Semantic Context Aware Authorization for Securing Data in Cloud

1Auxilia M and 2Dr. K. Raja

1Research Scholar, Sathyabama University, Chennai, India.
2Principal, Alpha College of Engineering, Chennai, India.

ABSTRACT

Data security is an open research issue in cloud computing. Small and medium sized businesses are now grabbing cloud for their infrastructure and storage needs. Once data are placed on cloud users can access it from anywhere, anytime using any devices ranging from desktops to smartphones with ubiquitous high speed Internet connection. Access control policies based on roles, identities are seldom enough to address data security risks in cloud. The reasons are cloud is highly dynamic, scalable and highly autonomic. Context in which the user may access the data also cannot be predicted. To deal with these issues, this paper advocates the implementation of a semantic context aware authorization model. This model controls data access based on context information and applies semantic technologies for both policy and context specification. We also present some set of rules we have framed to handle dynamic context changes and its adaptation in policies.

INTRODUCTION

The earliest and the oldest computing paradigm are computers occupying large rooms with little computing facility. They are later replaced by Mainframe, PCs, and Client-Server Computing. Clusters and grid are the well known paradigms for distributed systems. Cloud Computing is a paradigm shift that aroused in the year 2008. As per NIST, Cloud computing is a model for supporting appropriate network access to a group of computing resources with minimal human interaction based on the consumers need instantly at affordable cost Mell P and T. Grance (2009).

Cloud supports deployment models such as private cloud, public cloud, community cloud and hybrid cloud and three service delivery models, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) Antonios Gouglidis et al (2014), Chunming Rong et al (2012), Diogo A. B. Fernandes et al (2014), Hong Sun et al (2014). Cloud computing possess the distinctive features like on demand access, resource pooling, rapid elasticity, pay as you go service and ubiquitous network access. These features make it more attractive and affordable.

Confidentiality (guarantying that data are not disclosed to unauthorized individuals), Integrity (Ensuring that data hasn't been altered in transit from its original form, either accidentally or deliberately), and Availability (avoidance of the unauthorized withholding of data) are considered as security requirements Antonios Gouglidis et al (2014).

Nowadays anyone carrying his/her own device can access data from cloud from their workplaces. This is widely called as Bring Your Own Device (BYOD). Some of the advantages of BYOD are namely: customers need not to depend upon the resources of the organization they are working in. All they require is Internet connectivity in that device. Even organizations do not control this as their cost of investment on resources is being cut down and hence it improves productivity. The need of the user may increase or decrease from time to time and their context may also vary rapidly. Hence it is necessary to consider context information into account while framing access control policies. Information portraying circumstances or situation of a person can be defined as context. Context can be a person’s location, activity and so on. And it is also not static, it may change over time. To interpret and understand context accurately and effectively Ontology is used. The reasons are 1) a general ontology facilitates knowledge sharing in distributed systems, 2) improves interoperability among different domains, and 3) promotes efficient reasoning of contextual dat.
Traditional Access Control Policies like identity based access control and role based and attribute based access control models are not sufficient for cloud computing [Pierangela Samarati1 and Sabrina De Capitani di Vimercati (2002), Salim Khamadia (2013), Takabi H et al (2010), Vivy Suhendra (2011)]. Thus this paper primarily aims at facilitating secure data access in cloud environments by means of a novel, efficient semantic context aware access control model. This model considers policy rules and requires user to provide all the necessary and relevant information for processing their request. When some user wants to access data, the security policy enforcement framework must check its policy rules and reveal it only if the policies are satisfied.

The main objective of this paper is to make clear how contexts can be detained using ontology and incorporated in access control policies.

In our paper Section II discusses various works related to our research. Section III explains our proposed semantic context aware authorization model. In section IV we present our part of implementation and section V shares our conclusion.

Related Work:

Grid differs from cloud computing in the way that, users have direct control over their resources whereas in cloud the users have no direct control over the remote resources. They rely upon the Cloud Service Providers. Hence raising additional security concerns in terms of access control. Reason is that the consumers rely upon the security mechanisms provided by the service providers Chunming Rong et al (2012).

A Semantic Access Control Policy Language (SACPL) was proposed by Hu et al. for overcoming the drawbacks of conventional authorization models in the cloud. They introduced a language called SACPL aiming to resolve the interoperability issue of distributed access control policies Hu et al (2010). However the language was not so expressive and did not have the means to include contextual information.

A work by Chang Choi et al. depicts the RBAC (Role Based Access Control), C-RBAC (Context aware RBAC), tRBAC (Temporal RBAC). But it did not suggest an effective and practical solution. Reason is they lack in considering security levels between objects and also not ample to form different kinds of dynamic relationships that can occur between objects Chang Choi (2012, 2014).

Ontology based approach towards RBAC was proposed by Hong Sun et al. It simplifies the architecture of roles in cloud storage Hong Sun (2014). For modeling the semantic knowledge RDF schema is used. RDF schema lacks in considering interoperability issues and autonomic nature of cloud computing.

Another work modifies the XACML Policy evaluation Engine and it does not meet the security requirements of cloud computing. Further it focused only on one type of security attack which is not enough for handling cloud environment Cristina Ba’cescu et al (2011). A work by Takabi et al (2010) proposes a global authorization model for handling data access on clouds. But it fails to consider the conflict resolution of policies and also the context information for access control. In autonomic computing, a semantic policy framework called KAoS Yuh-Jong Hu (2012) is used. But this paper does not have a clear separation between the access management and context management.

2. Semantic Context Aware Access Control Model:

In this section we briefly explain the architecture details of our model and Data Flow Diagram depicting access control decision is made.

A. Architecture:

The architecture of Semantic Context-Aware Access Control Model is shown in Figure 1. This model is composed of two parts: Context handling system and Access Control System. The implementation is separate for both the systems. But still, they are connected through an ontology base.

![Fig. 1: Architecture of Semantic Context Aware Access Control Model](image-url)
The model consists of a Context Handler, Ontology Base, Policy Decision Point and Policy Administration Point. A Context Handler acts as a mediator between real world information and the Ontology Base. It performs two operations: context retrieval and rule management. Context retrieval process gathers information from the real world and asserts the context attributes into the Ontology Base. For retrieving context from the real world, sensors can be used. For example, if a user makes access request through a smart phone, then the inbuilt sensors in the phone like Google map will retrieve the location details. Thus, it performs direct assertion of concepts based on the context categorization. Rule management is the process of inferring over the attributes asserted by the context retrieval process. Thus, an assertion is made twice in this model proving it to be an effective acquisition of context information. The context information may include subject, object, action and environment attributes.

Ontology Base is a repository for storing knowledge base about the context. This ontology can be used for acquiring the knowledge about the context and information can be inserted into the access control policy as attributes. After creating the ontology, the Ontology Base submits the information to Context Handler.

Policy Decision Point evaluates the access control request made by the user against the access control policy. If the real world context attributes matches the attributes in the access control policy, then the PDP will make a decision to grant the request. Otherwise, the request will be denied.

B. Data Flow Diagram:

Our Semantic Context-Aware Access Control Model makes access control decisions in the following manner.

0. This is a preprocessing step.
   a. Ontology is created. Policy Administration point uses the ontology for framing the policies.
   b. Ontology is loaded into the context manager. The context manager stores the deduced facts associated with the user, resource and contexts and actions.

1. The Policy Administration Point creates a policy and sends it to Policy Decision Point. The PAP in this model differs from PAP in standard eXtensible Access Control Markup Language (XACML) in the way of looking up context ontology. It looks ontology in the ontology base to include the context attributes to frame context-based access control policy.

2. A user request for data access in the cloud. The request is submitted to the context handler. The context handler attaches the attributes of subject, object, action and environments.

3. At run time, the context handler creates an access request using eXtensible Access Control Markup Language (XACML).

4. The PDP request for additional user, resource, action and the context attributes to the context handler.

5. The context handler forwards the request to the attribute retriever.

6. The attribute retriever gets the requested attributes from the context manager. The context manager will have the semantic semantics for user (the entity making access request), resource (the entity for which access request is made), action (the operation to be performed on the object) and context (the location information). From these semantics, attributes are inferred using SWRL rules. The inferred attributes are sent to the context handler as response.

7. The context handler in turn forwards the response to the Policy Decision Point.

8. The PDP assess the access request against the policies that are already framed.

9. The PDP returns access decision to the context handler.

10. If the decision is allow, the user can access the resources, else the access is denied.

![Data Flow Diagram](image-url)
Discussion:
This section describes our early stage of our model implementation. As the first step, we developed Ontology for context information. The domain, we have chosen is Healthcare for sharing the medical records among the physicians in case of emergency situations. Cloud proves to be an effective solution to health care. Hospitals can store their patient details in the cloud so that they could get help from the specialist anywhere anytime in case of emergency.

A. Construction of Context Ontology
To design ontology, we have chosen Protege4.2 Ontology development tool. The ontology is created using two terms: class and property. Class refers to set of individuals. Property refers to attributes linking an individual to another individual. Super class and sub class relationship is followed among classes. The default superclass for all the subclasses is Thing. Figure 3 shows the class and subclass relationship for our context ontology.

**Fig. 3:** Class and Subclass Relationship in Context Ontology

Figure 4 shows the how classes are interrelated via properties. For the health care domain, we have created properties like "inCall", "inSameCall", "inConsult", "hasLoc", and "isA". These attributes are inserted into the access control policy automatically and also manually.

**Fig. 4:** A Visual Depicting How Classes are Related Using Properties

The following fragment shows how the context attributes are inserted into the access control policy.

```xml
<Policy policyID="1" Decision="allow">
  <Target>
    <Subject>
      <Anybody/>
    </Subject>
    <Object>
      <Anything/>
    </Object>
    <Action>
      <AnyOperation/>
    </Action>
  </Target>
  <Context>
    <Enforce>
      <SubjectAtt Designator AttID="inConsult" DataType="string"/>
    </Enforce>
  </Context>
</Policy>
```
We are still at the verge of implementing this model in cloud. We would use EUCALYPTUS cloud.

Conclusion:
Cloud computing is believed to offer effective solutions for all types of businesses and users. Cloud also possesses the features of autonomic computing. Any networked device is used to access the resources from anywhere at any time. Its rapid growth leads researchers to provide more promising solutions to open research issues like security, energy efficiency, Quality of Service and so on.

In this paper, we addressed the issue of data security limiting the widespread adoption of the Cloud. Traditional access control models are ill-suited for cloud. A contemporary access control mechanism is needed. Thus in addition to the access control mechanism favored by CSPs we introduce the semantic knowledge of context using ontology and OWL to represent the concepts. Thus, we incorporate a clear separation between access management and context management.

This separation is noteworthy because the administrators do not have to worry about the context meaning while writing access control policy. Adjaently, we can update and add the meaning of context without changing the policy.

Our early stage of implementation demonstrates construction of the context ontology, showing the classes and properties of context and assertion as a result of rules using SWRL. We thought of incorporating our model into Eucalyptus Cloud or Cloudsim as future work.

REFERENCES


