A Survey Report on Privacy-Enhanced Web Service Composition
SUGUNA RAJU MEESALA1, J. RAJAKALA2
1PG Scholar, Dept of CSE, Akula Sriramulu Engineering College, AP, India.
2Associate Professor, Dept of CSE, Akula Sriramulu Engineering College, AP, India.

Abstract: Data as a Service (DaaS) builds on service-oriented technologies to enable fast access to data resources on the Web. However, this paradigm raises several new privacy concerns that traditional privacy models do not handle. In addition, DaaS composition may reveal privacy-sensitive information. In this paper, we propose a formal privacy model in order to extend DaaS descriptions with privacy capabilities. The privacy model allows a service to define a privacy policy and a set of privacy requirements. We also propose a privacy-preserving DaaS composition approach allowing to verify the compatibility between privacy requirements and policies in DaaS composition. We propose a negotiation mechanism that makes it possible to dynamically reconcile the privacy capabilities of services when incompatibilities arise in a composition. We validate the applicability of our proposal through a prototype implementation and a set of experiments.

Keywords: Service Composition, DaaS Services, Privacy, Negotiation.

I. INTRODUCTION
Web services have recently emerged as a popular medium for data publishing and sharing on the Web [18]. Modern enterprises across all spectra are moving towards a service-oriented architecture by putting their databases behind Web services, thereby providing a well-documented, platform independent and interoperable method of interacting with their data. This new type of services is known as DaaS (Data-as-a-Service) services [33] where services correspond to calls over the data sources. DaaS sits between services-based applications (i.e. SOA-based business process) and an enter prise’s heterogeneous data sources. They shield applications developers from having to directly interact with the various data sources that give access to business objects, thus enabling them to focus on the business logic only. While individual services may provide interesting information/functionality alone, in most cases, users’ queries require the combination of several Web services through service composition. In spite of the large body of research devoted to service composition over the last years [24], service composition remains a challenging task in particular regarding privacy. In a nutshell, privacy is the right of an entity to determine when, how, and to what extent it will release private information [16]. Privacy relates to numerous domains of life and has raised particular concerns in the medical field, where personal data, increasingly being released for research, can be or have been, subject to several abuses, compromising the privacy of individuals [3].

TABLE I: Subset of DaaS Services

<table>
<thead>
<tr>
<th>DaaS services</th>
<th>Semantics services Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1.1(x1, x2)</td>
<td>Returns patients SNN, s, infected with a disease x</td>
</tr>
<tr>
<td>S2.1(x3, x4)</td>
<td>Returns date of birth, d, and gender, g, of a patient identified by s</td>
</tr>
<tr>
<td>S3.1(x5, ..., x7)</td>
<td>Returns zip. code, z, and salary, r, of a patient identified by s</td>
</tr>
<tr>
<td>S4.1(x8, x9)</td>
<td>Returns DNA, n, of a patient identified by s</td>
</tr>
<tr>
<td>S5.1(x10, x11)</td>
<td>Returns Weather condition, p, of a address z</td>
</tr>
</tbody>
</table>

A. e-Epidemiological Scenario
Let us consider the services in Table 1 and the following epidemiologist’s query Q “What are the ages, genders, address, DNA, salaries of patients infected with H1N1; and what are the global weather conditions of the area where these patients reside?”. We proposed in [2] a mediator-based approach to compose DaaS. The mediator selects, combines and orchestrates the DaaS services (i.e., gets input from one service and uses it to call another one) to answer received queries. It also carries out all the interactions between the composed services (i.e., relays exchanged data among interconnected services in the composition). The result of the composition process is a composition plan which consists of DaaS that must be executed in a particular order depending on their access patterns (i.e., the ordering of their input and output parameters). Thus, Q can be answered by composing the following services S1.1•S4.1•S2.2•S3.1•S5.1. It means that S1.1 firstly is invoked with H1N1, then for each obtained patient, S4.1 is invoked to obtain their DNA, S2.2 and S3.1 to obtain date–of–birth, zip–code and salary of obtained patients. Finally, S5.1 is invoked with the patients’ zip–code to get information about the weather–conditions.

B. Challenges
Two factors exacerbate the problem of privacy in DaaS. First, DaaS services collect and store a large amount of private information about users. Second, DaaS services are able to share this information with other entities. Besides, the emergence of analysis tools makes it easier to analyze and synthesize huge volumes of information, hence increasing...
the risk of privacy violation [21]. In the following, we use our epidemiological scenario to illustrate the privacy challenges during service composition.

Challenge 1: Privacy Specification: Let us consider services S4.1 and S5.1 in Table 1. The scientist con-siders both input and output parameters of S4.1 (i.e., SSN and DNA) as sensitive data. Let us now assume that this scientist states the following hypothesis: “weather−conditions” has an impact on H1N1 infection.” For that purpose, he/she invokes S5.1. The scientist may want to keep S5.1 invocation as private (independently of what S5.1 takes and returns as data) since this may disclose sensitive information to competitors. The aforementioned first challenge puts in evidence the need for a formal model to specify private data and how it will be defined.

Challenge 2: Privacy Within Compositions: Component services (that participate in a composition) may require input data that cannot be disclosed by other services because of privacy concerns. They may also have conflicting privacy concerns regarding their exchanged data. For instance, let us assume that S1.1 states to disclose its data (SSN) to a third-party service for use in limited time. S3.1 meanwhile attests that it uses collected data (SSN) for an unlimited time use. Then, S1.1 and S3.1 have different privacy constraints regarding the SSN. This will invalidate the composition in terms of privacy concerns.

Challenge 3: Dealing with Incompatible Privacy Policies in Compositions: The role of the mediator is to return composite services with compatible component services with respect to privacy. The simplest way to deal with compositions with incompatible privacy policies is to reject those composition. However, a more interesting and challenging approach would be to try to reach a consensus among component services to solve their privacy incompatibilities, hence increasing the number of composition plans returned by the mediator.

C. Contributions Privacy Model

We describe a formal privacy model for Web Services that goes beyond traditional data-oriented models. It deals with privacy not only at the data level (i.e., inputs and outputs) but also service level (i.e., service invocation). In this paper, we build upon this model two other extensions to address privacy issues during DaaS composition. The privacy model described in this paper is based on the model initially proposed in [10] and [18].

Privacy-aware Service Composition: We propose a compatibility matching algorithm to check privacy compatibility between component services within a composition. The compatibility matching is based on the notion of privacy subsumption and on a cost model. A matching threshold is set up by services to cater for partial and total privacy compatibility.

Negotiating Privacy in Service Composition: In the case when any composition plan will be incompatible in terms of privacy, we introduce a novel approach based on negotiation to reach compatibility of concerned services (i.e., services that participate in a composition which are incompatible). We aim at avoiding the empty set response for user queries by allowing a service to adapt its privacy policy without any damaging impact on privacy. Negotiation strategies are specified via state diagrams and negotiation protocol is proposed to reach compatible policy for composition.

D. Paper Organization

The rest of this paper is organized as follows: In Section 2 we review the composition approach proposed in [2] as part of the PAIRSE project. We present our privacy model in Section 3. We introduce the notion of compatibility between privacy policies and requirements in Section 4. In Section 5 we show how our DaaS composition approach is extended within privacy-preserving mechanism. We present our negotiation model in Section 6 to deal with the issue of privacy incompatibility. In Section 7 we describe our prototype implementation and evaluate the performance of the proposed approach.

II. THE PAIRSE PROJECT: BACKGROUND

The approach presented in this paper is implemented as part of the PAIRSE project which deals with the privacy preservation issue in P2P data sharing environments, particularly in epidemiological research where the need of data sharing is apparent for making better a health environment of people. To support the decision process, epidemiological researchers should consider multiple data sources such as the patient data, his social conditions, the geographical factors, etc.

![Fig. 1. PAIRSE global architecture.](image-url)
A Survey Report on Privacy-Enhanced Web Service Composition

The data sources are provided by DaaS services and are organized with peers. DaaS services differ from traditional Web services, in that they are stateless; i.e. they only provide information about the current state of the world but do not change that state. When such a service is executed, it accepts from a user an input data of a specified format (“typed data”) and returns back to the user some information as an output. DaaS services are modeled by RDF views. Fig.1 summarizes the architecture of this project. The Multi-Peer Query Processing component is in charge being stolen by a competing company. We give below the definition of the privacy level. Privacy level on resource rsS is defined as follows: (i) L=“data” if rs is an input/output of S of answering the global user query. The latter has to be split local queries (i.e., sub-queries) and has to determine which peer is able to solve a local query. Each sub-query is expressed in SPARQL. Each peer handles a Mediator equipped with a Local Query Processing Engine component. The mediator exploits the defined RDF views within WSDL files to select the services that can be combined to answer the local query using an RDF a query rewriting algorithm [2]. Then, it carries out all the interactions between the composed services and generates a set of composition plans to provide the requested data.

III. PRIVACY MODEL

In this section, we describe our privacy model for DaaS services. Each service S has a privacy policy (noted as PPS) specifying the set of privacy practices applicable on any collected data and privacy requirements (noted as PRS) specifying the set of privacy conditions that a third-party service T must meet to consume S’s data. A preliminary version of the model described in this section was proposed in [8].

A. Privacy Level

We define two privacy levels: data and operation. The data level deals with data privacy. Resources refer to input and output parameters of a service (e.g., defined in WSDL). The operation level copes with the privacy about operation’s invocation. Information about operation invocation may be perceived as private independently on whether their input/output parameters are confidential or not [10]. For instance, let us consider a scientist that has found an invention about the causes of some infectious diseases, he invokes a service operation to search if such an invention is new before he files for a patent. When conducting the query, the scientist may want to keep the invocation of this operation private, perhaps to avoid part of his idea.

B. Privacy Rule

The sensitivity of a resource may be defined according to several dimensions called privacy rules. We call the set of privacy rules RulesSet(RS). We define a privacy rule by a topic, domain, level and scope. The topic gives the privacy facet represented by the rule and may include for instance: the resource recipient, the purpose and the resource retention time. The “purpose” topic states the intent for which a resource collected by a service will be used; the “recipient” topic specifies to whom the collected resource can be revealed. The level represents the privacy level on which the rule is applicable. The domain of a rule depends on its level. Indeed, each rule has one single level: “data” or “operation”. The domain is a finite set that enumerates the possible values that can be taken by resources according to the rule’s topic. For instance, a subset of domain for a rule dealing with the right topic is {“no-retention”, “limited-use”}. The scope of a rule defines the granularity of the resource that is subject to privacy constraints. Two rules at most are created for each topic: one for data and another for operations.

Example 1. We give two examples of rules R1 and R3 in RS, where: R1={Ti, L1, Di, Se1} where Ti=“recipient”, Di=“public, research-lab, government, hospital, university” and L1=“data” and Se1=“total”. R3={T3, L3, Di, Sc3} where T3=“retention”, D3={[0,1,...,Unlimited]} (defining retention in day), L3=“data” and Sc3=“total”. ∅

C. Privacy Annotation for WSDL-based DaaS

In our previous work detailed in [22], we have defined a mechanism to annotate WSDL 2.0 descriptions under the interface element that describes the abstract part of the service with privacy specification of service. We choose to annotate WSDL descriptions at the three following places: interface, operation, input and output. Furthermore, we note that services are located in Peer-to Peer environment which is controlled and managed by a super-peer. A service S wanting to adhere to this environment, has to undertake to respect its PR and PP by the signing of an e-contract with the responsible peer.

IV. THE PRIVACY COMPATIBILITY CHECKING

In this section, we introduce the notion of compatibility between privacy policies and requirements. Then, we define the notion of privacy subsumption and present our cost model-based privacy matching mechanism.

A. Privacy Subsumption

Let us consider a rule Ri={Ti, Li, Di, Si}. Defining an assertion A(Ri, rs)=<pf, g> for rs involving assigning value(s) from Di to the propositional formula pf of A. The values in Di are related to each other. For instance, let us consider the domain {public, government, federal tax, research} for a rule dealing with topic Ti=“recipient”. The value public is more
privacy matching degree and threshold. Due to the large number and heterogeneity of Daas services, it is not always possible to find policy PPS’ that fully matches a S’s requirement PRS. The privacy matching degree gives an estimate about the ratio of PRS assertions that match PPS’ assertions. We refer to M⊂ PRS as the set of all such PRS assertions. The degree is obtained by adding the weights of all assertions in M : Degree(PRS, PPS’) = wj for all assertions (Aj(Ri, rsk), wj, Mj) ∈ M. The privacy matching threshold τ gives the minimum value allowed for a matching degree. The value of τ is given by the service and gives an estimate of how much the service is willing to sacrifice.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a dynamic privacy model for Web services. The model deals with privacy at the data and operation levels. We also proposed a negotiation approach to tackle the incompatibilities between privacy policies and requirements. Although privacy cannot be carelessly negotiated as typical data, it is still possible to negotiate a part of privacy policy for specific pur-poses. In any case, privacy policies always reflect the usage of private data as specified or agreed upon by service providers. As a future work, we aim at designing techniques for protecting the composition results from privacy attacks before the final result is returned by the mediator.

Algorithm I: PFM

input : $\mathbb{P}^{F}$=((Aj(Ri, rsk)), $j \leq |\mathbb{P}^{F}|$, $i \leq |\mathbb{S}|$, $k \leq |\mathbb{F}|$, $rsk \in \mathbb{P}$, $Ri \in \mathbb{S}$ (assertion of privacy requirements)

output : $\mathbb{D} 
\mathbb{C}$ (The set of incompatible assertion couple);

1. foreach $rsk = rsk_i$ do
2. for $i = 1, i \leq |\mathbb{S}|$ do
3. for $j = 1, j \leq |\mathbb{P}^{F}|$ do
4. for $j' = 1, j' \leq |\mathbb{P}^{F}|$ do
5. if $(Aj(Ri, rsk_i) \subseteq (Aj(Ri, rsk_i))$ then
6. $Aj(Ri, rsk_i)$ is compatible with $Aj(Ri, rsk_i)$
7. else $\mathbb{D} \leftarrow (Aj(Ri, rsk_i), Aj(Ri, rsk_i))$

VIII. REFERENCES

A Survey Report on Privacy-Enhanced Web Service Composition


