A Privacy Policy Framework for Service Aggregation with P3P

Liju Dong*†, Yi Mu*, Willy Susilo*, Peishun Wang*, Jun Yan‡
*Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
† School of Information Science and Engineering, Shenyang University, Shenyang 110044, P. R. China
‡ School of Information System and Technology, University of Wollongong, Wollongong, NSW 2522, Australia
Email: {liju,ymu,wsusilo,peishun,jyan}@uow.edu.au

***Abstract*—Service aggregation has exhibited useful features for efficient and reliable services, especially for the Internet. Recent advances of service aggregation pose a new challenge to privacy policy management due to the nature of policy aggregation and policy inconsistency. Previous studies in privacy policies do not capture privacy issues in service aggregation. In this paper, we present a formal result to demonstrate privacy policy aggregation. In particular, we show how to implement privacy policy aggregation with Platform for Privacy Preferences (P3P).**

Keywords: Privacy Policy, Service Aggregation, P3P.

I. Introduction

The advances of the Web technologies and service oriented architecture enable much better services to Web users in terms of the volume of services and the efficiency of services. It is a trend to combine various services from different providers in order to offer better and efficient services to customers. As an emerging technology, Service Aggregation has been regarded as a promising candidate for integrate services from multiple service providers [1], [2], [3]. Its benefit originates from the added value generated by the possible interactions and by the large scale rather than by the capabilities of its individual service provider separately. This technology has created tremendous opportunities to businesses. On the other hand, it also raises new security issues, as services are provided by service providers from distributed network domains [4]. These issues have not been addressed in the literature.

Privacy is always an important issue in web services [5], [3], [6]. Many organizations now publish their privacy policies on their online service web sites. In a single domain environment, a well-defined privacy policy can be formally presented with the well-known privacy policy languages such as P3P [7], [8] and XACML [9]. However, the situation is entirely different, while the service is provided through an aggregate server, where multiple service providers behind the aggregate server normally adopt different privacy policies and the aggregate service requires an aggregate privacy policy. The major difficulty to policy aggregation is due to inconsistency and conflict of the corresponding policies, where each server provides a part of the service. There exist several other useful tools for privacy policy management, such as APPEL [10], EPAL [11], [12], [13] and ASL [5]. These tools provide formal approaches for describing privacy policies, but they do not capture privacy policies in service aggregation.

There exist several privacy policy models for multiple privacy policies in the literature. As one of the most notable works, Backs *et al.* [11] proposed a formal model for composing enterprise privacy policies. The aim of the model is to provide the compliance with different privacy policies when several parts of an organization or different enterprises cooperate. This work is based a superset of the syntax and semantics of IBM's Enterprise Privacy Authorization Language (EPAL). They provided an elegant solution to handle conjunction and disjunction of privacy policies, which are not well defined in EPAL. We notice that this policy model does not accommodate our model where the conflicts in privacy policy aggregation possess a more complex nature, which cannot been captured with logical AND and OR defined in their model.

Backes et al. [11] proposed a formal model for comparison of enterprise privacy policies in P3P (E-P3P), based on EPAL. Although well-established in the theory, the problem addressed in their work is mainly about how to efficiently check whether one policy refines another. This privacy policy model does not capture all in privacy policy management, especially in conflict resolution in service aggregation. With other novel functionalities, several other privacy policy comparison models were introduced [14], [15], [5], [16], [17]. However, these methods do not address the privacy management for service aggregation either. In particular, they do not consider large privacy policy sets from multiple parties.

In this paper, we formally define the privacy policy aggregation and provide an instantiation with P3P to demonstrate how to implement aggregate privacy policies. In particular, we present the definitions including privacy policy aggregation, privacy policy aggregation with P3P, policy comparison for P3P, and concrete P3P examples. We also provide a solution to privacy policy conflict and constraint.

The remaining sections of this paper are organized as follows. In section II, we present an overview of service aggregation and provide a description about the service aggregation model we consider. In Section III, we preset the syntax and semantics of our privacy policy model. In Section IV, we provide the proposed implement of our framework to P3P. In Section V, we conclude this paper.

II. Overview of Service Aggregation

Service aggregation is associated with methods and tools that create composite services and their lifecycle management including alignment of privacy policies, security, transaction management, quality of service and other elements of service provision. In the Internet, the service providers are geographically distributed. A service aggregator manages the services

ISBN: 978-1-61208-124-3

requested by a client. The aggregated service could be one or multiple services in terms of the client's requirement. A general view of a service aggregation system is given in Figure 1.

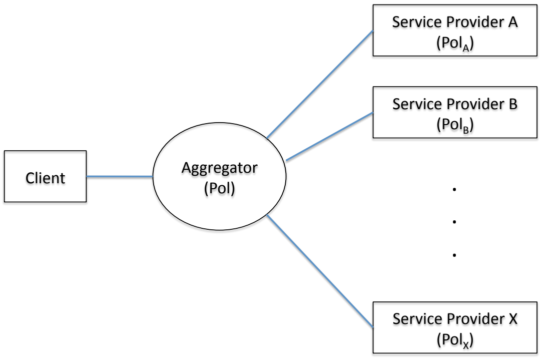

Figure 1. Selected services are aggregated by the Aggregator. Accordingly the related privacy policies $\{\mathrm{Pol}_i\}$ are aggregated into Pol.

In general, the service aggregator is the only entity that is visible to the client. To obtain a suitable service, a client needs to register with the aggregator. For getting a service, a client only needs to contact the aggregator, who in turn contacts the related service providers located at the backends. The merit of such type of services is that clients do not need to hunt required services, while they only need to contact the aggregator; therefore, it simplifies the process.

Because of the distributed nature, security and privacy are important issues in service aggregation. Nowadays, Web services are required to publish their privacy policies. In service aggregation, the final service might consist of multiple services whose privacy policies are different. As depicted in Figure 1, only privacy policy that is visible to the client is an aggregated privacy policy from a multiple policies of involved service providers. The aggregated policy must reflect all the policy rulesets from the service providers. It is desirable that the aggregated policy is dynamically formed, as it depends on the related services selected by the service aggregator. To ensure efficiency and accuracy in forming an aggregated policy, in this paper, we develop a privacy aggregation framework along with associated policy aggregate algorithms.

III. Syntax and Semantics

In this section, we formally define privacy policy aggregation, with the consideration of the P3P instantiation. The objective of this section is to provide the reader with clear definitions that illustrate the P3P instantiations presented in the next section. Those definitions are presented with a simple formal language so that it can be easily understood by practitioners.

A. Syntax of Privacy Policies

A privacy policy is a tuple of a vocabulary, a set of authorization rules, and a default ruling. The vocabulary defines subjects, objects, operations, and purposes. Subjects are a set of users. Objects are data. Our model also includes Purpose, which is an important element for privacy policies.

Definition 1: (Vocabulary) A vocabulary is a tuple $\mathcal{V}$ = (S, O, OP, P) where S, O, OP, and P are elements called subject, object, operation, and purpose.

To define the privacy policies, we require that all components also get a subscript i and an authorization ruling $\mathcal{A} = \{+, -, *\}$, where $\{+, -, *\}$ denote "allow", "do not allow", and "do not care" or from user perspective, "require privacy information", "do not require privacy information", and "do not specify".

Definition 2: (Ruleset) A ruleset for a vocabular $\mathcal{V}$ is a subset of $\mathcal{I} \times S \times O \times OP \times P \times \mathcal{A}$, where $\mathcal{I}$ denotes an index set. An instance of a ruleset is denoted by a complete set $rs = (i, s, o, op, p, a)$ or a subset of (i, s, o, op, p, a).

Definition 3: (Privacy Policy) A privacy policy Pol is a triple $(\mathcal{V}, \mathcal{R}, \mathcal{A})$ of a vocabulary $\mathcal{V}$, a ruleset $\mathcal{R}$, and a related ruling $\mathcal{A}$.

We assume that the components of a privacy policy Pol are always called as in Definition 3. For simplicity, we denote by $Pol = (r_1, r_2, \cdots, r_n)$ a privacy policy that consists of a tuple of n rules, where r_i is an instance of $(\mathcal{V}, \mathcal{R}, \mathcal{A})$.

B. Semantics of Privacy Policy Aggregation

Definition 4: (Policy Aggregation) Let $Pol_k = (r_{k,1}, r_{k,2}, \cdots, r_{k,n_k})$ for $k = 1, 2, \cdots, m$ be m privacy policy sets, where $r_{k,i}$ denotes the corresponding rule. The aggregate privacy policy is defined as

$$Pol(R_I) \leftarrow \bigoplus_{k=1}^{m} Pol_k(r_{k,i}) \tag{1}$$

where by $Pol(r)$ we denote that rule r belongs to policy Pol, by $r_{k,i}$ we denote a rule that is indexed by i and belongs to policy k, by R_I we denote the resulting rules and by $\oplus$ we denote a generic aggregate operator representing Union, Intersect, Minus, or Conflict, depending upon the properties of the policy rules. '$\leftarrow$' denotes an assignment, where the policies on the right are aggregated to yield the policy on the left.
As an example of two policy sets, we have

$$Pol(R) \leftarrow Pol_1(a) \oplus Pol_2(b)$$

where a, b are two rules and R is the aggregate rule set, which could be a, b, (a, b), or an empty set. The "rule" is a generic term that can be an element, a statement, or a constraint in P3P.

Definition 5: (related) Two rules a and b are said related (or $a \sim b$), if they are associated with the same privacy property. If two rules are non-related (or $a \nsim b$), then the aggregate rule is an union of them; that is,

$$Pol(a, b) \leftarrow Pol_1(a) \oplus Pol_2(b).$$

As an example, if both a and b are associated with gender, they are regarded as "related" Otherwise, if a is about gender and b is about driving license, then they are "non-related".

Definition 6: (privacy-expose) A rule is said privacy-expose, if the rule queries a piece of private information; otherwise, the rule is non-privacy-expose. A privacy-expose rule overrides an non-privacy-expose rule if they are in conflict and related.

For example, a_i is a policy item requesting a piece of personal information (privacy-expose) and b_j does not require it. In the view of the customer, he only cares about his privacy before he commits to the aggregate service.

The rules in a privacy policy are subject to the following operations. '$=$' is an operator indicating that the left and the right are equivalent. '$>$' is an operator indicating the left overrides the right. '$<>$' is an operator indicating that the left is in conflict with the right. They are captured by the following three definitions.

Definition 7: (equivalent) If two policy rules a and b represent the same privacy policy, then they are equivalent; that is $a = b$.

Definition 8: (overrides) If a policy rule a dominates another rule b, a overrides b (or $a > b$).
It is dependent on the condition that a rule is allowed to override the other. For example, a privacy-expose rule can override a non-privacy-expose rule. To deal with the conflict of policies, we introduce the conflict resolution definitions.

Definition 9: (in conflict) Given two rules a and b, if neither rule overrides the other, then they are in conflict or $a <> b$.

In the following, we demonstrate how to implement aggregation by using some typical examples.

Example 1. If $a_i \in Pol_1$ and $b_j \in Pol_2$ are not equivalent $(a_i \neq b_j)$ and are unrelated $(a_i \not\sim b_j)$, then we have

$$Pol(a_i, b_j) \leftarrow Pol_1(a_i) \oplus Pol_2(b_j).$$

In this instance, $Pol \leftarrow Pol_1 \cup Pol_2$. Here, we have omitted other rules in the policy sets for simplicity.

Example 2. If $a_i \in Pol_1$ and $b_j \in Pol_2$ are equivalent, then $a_i = b_j$. We have

$$Pol(a_i) \leftarrow Pol_1(a_i) \oplus Pol_2(b_j)$$

$$\text{or} \quad Pol(b_j) \leftarrow Pol_1(a_i) \oplus Pol_2(b_j).$$

In this instance, $Pol \leftarrow Pol_1 \setminus Pol_2$ or $Pol \leftarrow Pol_2 \setminus Pol_1$, where "$\setminus$" denotes "exclude".

Example 3. If $a_i \in Pol_1$ is a privacy-expose rule and $b_j \in Pol_2$ is a non-privacy-expose rule, with $a_i \sim b_j$, then $a_i > b_j$. We have

$$Pol(a_i) \leftarrow Pol_1(a_i) \oplus Pol_2(b_j).$$

In this instance, $Pol \leftarrow Pol_1 \setminus Pol_2$

Example 4. $a_i \in Pol_1$ and $b_j \in Pol_2$, with $a_i \sim b_j$. $a_i <> b_j$, if there is no any conflict resolution. We have

$$Pol() \leftarrow Pol_1(a_i) \oplus Pol_2(b_j).$$

In this instance, $Pol \leftarrow Pol_1 \cap Pol_2$.

Definition 10: (Semantics of Policy Aggregation). Given privacy policy Pol_k, the evaluation result of Pol_k is defined by the following algorithm:

1. Select as input two policy rules a_i and b_j from two policy sets: $Pol_1(a_1, \cdots, a_{n_1})$ and $Pol_2(b_1, \cdots, b_{n_2})$.
2. Compare the selected rules in terms of the definitions 5-9 and output the aggregate policy.
3. Repeat the process till the all rules in the target policy sets are checked.

The aggregation algorithm is referred to as Algorithm 1 described in Figure 2 in detail.

```
input $n_1, n_2$
while $i < n_1$ and $j < n_2$, do
    input: $Pol_1(a_i), Pol_2(b_j)$
        if $a_i = b_j$ then
            return $Pol(a_i)$ or $Pol(b_j)$;
        end if
        if $a_i \not\sim b_j$ then
            return $Pol(a_i, b_j)$;
        else
            if $a_i <> b_j$ then
                return $Pol()$;
            else
                if $a_i > b_j$ then
                    return $Pol(a_i)$;
                else
                    return $Pol(b_j)$;
                end if
            end if
        end if
end
```

Figure 2. Algorithm 1: an algorithm of privacy policy aggregation.

IV. P3P Policy Implementation in SA

A. P3P Deployment

A common way to express privacy principles are privacy policies expressed in formal implementable languages, such as P3P (the Platform for Privacy Preferences) [18], XACML [19] and some other languages. P3P is the most popular policy language since 2006, which is an industry-supported self-regulation approach to privacy protection [20]. It is a W3C recommendation as a protocol to communicate how a service intends to collect, use, and share personal information about its visitors [21], [22]. The current development status of P3P is the Working Group Note of the P3P 1.1 Specification, published in November 2006. P3P is an industry standard for privacy protection, designed to give users more control over their personal information when visiting services. We choose P3P as an example for the privacy policy implementation in SA.

A policy set $Pol(a_1, \cdots, a_n)$ is represented by a P3P statement. A rule in a P3P policy set can be represented by an element or a constraint in P3P. According to this definition, a P3P policy may consist of several policy sets:

$$\{Pol_1(a_1, \cdots, a_{n_1}), Pol_2(b_1, \cdots, b_{n_2}), \cdots,$$

$$Pol_m(x_1, \cdots, x_{n_m})\}.$$

To clarify our definition, we consider the following example. There are several elements in one statement, such as Purpose, Retention, Recipient, Data and other options. For example, Pol_i is a statement as below:

```
<STATEMENT>
    <PURPOSE><current/><develop/></PURPOSE>
    <RECIPIENT><ours/><delivery/></RECIPIENT>
    <RETENTION><indefinitely/></RETENTION>
    <DATA-GROUP>
       <DATA ref="#thirdparty.name"/>
       <DATA ref="#thirdparty.home-info"/>
       <DATA ref="#thirdparty.business-info"/>
    </DATA-GROUP>
</STATEMENT>
```

The statement is referred to as a policy set:

$$Pol_i(\texttt{PURPOSE,RECIPIENT,RETENTION,}\\ \texttt{DATA-GROUP}).$$

Obviously, Pol_i does not reflect the entire policy, as there are multiple layers in P3P. In the following section, we present a solution by considering the entire P3P setting.

B. P3P Implementation

To implement P3P policies, we classify a P3P statement into levels in terms of depth.

Definition 11: (Element Set (ES)) An Element Set consists of P3P elements, which can be normal elements and optional elements. Optional elements have an optional value 0, 1, 2, or 3, which denote none, always (as default), opt-in, and opt-out, respectively. These elements are arranged with levels (ESL): 0, 1, ..., n, in terms of depth, where 0 is the root and n is the last ESL or leaves.

Taking the above P3P statement as an example, we have

- ESL = 0: <STATEMENT>.
- ESL = 1: <PURPOSE>.
  ESL = 2: <current/><develop/>.
- ESL = 1: <RECIPIENT>.
  ESL = 2: <ours/><delivery/>.
- ESL = 1: <RETENTION>.
  ESL = 2: <indefinitely/>.
- ESL = 1: <DATA-GROUP>.
  ESL = 2: <DATA ref="#thirdparty.name"/>,
  <DATA ref="#thirdparty.home-info"/>,
  <DATA ref="#thirdparty.business-info"/>.

In the last P3P example, the statement has a depth of 2. The root level (ESL = 0) is <STATEMENT>. The first level (ESL = 1) contains a set of default tags such as <PURPOSE>, <RETENTION>, <DATA-GROUP>, etc. The second level (ESL = 2) contains a number of elements depending on their parent. As shown in Table I, <PURPOSE> includes a set of children, which could be either optional or non-optional.

The aggregate Algorithm 2 along with Algorithm 3 and Algorithm 4, as shown in Figure 3, can be utilized to achieve an aggregate privacy policy. It illustrates how two sets of privacy policies can be aggregated. The algorithm can be extended to more policy sets, when the input is altered.

C. P3P Example

To illustrate our privacy policy aggregation algorithms, we provide a concrete P3P example. Assume that Pol_1(statement) and Pol_2(statement) are privacy policies of two online bookshops, respectively.

Table I
THE THIRD COLUMN LISTS THE P3P ELEMENTS OF <PURPOSE>, WHICH CAN BE EITHER OPTIONAL OR NON-OPTIONAL WITH OPTIONAL VALUES GIVEN IN THE FIRST COLUMN. AS AN EXAMPLE, <CONTACT/> IN THE SECOND COLUMN IS USED TO COMPARE WITH THE ELEMENTS LISTED IN THE THIRD COLUMN. AS A RESULT, THEY COULD BE EITHER RELATED OR NON-RELATED, AS DEFINED IN SECTION 2.

	ES (ESL = 2)	ES (ESL = 2)	Related
Optional	<contact/>	<current/>	no
Value	<contact/>	<admin/>	no
(Ovalue):	<contact/>	<develop/>	no
	<contact/>	<tailoring/>	no
0 none	<contact/>	<pseudo-analysis/>	no
1 always	<contact/>	<pseudo-decision/>	no
2 opt-in	<contact/>	<individual-analysis/>	no
3 opt-out	<contact/>	<individual-decision/>	no
	<contact/>	<contact/>	yes
	<contact/>	<historical/>	no
	<contact/>	<telemarketing/>	no
	<contact/>	<other-purpose>	no

Pol_1(statement) says that the name, postal address, and miscellaneous online data are used for completing the current data transaction. P3P policies collect personal information only for the current service. Considering the attributes of the purpose opt-in and opt-out values, we can simplify them as <contact required=opt-out/>. Pol_1(statement) also uses users' data history and offers personalized book recommendations by categories <preference/>.

Pol_1(statement) = Pol_1(purpose, recipient, retention, data-group) represents the following P3P privacy policy.

```
<STATEMENT>
    <PURPOSE>
         <current/><admin/>
         <contact required="opt-out">
     </PURPOSE>
    <RECIPIENT><ours/></RECIPIENT>
    <RETENTION><stated-purpose/></RETENTION>
    <DATA-GROUP>
         <DATA ref="#user.name"/>
         <DATA ref="#user.home-info.postal"/>
         <DATA ref="#dynamic.miscdata">
              <CATEGORIES><online/>
              </CATEGORIES>
         </DATA>
         <DATA ref="#dynamic.miscdata">
              <CATEGORIES><preference/>
              </CATEGORIES>
         </DATA>
     </DATA-GROUP>
</STATEMENT>
```

Pol_2(statement) = Pol_2(purpose, recipient, retention, data-group), as given below, says that Pol_2(statement) requires to use the miscellaneous purchase data to create personal recommendations, where the user name and miscellaneous purchase data will be used for the current purchase transaction.

```
<STATEMENT>
     <PURPOSE>
         <current/><admin/>
         <contact required ="opt-in"/ >
     </PURPOSE>
     <RECIPIENT><ours/></RECIPIENT>
```

```
Algorithm 2: for P3P privacy policy aggregation.
  Input $Pol_1$ and $Pol_2$.
  1. Start iteration.
  2. Search a pair of matching statements of $Pol_1$ and $Pol_2$. If a match is
     found, set ESL = 0 and go to the next step; otherwise $Pol \leftarrow Pol_1 \cup Pol_2$
     and exit.
  3. Search elements of $Pol_1$ and $Pol_2$ with $ESL = 1$. If a match is found,
     go to the next step, else $Pol(ES_1 \cup ES_2) \leftarrow Pol_1(ES_1) \oplus Pol_2(ES_2)$
     and go to step 2.
  4. Search the elements of $ESL = 2$ under the current parent. If an element
     is found, call Algorithm 4.
     Go to step 2 until all elements are reached.
```

```
Algorithm 3: for optional elements.
  Input a pair of optional elements $(a_i, b_j)$.
  1. Retrive Ovalues of $a_i$ and $b_j$.
  2. Call Algorithm 1 with the Ovalues, $a_i$ and $b_j$ as input. (note: the overriding rule
     is based on the Ovalues.
  3. Return.
```

```
Algorithm 4: for recursive calls.
  1. If they are not empty and have no child, get an element from $Pol_1$ and an
     element from $Pol_2$ and call Algorithm 1.
  2. If a pair of optional elements is encountered, call Algorithm 3. Loop over all
     elements under the current parent tag and go to step 2.
  3. If a child is found, go to step 1.
  4. Return.
```

Figure 3. P3P aggregation algorithms.

```
      <RETENTION><indefinitely/></RETENTION>
      <DATA-GROUP>
         <DATA ref="#user.name"/>
         <DATA ref="#dynamic.miscdata"
                           optional="yes">
             <CATEGORIES><content/>
             </CATEGORIES>
         </DATA>
         <DATA ref="#dynamic.miscdata"
                           optional="yes">
             <CATEGORIES><purchase/>
             </CATEGORIES>
         </DATA>
       </DATA-GROUP>
</STATEMENT>
```

Following Algorithm 2, the <STATEMENT> is set as the root or ESL = 0. The elements for ESL = 1 are found:

$ES_1(ESL = 1)$ =
(purpose,recipient,retention,data-group).
$ES_2(ESL = 1)$ =
(purpose,recipient,retention,data-group).

These elements are then reached one by one. As the first element in <PURPOSE> for both statements, the children are checked and matching elements are compared with Algorithm 1. As a result, <current/>,<admin/> in both statements are equivalent and therefore they stay. The third element <contact/> is optional, hence Algorithm 3 is invoked to resolve it and the output is <contact required ="opt-out"/> as Ovalue for opt-out is 3, which is greater than the Ovalue of opt-in (Ovalue = 2). After all children of <PURPOSE> are reached, the second element <RECIPIENT> at ESL = 1 is checked. for ESL greater than 2, we invoke Algorithm 3 for recursive calls. This will continue until the last element is reached. Consequently, we obtain the aggregated privacy policy:

$Pol(statement)$ = Pol(purpose, recipient, retention, data-group)

V. Conclusion

We introduced a new notion of privacy policy aggregation for P3P, which has not been previously explored. We presented a framework for handling P3P privacy policies for service aggregation, which is seen as an emerging technology for providing efficiency and quality of web services. We formally defined the syntax and semantics of of our privacy policy aggregation language and provided algorithms based on the formal definitions of privacy policy aggregation. We presented an P3P example to demonstrate how our scheme works. We found that our framework captures all necessary needs for privacy policy aggregation.

References

[1] R. Kanneganti and P. Chodavarapu, "SOA security," in *Proceedings of SACMAT'07*. Manning Publications Co. Greenwich, CT, 2008.

[2] R. R. Khalaf and F. Leymann, "On web services aggregation," in *Technologies for E-Services 2003, LNCS*. Springer, Heidelberg, 2003, pp. 1–13.

```
<STATEMENT>
    <PURPOSE>
        <current/>  <!-- ai = bj -->
        <admin/>    <!-- ai = bj -->
        <contact required ="opt-out"/>
        <!-- ai(<contact required="opt-out">) > bj("opt-in") -->
    </PURPOSE>
    <RECIPIENT>
        <ours/>      <!-- ai = bj -->
        <delivery/>  <!-- ai = bj -->
    </RECIPIENT>
    <RETENTION>
        <indefinitely/>
        <!-- bj(<indefinitely/>) > ai(<stated-purpose/>) -->
    </RETENTION>
    <DATA-GROUP>
         <DATA ref="#user.name"/> <!-- ai = bj -->
         <DATA ref="#user.home-info.postal"/>
         <!-- aj(<"#user.home-info.postal">)>bi(empty) -->
         <DATA ref=#dynamic.miscdata optional="yes">
         <!-- bj(optional="yes" > ai(empty) -->
              <CATEGORIES>
              <content/>
              <!--  bj(<content/>) > ai(<online/>) -->
         </CATEGORIES>
         </DATA>
         <DATA ref="#dynamic.miscdata">
                <CATEGORIES>
                    <purchase/>
                    <!--  bj(<purchase/>) > ai(<empty/>) -->
                </CATEGORIES>
         </DATA>
         <DATA ref="#dynamic.miscdata">
             <CATEGORIES>
                  <preference/>
                   <!-- ai(<preference/>) > bj(<empty/>) -->
             </CATEGORIES>
         </DATA>
    </DATA-GROUP>
</STATEMENT>
```

Figure 4. Example of the aggregated policy.

[3] A. I. Anton, E. Bertino, N. Li, and T. Yu, "A roadmap for comprehensive online privacy policy management," *Communications of the ACM*, vol. 50, pp. 109–116, 2007.

[4] E. C. Lupu and M. Sloman, "Conflicts in policy-based distributed systems management," *IEEE Transactions on Software Engineering*, vol. 25, pp. 852–869, 1999.

[5] G. Karjoth and M. Schunter, "A privacy policy model for enterprises," in *Proceedings of the 15th IEEE CSFW'02*. IEEE, 2002, pp. 271–274.

[6] B. Berendt, S. Preibusch, and M. Teltzrow, "A privacy-protecting business analytics service for online transactions," *International Journal of Electronic Commerce*, vol. 12, pp. 109–116, 2008.

[7] T. Yu, N. Li, and A. I. Anton, "A formal semantics for P3P," in *Proceedings of ACM Workshop on Secure Web Services*. ACM, 2004, pp. 1–8.

[8] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "An xpath-based preference language for p3p," in *Proceedings of WWW'03 2003*. ACM Press, 2003, pp. 629–639.

[9] P. Mazzoleni, E. Bertino, and B. Crispo, "XACML policy integration algorithms," in *Proceedings of SACMAT'06*, 2006, pp. 219–227.

[10] "A P3P Preference Exchange Language 1.0 (APPEL1.0)," `http://www.w3.org/TR/P3P-preferences/`.

[11] M. Backes, W. B. G. Karjoth, and M. Schunter, "Efficient comparison of enterprise privacy policies," in *Proceedings of SAC'04*. ACM Press, 2004, pp. 375–382.

[12] A. H. Anderson, "A comparison of two privacy policy languages: EPAL and XACML," in *Proceedings of SWS'06*, 2006, pp. 53–60.

[13] Y. H. Li, H.-Y. Paik, and B. Benatallah, "Formal consistency verification between BPEL process and privacy policy," in *Proceedings of PST 2006*, 2006, pp. 1–10.

[14] M. Backers, M. Durmith, and R. Steinwandt, "An algebra for composing enterprise privacy policies," in *Proceedings of ESORICS 2004*. LNCS 3193, 2004, pp. 33–52.

[15] S. Gevers and B. D. Decker, "Privacy friendly information disclosure," in *Proceedings of OTM Workshops 2006*. LNCS 4277, 2006, pp. 636–646.

[16] P. Bodorik, D. Jutla, and M. X. Wang, "Consistent Privacy Preferences (CPP): Model, semantics, and properties," in *Proceedings of SAC'08*, 2008, pp. 2368–2375.

[17] D. Lin, P. Rao, and E. Bertino, "An approach to evaluate policy similarity," in *Proceedings of SACMAT'07s*, 2007, pp. 1–10.

[18] "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation (April 2002)," http://www.w3.org/TR/P3P/.

[19] "OASIS. security services technical committee. extendible access control markup language (XACML) version 2.0," 2006, http://docs.oasis-open.org/xacml/xacmlrefs.html.

[20] I.V.Ramakrishnan and R. S. W. Xu, "On supporting active user feedback in P3P," in *Proceedings of 2nd Workshop on Secure Knowledge Management (SKM '06)*, 2008, pp. 1–6.

[21] H. Hochheiser, "The platform for privacy preference as a social protocol: An examination within the U.S. policy context," *ACM Transactions on Internet Technology*, vol. 2, pp. 276–306, 2002.

[22] L. F. Cranor, *Web Privacy with P3P*. O'Reilly & Associate Inc, 2002.

ISBN: 978-1-61208-124-3