I J C T A, 9(2) 2016, pp. 1403-1408
© International Science Press

Enhancing the privacy policies of hospital using XACML

B. Muruganantham*, K. Vivekanandan and S. Radhika*****

ABSTRACT

Technology advancements facilitate the online collection and publication of data about individuals, which could potentially be distributed among several organizations such as testing labs, research institutes, etc. Each organization may manage its data access and usage through a specialized Web service. In such services based interactions, data can be accessed in several ways, including manual query submission through SPARQL endpoints, automated analysis pipelines and scientific workflows, and mashup service APIs with minimal human interaction. In line with the different access scenarios, health science data is a prime example, where the focus has been on transforming the data into ontology-based repositories using RDF as a universal healthcare exchange language. Each repository defines ontology in OWL format of all concepts that can be searched for any request. OWL defines classes as a generic concept of individuals and data type properties to link individuals of those classes to their data values. Dynamic service composition may be involved, especially since the queried data may not necessarily get retrieved from a single web service. In this proposed approach, provide XACML-based implementation of a semantic-based privacy management framework that incorporates context into dynamic rule evaluation and decision enforcement.

Index Terms: Privacy, RDF, Service composition, Ontology, Dynamic

1. INTRODUCTION

Web service is a self-contained, modular application that can be described, published, located and invoked over the Web. Platform-neutral and based on open standards, Web Services can be combined with each other in different ways to create business processes that enable you to interact with customers, employees, and suppliers. Web Services use the Web to perform application to application Integration. Web Services are business process interfaces. Each Web Service is a point of interaction to give input and receive output from a business process. Allow previously incompatible applications to interoperate on the Web regardless of language, platform, and operating systems. A standalone function that can be called by many different applications.

The application web services can be addressing many real world challenges. It includes variety of fields like security, research and business. Web services a major new trend in standard- based software technology, is made up of pieces of custom- developed code that lets two or more web-based application talk to each other.

Web service reduces complexity by encapsulating business processes into reusable components. Improve interoperability by acting as a wrapper around legacy or platform-specific applications. Promote true interoperability through platform and implementation neutrality. Develop applications much faster than before. A Web Service is a simple, reliable way to blend existing systems with new applications and services.

The basic standards for web services are: XML (Extensible Markup Language), SOAP (simple object access protocol), WSDL (web services description language) and UDDI (universal description, discovery

* Assistant Professor (Sr. G), Professor, M. Tech Scholar, *Email: b.muruganantham@gmail.com*

** Department of Computer Science and Engineering, *Email: k.vivekanandan@pec.edu*

** SRM University, Kancheepuram, India, *Email: radhika.duryodhan@gmail.com*

and integration). **XML:** All Web Services documents are written in XML. XML Schema are used to define the elements used in Web Services communication.

WSDL: A WSDL definition describes how to access a Web Service and what operations it will perform. A WSDL document provides information on the data being exchanged, the sequence of messages for an operation, protocol bindings (HTTP), and the location of the service. A WSDL document defines services as a collection of endpoints.

UDDI: UDDI is an XML-based registry for businesses worldwide to list themselves on the Internet for Web Services and for other electronic and non-electronic services. Allows businesses to list themselves by name, product, location, or the Web Services they offer (Yellow Pages). Service consumers can use UDDI to discover services that suit their requirements.

SOAP: A way for a program running in one operating system to communicate with a program in the same or another operating system. A standard for exchanging XML based messages (Request/Response) normally using HTTP.

Web Services involve three major roles Service Provider, Service Registry, Service Consumer. Three major operations surround web services: Publishing–making a service available, Finding–locating web services, Binding–using web services.

One of the most widely used privacy policy languages is XACML. According to a standard XACML-based privacy policy management model, the organization hosting the web service should define a Policy Administration Point (PAP), through which policies can be defined and deployed to a Policy Decision Point (PDP). Context handling in XACML is a protocol of communication between a PDP and a Policy Enforcement Point (PEP). The PEP forms an XACML request and sends it to the PDP through the *Context Handler*, which collects initials attributes from the Policy Information Point (PIP). The PDP then uses those attributes from the PIP. The PDP then uses those attributes to evaluate policies. The PDP requests additional attributes from the context handler as needed and finally returns a Permit or Deny decision to the PEP, which enforces the final decision.

2. RELATED WORK

The literature has several works that have proposed context-aware privacy management systems [1]. Dynamic composition of different data items may be misused by adversaries to reveal sensitive information, which was not deemed as such by the data owner at the time of data collection. Atomically, these data items may not reveal personally identifiable information, but linking those items may lead to unintended breach of privacy. The problem of privacy management in Services-based interactions raises challenges especially since in web browsing, privacy protection need to be performed while the user is looking for data online.

Several technologies have been applied to achieve privacy policy enforcement by considering the requester's permission, the owner's consent and the context. Agrawal [2] leverage the Active Enforcement module of Hippocratic Database Technology (HBD) by transforming an original query to another query that is policy-compliant. These technologies include (1) active enforcement of fine-grained data disclosure policies (2) efficient auditing of past database access to verify compliance with policies (3) privacy-preserving data mining (4) de-identification of personal data using an optimal method of kanonymization and (5) secure information sharing among autonomous data sources. They describe the functionality of each component, offer example scenarios to demonstrate their usefulness, and identify remaining research challenges in securing electronic health records.

A key component of the smart grid is the ability to enable dynamic residential pricing to incentivize the customer and the overall community to utilize energy more uniformly [3]. However, the complications involved require that automated strategies be provided to the customer to achieve this goal. This paper presents a solution to the problem of optimally scheduling a set of residential appliances under day-ahead variable peak pricing in order to minimize the customer's energy bill (and also, simultaneously spread out energy usage). We map the problem to a well known problem in computer science – the multiple knapsack problems – which enables cheap and efficient solutions to the scheduling problem. Results show that this method is effective in meeting its goals.

Mashup is a web technology that combines information from more than one source into a single web application. This technique [4] provides a new platform for different data providers to exibly integrate their expertise and deliver highly customizable services to their customers. Nonetheless, combining data from different sources could potentially reveal person-specific sensitive information. In this paper, we study and resolve a real-life privacy problem in a data mashup application for the nancial industry in Sweden, and propose a privacy-preserving data mashup (PP Mashup) algorithm to securely integrate private data from different data providers, whereas the integrated data still retains the essential information for supporting general data exploration or a specific data mining task, such as classification analysis. Experiments on real-life data suggest that our proposed method is elective for simultaneously preserving both privacy and information usefulness, and is scalable for handling large volume of data.

The aim of privacy preserving data mining (PPDM) algorithms is to extract relevant knowledge from large amounts of data while protecting at the same time sensitive information. An important [5] aspect in the design of such algorithms is the identification of suitable evaluation criteria and the development of related benchmarks. Recent research in the area has devoted much effort to determine a trade-off between the right to privacy and the need of knowledge discovery. It is often the case that no privacy preserving algorithm exists that outperforms all the others on all possible criteria. Therefore, it is crucial to provide a comprehensive view on a set of metrics related to existing privacy preserving algorithms so that we can gain insights on how to design more effective measurement and PPDM algorithms.

3. PROPOSED WORK

In the previous system, collaborative service-based data sharing environments**,** data accessed from various web servers may be misused by adversaries to reveal sensitive information. A key challenge in web services security is the design of effective access control schemes that can adequately meet the unique security challenges posed by the web services paradigm [11]. In dynamic compositions of data items, queried data may not necessarily get retrieved from single Web Service. A composition plan will be required.

We build a dynamic, semantic-based privacy policy management framework on the top of the XACML reference architecture for policy-based access control. Context handling in XACML is a protocol of communication between a Policy Decision Point (PDP) and a Policy Enforcement Point (PEP) (located either on the user agent side, the Web service side, or on a gateway between the user and the service). The PEP forms an XACML request and sends it to the PDP through the Context Handler. The PDP then uses those attributes to evaluate policies. The PDP requests additional attributes from the context handler as needed and finally returns a Permit or Deny decision to the PEP, which enforces the final decision.

Composition plan will be generated where any service WS1 which depends on another service WS2. To manage data privacy, Web Services defines a privacy policy for each instance in its OWL repository.

Bio2RDF is such a system, built from rdfizer programs written in JSP, the Sesame open source triple store technology and OWL ontology [6]. Each repository manages data access through SPARQL endpoint. Whenever a user submits a query, the query first goes to the PEP, which wraps it into an XACML request and forwards the request to the PDP, which communicates with the PIP to fetch the required attributes.

In the proposed system, the PIP communicates with the Semantic Handler (SH), which looks up the required attributes in the service's repository. The PDP then uses the attributes values to evaluate the

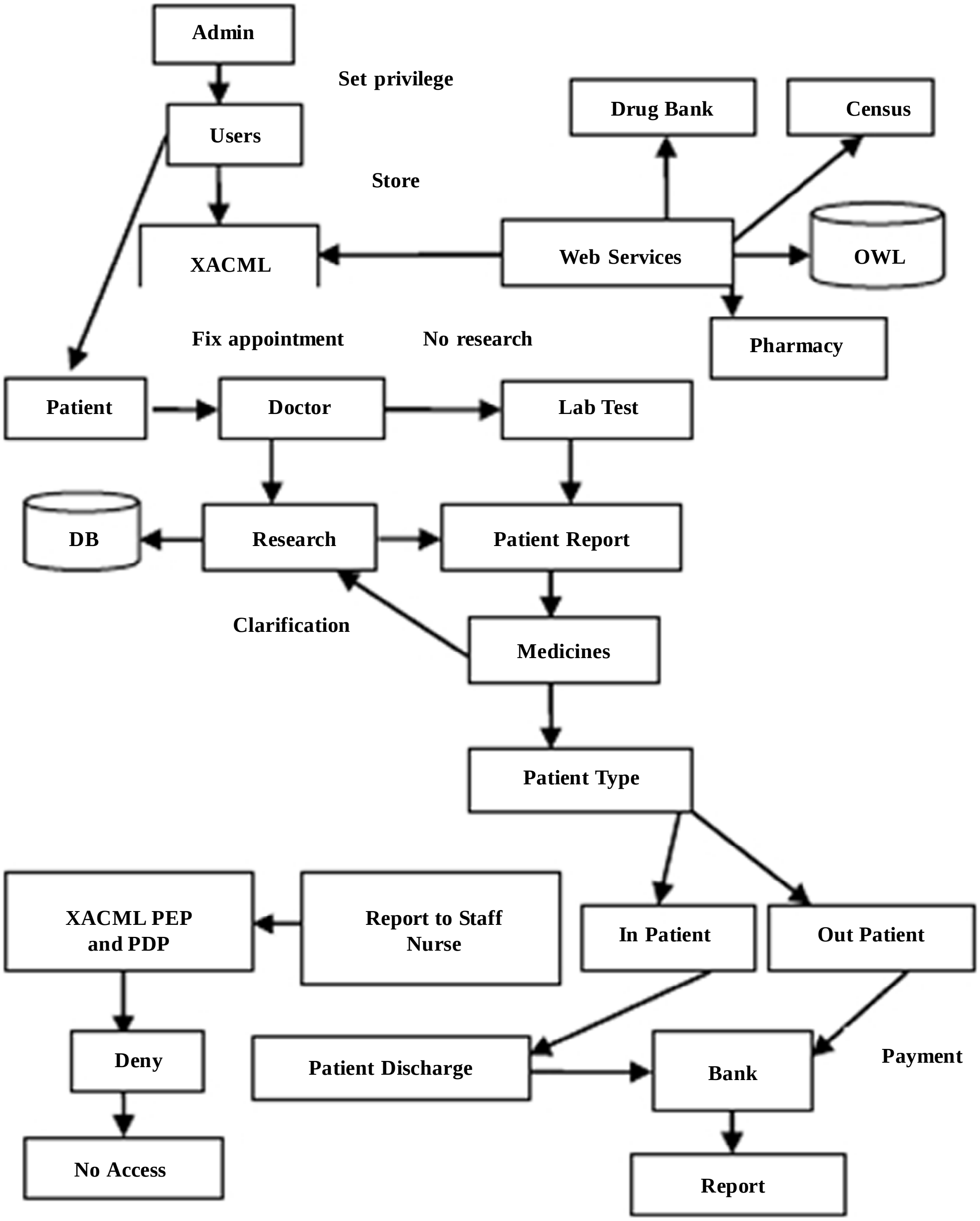

Figure 1: System Architecture

request. If a permit decision is returned, the PDP consults the semantic handler for previously recorded context of the matching data instances. The retrieved context is considered as a resource bag of context elements.

The PDP then wraps the context bag as an XACML obligation element and sends it over to the PEP together with the obligation logic to be performed. The PEP uses the obligation to perform further check by communicating with Semantic Handler. The Semantic handler passes the set of instances that match the query together with the query to the Context Handler. The context handler consists of two sub components: the *Classifier,* which dynamically classifies a query as being potentially *malicious* or *legitimate,* and the *Sensitive Data Detector,* which dynamically determines the subset of data type properties in a query that could potentially be sensitive.

3.1. Registration and Appointment

Users in the hospital environment will have an initial registration at the web end. The server in turn stores the information in its database. Now the patient login and fix appointment to the Doctor by mentioning date and time of the appointment, disease, specialist and doctor name. Each Doctor views their appointment in their appointment page.

3.2. XACML Policy for Resource Access

Admin set privileges to staffs from data accessed from different web services. Staffs will categorize as Doctor, Staff Nurse, and Lab Technician. Each web service has an Ontology repository. Data accessed from Web Service will be classified into three categories: Sensitive, Low Sensitive and High Sensitive. Based on category of the Staff, an XACML Policy will be created by the admin. Dynamic rules can be created in XACML Policy.

3.3. Web Service Composition, Diagnosis and Patient Report

Doctor view patient information such as disease, prescription etc. If doctor has a doubt about disease, he/ she can contact Research Department to retrieve Medicine or Treatment Type detail. Patient is advised to take lab test. Lab Technician provides test result to patient. If Lab Technician has doubt to deciding lab result, he/she can contact Research Department. XACML Policy will be applied to Lab Technician. Decision to access lab result will be based on Lab Technician XACML Policy. Based on test result, Doctor decides patient type: In Patient or Out Patient.

3.4. Hospital Automation and Billing

Out Patient Information will be sent to Patient Page. Patient Page contains hospital fees to be paid including lab fees, doctor fees etc. Patient will be redirected to Bank to pay fees. After successful transaction, Patient Report will be generated. If the Patient type is In Patient, Doctor sends report to Staff Nurse. If Staff Nurse view attributes, access decision check in PEP and PDP. If the access is Permit, Staff Nurse can view otherwise not. If the Patient is discharged from hospital, he/she will be redirected to Bank to pay fees. After successful transaction, Patient Report will be generated.

4. CONCLUSION

The proposed system produces an Access control set using a XACML based privacy-policy management framework. We present a dynamic, context-sensitive and semantic-based approach for privacy policy management. Composition plan will be generated for querying from multiple Web Services.

REFERENCES

[1] Abdelmonaam, Elisa Bertino ,Nariman Ammar and Zaki Malik, “XACML policy evaluation with dynamic context handling”, IEEE transcation on knowledge and data engineering, Vol. X, No. Y, Nov. 2014.

[2] Christopher Johnson, Rakesh Agrawal, “Securing Electronic Health Records without Impeding the Flow of Information”, International journal of medical informatics, 2007.

[3] Kumaraguruparan N., Sachin S. Sapatnekar and Sivaramakrishnan H, “Residential Task Scheduling Under Dynamic Pricing Using the Multiple Knapsack Method”, IEEE conference of Innovative technology, 2010.

[4] A.L. Gancarski, C. Ghedira, D. Benslimane, and M. Barhamgi, “Privacy-preserving data mashup”, in international conference on advanced information networking and application, 2011.

[5] Dan Lin, Elisa Bertino, and Wei Jiang, “A Survey of Quantification of Privacy Preserving Data Mining Algorithms”, IEEE transaction, 2011.

[6] B. Francois, J. Morissette M.A. Nolin, N. Tourigny and P. Rigault, “Bio2RDF: towards a mashup to build bioinformatics knowledge systems,” J. biomedical informatics, vol. 41, no. 5, pp. 706–716, 2008.

[7] EHealth Information Platforms (EHIP). [Online]. Available: http:// distrinet.cs.kuleuven.be/research/projects/EHIP, Dec. 2013

[8] Axiomatic Language for Authorization (ALFA). [Online] Available: http://www.axiomatics.com/solutions/products/authorization-for-applications/developer-tools-and-apis/192-axiomatics-language-for-authorization-alfa.html

[9] Sun's XACML Implementation. [Online]. Available: http://sunxacml.sourceforge.net/, 2003.

[10] WSO2 Balana Implementation. [Online]. Available: https://github.com/wso2/balana, 2013.

[11] A. Ghafoor, E. Bertin and R. Bhatti, "A trust-based context aware access control model for web-services," in Proc. Int. Conf. Web Services, 2004, pp. 184–191.