International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 3, March 2016

Enhancing the Privacy Policies of Hospital Using XACML

[1] Mr. B. Muruganantham, [2] Dr. K. Vivekanandan [3] Ms. S. Radhika
[1] Assistant Professor (Sr. G) [2] Professor [3] M.Tech Scholar
[1][2][3] Department of Computer Science and Engineering
SRM University, Kancheepuram, India
[1] b.muruganantham@gmail.com, [2] k.vivekanandan@pec.edu, [3] radhika.duryodhan@gmail.com

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 prime factor where the focus is on transforming the data into ontology-based repositories using RDF as a 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 is retrieved from than a single web service. In this proposed approach, XACML-based implementation of a dynamic privacy policy management framework that incorporates context into dynamic rule evaluation and decision enforcement.**

Index Terms **- Privacy, RDF, Service composition, Ontology, Dynamic**

I. INTRODUCTION

Web service is a type of application which can be invoked either by describing or publishing over the web. Web service is free from platform and is based on open standards. Web Services is combined with different service in different ways to create business processes that allows customers, employees, and suppliers to interact with each other. Web Services perform application-to application Integration using web. Web Services are business process interfaces. Each service over the web is a point of interaction in a business process. Web service allows incompatible applications to operate 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 made up of custom-developed code that allows two or more web-based application to interact with each other. Web service reduces complexity by encapsulating business processes into reusable components. Web service promotes true interoperability through platform and implementation neutrality. Develop applications much faster than before.

Web Service is a simple and reliable way to blend existing systems with new applications and services.

XML: All Web Services documents are written in XML. XML Schemas are used to define the elements used in Web Services communication.

***WSDL*:** The format of XML describing the services of network set as endpoint of the operating message contained the information of either the procedure-oriented or document-oriented.

UDDI: Universal Description Discovery and Integration (UDDI) is described as the "a services set for supporting the business discovery and description of the business and organization and other providers of the web services, the web services, which making the availability of the web services and interfacing with the technical aspects of those services.

***SOAP*:** Simple Object Access Protocol is a protocol based on the XML, which allows the exchanging of the data over the HTTP (Hypertext Transfer Protocol). The World Wide Web Consortium (W3C) [presenting the definition of the Simple Object Access Protocol (SOAP) is a protocol which have the light weight intended for the infrastructure

exchanging information in a distributed and decentralized environment. 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 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*, initials attributes is collected from Policy Information Point (PIP). The PDP then uses those attributes from the PIP. 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.

II. 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 important to the data owner at the time of data collection.

Atomically, these data items may not reveal personally identifiable information, but linking these data items may lead to unintended breach of privacy. The problem faced in a privacy management in Services-based interactions raises challenges especially in web browsing, since privacy protection need to be performed simultaneously while the user is looking for data online.

Several technologies has 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) disclosure of finegrained data by active enforcement policies (2) to verify and access compliance with policies efficient auditing of past database is done (3) privacypreserving policy data mining (4) kanonymization method is used for de-identification of personal data and (5) sharing of information among autonomous and secure data sources. These technologies describe the functionality of each component, often offer example scenarios or condition to demonstrate their use and identify remaining research challenges in securing electronic health records focussing on privacy of data.

A key component of the smart grid is the ability to enable dynamic or simultaneous residential pricing to incentivise 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. In this paper the authors gives a solution for the problem of optimally scheduling a set of residential appliances under day in order to minimize the customer's energy bill which is ahead of variable peak pricing (and also, simultaneously spread out energy usage). The authors map the problem to a well known problem in computer science – the multiple knapsack problems – which gives cheap and efficient solutions to the scheduling problem. Results show that this method is effective in meeting its expected goals. Mashup is a web technology that combines information from one or more web source into a single web application. This technique [4] provides a new platform for different data providers to extensively integrate their expertise and deliver high customizable services to their customers. Nonetheless, combining data from different sources or services could potentially reveal person-specific sensitive information which breaches the privacy. In this paper, the authors study and resolve a reallife privacy problem in a data mashup application for a industry located in Sweden, and propose a privacy-preserving data mashup (PPMashup) algorithm to secure integrate private and sensitive data from different data providers, whereas the integrated data still holds the minimal information for supporting general data disclosure or a specific data mining task, such as classification analysis. Experiments on real-life data like personal details of a user, proves that our proposed method is effective for simultaneous use of preserving privacy and is scalable for handling large volume of data.

The aim of privacy preserving data mining (PPDM) algorithms is to take enormous 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 to identify the suitable evaluation method and the development of related application. Recent research in the area has dedicated much effort to determine the performance between the right to privacy and the need of knowledge discovery. It is often the case that no privacy preserving algorithm exists that performs less than the possible criteria mentioned in the method. Therefore, it is crucial to provide a composed view on a set of metrics related to the existing privacy preserving algorithms so that we can get an idea how to design more effective and fast PPDM algorithms.

III. PROPOSED WORK

In the previous system, collaborative service-based data sharing environments**,** data accessed from various web servers may be misused by any third party to reveal sensitive information. A key challenge in the web

services security is the design of an effective access control mechanism which provides privacy for the user and can 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). The PEP forms an XACML request and sends through context handler to the PDP. The PDP later uses those attributes to evaluate policies. The PDP sends additional requests for additional attributes from the context handler as desired and finally sends a Permit or Deny decision as reply to the PEP, which enforces the final decision.

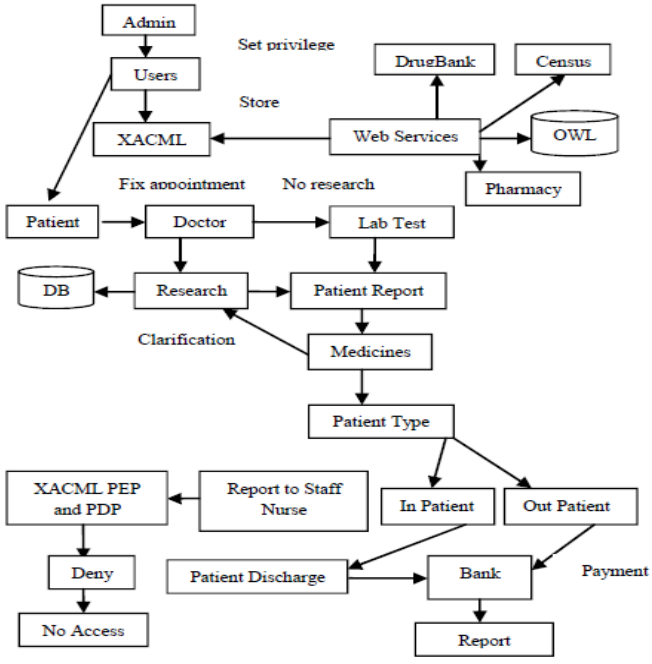

Fig 3.1 System Architecture

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 a system, 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 in turn communicates with the PIP to fetch the required attributes.

In the proposed system, the PIP communicates with the Semantic Handler (SH), to look for the required attributes in the service's repository. The PDP later uses these attributes values to evaluate the request sent by the PIP. If the decision is permit, the PDP ask 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 carried out. The PEP uses these obligations to perform further checking 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 is composed of two sub components: the *Classifier*, which dynamically classifies a query as being potentially *harmful* or *legitimate* and the *Sensitive Data Detector* dynamically, determines potentially sensitive data from the subset of data type properties in a query.

3.1Registration 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

The Patient Information i.e. report which contains all the information about the patient 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.

IV. CONCLUSION

The proposed system produces an Access control set using a XACML based privacy-policy management framework. We describe a dynamic, semantic-based and context-sensitive 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/auth orization-for-applications/developer-tools-andapis/ 192-axiomatics-language-for-authorizationalfa. 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 trustbased context aware access control model for webservices," in Proc. Int. Conf. Web Services, 2004, pp. 184–191.