An Active Storage Framework Based On Storage Security in Cloud Computing

Mr.K.Krthick¹ Dr. M. Kamarajan²
¹M.E (Final Year), ²M. E., Ph.D.,
¹²RatnaVel Subramaniam College of Engineering & Technology

Abstract— 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. A privacy-preserving DaaS composition approach allowing verifying the compatibility between privacy requirements and policies in DaaS composition. 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. In this paper, we aim at designing techniques for protecting the composition results from privacy attacks before the final result is returned by the mediator. And also we extend the previous composition approach to deal with the privacy-preserving issue within composition.

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. Modern enterprises across all spectra are moving towards a service-oriented architecture by putting their databases behind Web services, thereby providing a well-documented, and platform independent and interoperable method of interacting with their data. This new type of services is known as DaaS (Data-as-a-Service) services where correspond to calls over the data sources. DaaS sits between services-based applications (i.e. SOA-based business process) and an enterprise’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, 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. 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.

We aim at designing techniques for protecting the composition results from privacy attacks before the final result is returned by the mediator. We proposed in 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

A. Challenges

1) Challenge 1

Privacy Specification. Let us consider services $S_4$ and $S_5$ in table 1. The scientist considers both input and output parameters of $S_4$ (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 $S_5$. The scientist may want to keep $S_5$ invocation as private (independently of what $S_5$ 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 is and how it will be defined.

2) 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 $S_1$ states to disclose its data (SSN) to third-party service for use in limited time. $S_3$ meanwhile attests that it uses collected data (SSN) for an unlimited time use. Then, $S_1$ and $S_3$ 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.

3) 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 rejects that composition. However, a more interesting, yet 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.
II. CONTRIBUTIONS

A. 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.

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 submit ion and on a cost model. A matching threshold is set up by services to cater for partial and total privacy compatibility.

B. The PAIRSE Project: Background

Figure 1 summarizes the architecture of this project. The Multi-Peer Query Processing component is in charge of answering the global user query. The latter has to be split into local queries (i.e., sub-queries) and has to determine which peer is able to solve a local query.

![Figure 1](image)

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. 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 RULE & MODEL

We define two privacy levels: data and operation. The data level deals with data resources. 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. For instance, let us consider a scientist who 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 being stolen by a competing company. We give below the definition of the privacy level.

A. Privacy Protocol

The sensitivity of a resource may be defined according to several dimensions called privacy rules. We all the set of privacy rules Set (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.

B. Privacy Terms

A service S will define a privacy policy, PPS that specifies the set of practices applicable to the collected resources. Defining the privacy policy PPS of S is performed in two steps. First, the service S identifies the set (noted Pp) of all privacy resources. Second, S specifies assertions for each resource in Pp. Deciding about the content of Pp and the rules (from RS) to apply to each resource in Pp varies from a service to another. PPS specifies the way S treats the collected resources (i.e., received through the mediator). We give below a definition of privacy policy.

C. Privacy Needs

A service S will define a Privacy Requirements PRS stating S’s assertions describing how S expects and requires a third-party service should use its resources. Through privacy requirements, S applies it’s the right to conceal their data (i.e., output).

D. Privacy Annotation

Privacy Annotation for WSDL-based DaaS We has 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 & POLICY COMPATIBILITY VERIFICATION

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. Negotiation to Reach Compatibility

In the previous section, we showed how privacy is checked within composite services using the dependency graph and PCM algorithm. The mediator basically discards any composition plan which is subject to privacy incompatibility from the set response CP. We intend (to help scientists in achieving their epidemiological tasks) to avoid such empty set response (i.e., CP = ∅) in order to improve the usefulness of the system. The main idea behind avoiding empty responses is to reach a compatible CPI through a privacy-aware PP negotiation mechanism, i.e., negotiation is not achieved at the expense of privacy. We presented an early idea of privacy requirement-negotiation which is designed to offer incentives to component services in order to adapt their PR. Compared to in this paper; we revise the previous idea of negotiation and provide many improvements. First, the negotiation decision is cautiously taken according to a utility-based cost function defined by a service provider. Second, the negotiation is processed with the objective to adapt the privacy policy PP of service subject to incompatibility and not its privacy requirements PR. Also, we provide many additional experimental results to show the effectiveness of our proposed techniques. In the following, we detail our privacy-aware approach that aims at dynamically reconciling incompatible services’ privacy policies while always respecting the pri

B. Privacy-aware Negotiation

In services composition (cf. Section 2), a mediator selects one service from several candidate services to perform a sub-part of the user query. Several approaches in literature use non-functional (QoS i.e., quality of service) properties to select services [1][35], where the web services provide contracts that can guarantee a certain level of QoS. Contract compliance is usually assessed through a reputation mechanism. We use a similar notion to define a non-functional property called composition reputation as a criterion to select services during composition. Composition reputation (or simply, reputation) is defined as the number of times that a service S has accepted to adapt its PPS, divided by the number of times S received PPS adaptation requests from the mediator. The more S is willing to adapt its PPS, the higher is its reputation: Reputation(S) = NAdapt(PPS)/QAdapt(PPS).

which is guided by the offers sent by the mediator to Sc and the willingness of Sc to negotiate its PPSc. The Reputation-based Privacy negotiation Module (RPM) allows the mediator to decide whether a candidate S is chosen or not depending on Reputation(S). A mediator that requests a service for composition provides feedback on the service interaction afterwards. The negotiator component handles the negotiation process by creating instances of both mediator (M(proxy) and service consumer Sc (C(proxy)) to reach a mutually compatible solution. In what follows, we detail our negotiation approach.

![Fig. 3: The Negotiation Process overview](image)

Figure 3 gives an overview of the negotiation process, where the negotiation process is to reach a compatible CPI through a privacy-aware PP negotiation mechanism, i.e., negotiation is not achieved at the expense of privacy. We presented an early idea of privacy requirement-negotiation which is designed to offer incentives to component services in order to adapt their PR. Compared to in this paper; we revise the previous idea of negotiation and provide many improvements. First, the negotiation decision is cautiously taken according to a utility-based cost function defined by a service provider. Second, the negotiation is processed with the objective to adapt the privacy policy PP of service subject to incompatibility and not its privacy requirements PR. Also, we provide many additional experimental results to show the effectiveness of our proposed techniques. In the following, we detail our privacy-aware approach that aims at dynamically reconciling incompatible services’ privacy policies while always respecting the pri

![Fig. 4: Service Negotiation Strategy](image)

![Fig. 5: Mediator Negotiation Strategy](image)

V. NEGOTIATION PROTOCOL

We propose a dynamic protocol called ReP (Algorithm 2), handled by the negotiator module. This protocol aims at automatically reconciling the mediator’s and consumer’s negotiation strategies related to consumer assertions in InC. In this regard, the negotiation protocol incorporates two state machine diagrams using the reconciliation algorithm, and finds the first alternative assertion from STran An that is compatible with Au. The algorithm ReP checks if an incentive Iq, from MStat Ri , is accepted by STran An and then checks the compatibility of the related alternative assertion Aq (instead of Au_) from STran An , where the couple (Au, Au_) is the couple (Au, Au_). Otherwise, if Aq, related to the acceptance of Iq, is not compatible with Au, the algorithm ReP will check the next incentive from MStat R*; looks if it is accepted by STran An and the previous reasoning is observed. Thus, ReP is applied to all assertion couples (related to consumer services) of InC under the condition that there exist negotiation strategies specified for each assertion (of the corresponding privacy policy) of InC. The algorithm ReP returns Rec which contains the best alternative assertions that will be compatible.

VI. V. PROTOTYPE EVALUATION

The goal of our experiments is twofold: first, we study the performance of the proposed algorithms and protocols via extensive experiments. Second, we validate the applicability of our proposal on real-life scenarios. We first describe the prototype architecture in Section 7.1. We detail the
experiments setup in Section 7.2. In Sections 7.3 and 7.4, we study the performance evaluation of the proposed algorithms (privacy compatibility checking, PCM, and negotiation, ReP respectively). Then, in Section 7.5, we report our experiment results with three real scenarios from the healthcare domain to show the impact of PCM and ReP algorithms on service composition time processing, including server-side time consumption and client-side total response time.

A. Negotiation Performance

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Operation</th>
<th>Privacy</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sce1 N → C</td>
<td>216</td>
<td>1034.1</td>
<td>202.0</td>
</tr>
<tr>
<td>Sce1 C (neg)</td>
<td>216</td>
<td>1302.6</td>
<td>400.0</td>
</tr>
<tr>
<td>Sce2 N → C</td>
<td>72</td>
<td>675.8</td>
<td>296.0</td>
</tr>
<tr>
<td>Sce2 N → C (neg)</td>
<td>72</td>
<td>776.3</td>
<td>396.0</td>
</tr>
<tr>
<td>Sce3 N → C</td>
<td>64</td>
<td>149.0</td>
<td>36.0</td>
</tr>
<tr>
<td>Sce3 C (neg)</td>
<td>64</td>
<td>217.5</td>
<td>127.0</td>
</tr>
</tbody>
</table>

Table 1: Client-side timings of the different scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Operation</th>
<th>Privacy</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sce1 C</td>
<td>216</td>
<td>1281.8</td>
<td>205.0</td>
</tr>
<tr>
<td>Sce2 C</td>
<td>72</td>
<td>401.0</td>
<td>178.0</td>
</tr>
<tr>
<td>Sce3 C</td>
<td>64</td>
<td>137.3</td>
<td>42.0</td>
</tr>
</tbody>
</table>

Table 2: Client-side timings of the different scenarios

B. Cooperation Performance

In the following we evaluate the performance of our negotiation approach. We first describe the case of incompatibility considered by the negotiation approach, before presenting and discussing the most significant results obtained from our experiments. The negotiation proposal deals with the case of privacy incompatibilities between services within a composition plan. Two services \( S \) and \( S_\) within a \( CP \) (where \( S_\) depends on \( S \)) are incompatible in terms of privacy regarding dependent resource rs if \( PR_{S} \) does not subsume \( PP_{S_\} \) for those rs. In this case, the negotiation can be performed to reach a compatible \( CP \).

VI. 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 purposes. 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.

VII. References
