 174 - 181.

Korba, L, (2002), 'Privacy in Distributed Electronic Commerce', in *the 35th Hawaii International Conference on System Sciences*.

Moor, JH, (1997), 'Towards a Theory of Privacy in the Information Age', *ACM SIG CAS Computers and Society*, pp. 27 - 32. Retrieved: September 1997, from

Platform for Privacy Preferences (P3P) Project, (2004), W3C. Retrieved: 28/7/2005, from http://www.w3.org/P3P/.

Thibadeau, R, (2000), *A Critique of P3P: Privacy on the Web*. Retrieved: 30/08/2005, from http://dollar.ecom.cmu.edu/p3pcritique/.

Westin, AF, (1967), *Privacy and Freedom*, New York, Atheneum.