RESEARCH ARTICLE OPEN ACCESS

A Framework for Predicate Based Access Control Policies in Infrastructure as a Service Cloud

B.Srinivasa Rao*, Dr.G.Appa Rao**
*(Department of Computer Science & Engineering, GITAM University, Visakhapatnam - 45)
** (Department of Computer Science & Engineering, GITAM University, Visakhapatnam - 45)

ABSTRACT
Infrastructure as a Service (IaaS) is the service with which IT of enterprises integrated for on-demand services. Different deployment models of cloud further makes it flexible so as to meet the requirements of users. As the customers' policies are not same, Cloud Service Provider (CSP) needs a flexible architecture to accommodate the varied requirements of customers with respect to access control. The existing access control models such as Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC) have limitations. The combination of RBAC and ABAC also could not offer fine grained access control. We also studied the RBAC model offered by Open Stack and came to know its limitations in catering to diversified needs of customers. The One Size Fits for All policy cannot provide flexible access control due to the aforementioned reason. Therefore a more flexible access control model is required. In this paper we proposed a framework with Predicate Based Access Control (PBAC) in general and then implemented it in Open Stack. Our empirical results revealed that the proposed framework can improve the granularity with fine grained access control mechanism. Though our framework is at primitive stage, it shows significant step forward in access control policies for IaaS clouds.

Keywords **-** Authorization, predicate based access control, Infrastructure as a Service, Open Stack, fine-grained access control

I. INTRODUCTION

Cloud computing has changed the way IT assets are maintained and used by enterprises. As a new computing paradigm cloud is able to serve organizations and individuals with huge pool of shared computing resources. Such resources can be accessed in pay per use fashion. There are many services being offered by cloud. The three important services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Out of these services, the IaaS is the widely used service which provides storage and other infrastructure services on demand. Cloud has been maturing functional aspects of IaaS. However, the security and access control mechanisms are yet to be improved further. For cloud users, security has been a concern as the data is outsourced to remote servers and treated as untrusted. Another reason for this is that the data of cloud user is not maintained in the local system and there is no matured interoperability between could service providers. In case of outsourcing of IT infrastructure there are many challenges to be addressed. In the cloud computing scenario access control is inevitable. Infrastructure related resources sc as IaaS and Virtual Machines (VM), networks and storage.

With respect to traditional computing resources there are means to have controlled access to resources. Policies can be established and thus personnel stick to the policies while gaining access to the resources. However, in case of IaaS cloud the resources are virtual and remote in nature. The access control policies of this are very much different from that of physical world. The major issues include the policies of enterprises with in-house resources cannot be directly used with cloud computing environment as the resources are not owned by them. Different users want to have their own access control policies. Therefore keeping all of them built into the cloud infrastructure is not practically feasible. Therefore a flexible and feasible access control framework is desirable in the cloud computing environment. The present role based access control and extended role based access control mechanisms with attributes result in issues mentioned above. More fine grained access control is required in order to safeguard IaaS resources.

Table 1 – Acronyms used

Acronym	Description
IaaS	Infrastructure as a Service
RCFO	Runtime Control Flow Obfuscation
RBAC	Role Based Access Control
MAC	Mandatory Access Control
DAC	Discretionary Access Control
DRM	Digital Rights Management
TM	Trust Management
XACML	eXtensible Access Control Mark-up Language
RDA	Remote Data Auditing
MCC	mobile cloud computing
HPC	High Performance Computing

The diverse access control need for IaaS is realized. In this paper we present a novel framework with predicate based access control specially designed for data access control in IaaS cloud. We implemented out framework using OpenStack as IaaS cloud. Our contributions in this paper are as follows.

- We proposed a predicate based access control (PBAC) mechanism for IaaS cloud which provides more fine grained access to the cloud data.
- We proposed a framework with PBAC as underlying access control mechanism. Out implementation is done using OpenStack as IaaS cloud.

The remainder of the paper is structured as follows. Section II provides review of literature. Section III presents the proposed framework. Section IV provides details of the experimental results. Section V concludes the paper besides providing directions for future work.

II. PREDICATE BASED ACCESS CONTROL

Predicate based access control basics conceived by us are provided in this section prior to adapting it to the IaaS cloud implemented using OpenStack. In the context of relational and non-relational data stored in IaaS, Figure 1 shows a generic framework for the implementation of predicate based access control.

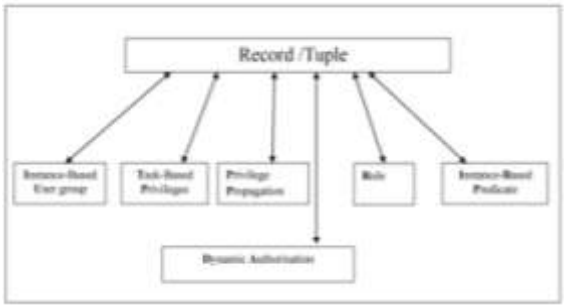

Figure 1 – Generic framework required for predicate-based access control model

Instance Based User Group: When a master record is created, there might be some users who are involved in that. Such user-group should be able to access that record to be precise. Therefore it is essential to have an instance-based user group associated with the master tuple.

Instance-Based Predicate: Having access control record for every master tuple or record is not an effective practice. It leads to more number of access control records which exceed actual records in master relations. Therefore it is essential to have a predicate based access control. A predicate is some clause that can be used with queries. For instance a doctor can access all healthcare records in which his ID is stored. This kind of predicate can avoid maintaining so many access control records pertaining to different master tuples.

Task-Based Privileges: Certain users are allowed to perform definite tasks for which privileges are to be granted. When performing a task user is allowed to access only one master record. And the same user may be allowed to gain access to multiple master tuples with respect to another task. Thus task-based privileges can simplify access control.

Privilege Propagation: In some select situations privileges are propagated from one role to another role. Such privileges are not determined statically. Therefore it is essential to have privilege propagation feature for effective access control mechanism. For instance a user in clerk role needs to access different loan records based on the field officers' recommendations. Therefore they need to have different privileges in different situations though the task remains same.

Role: Role plays a vital role in controlling access. Even the predicate – based access control model presented in this paper can enjoy the advantages of role based access control. While performing a particular task a user who belongs to a role can gain access to a particular tuple only. It is true with all users of all roles. An important observation is here is that different users of a similar role also can involve in different process instances. Thus it is very clear that the concept of role and the concept of instance-based user group are distinct. They are not interchangeable.

Dynamic Authorization: There are some situations in which users can gain access to historical records for learning and better decision making. Nevertheless, there are some sensitive tuples of particular department that needs are to be exempted from the dynamic authorization. Stated differently, there should be provision in the access control model to provide access to historical data while exercising restrictions to sensitive tuples at the same time.

Components of Access Control Model

There are many components in the proposed access control model as shown in Figure 2. The components are subject, task, object, constraint and privilege.

Subject: It is the first component that is made up of user, and role, runtime instance based user group. A group of users is represented as **U**. Role represents a collection of privileges that are assigned to users of that specific role.

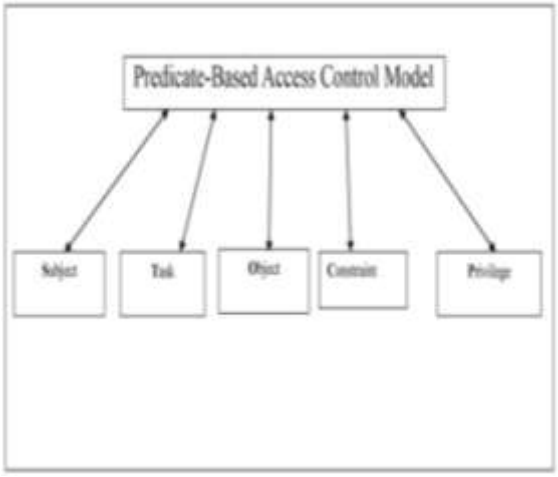

Figure 2 – Shows the components in PBAC model

Task: The task is a component. A set of components of workflow is represented as a tree.

Object: This is the third component. There are many objects involved and each object can have properties or attributes pertaining to security and access control. Such attribute is known as security attribute. These are used to define diversified set of files of different kinds such as audio, video, .exe, instance of Java classes, a relation instance, a database, and set of relations and so on. **O** represents set of objects.

Constraint: This is the fourth component denoted by **C** which refers to set of constraints. Every constraint is an expression that results in a Boolean value. There are many operators for which can produce Boolean result as shown in Figure 3.

Figure 3 – Shows operators that are used to build constraints

Privilege: This is the last component in the model. Let P represents set of access rights or privileges. These access rights are exercised by subjects on objects. There are different types of privileges such as new, destroy, select, insert, update, delete, read and edit. Out of them new, read, edit and destroy are for document files and the rest are for database objects.

Existing Access Control for OpenStack

OpenStack has its access control mechanism as illustrated in Figure 4. There are users, roles, objects, expressions and permissions. Permission is denoted as an operation on object. Users are assigned to roles. There is possibility that multiple roles are assigned to a user. Each operation is associated with Boolean expression. The expression is evaluated by interpreting it. It may result in true or false.

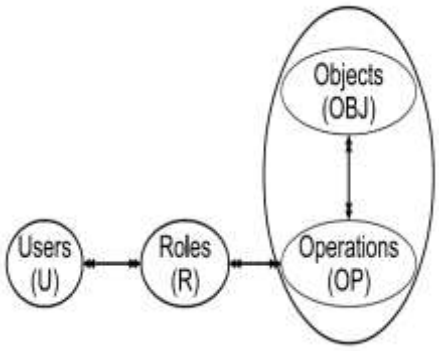

Figure 4 – Access control mechanism of Open Stack

When user tries to operate an object, there will be policy check based on the role and privileges of user. It is RBAC model has some standard operations. However, it doe has some limitations. First, users do not have their own access control policies. Second, the access control is not fine-grained. In other words it can be improved to have fine grained access control.

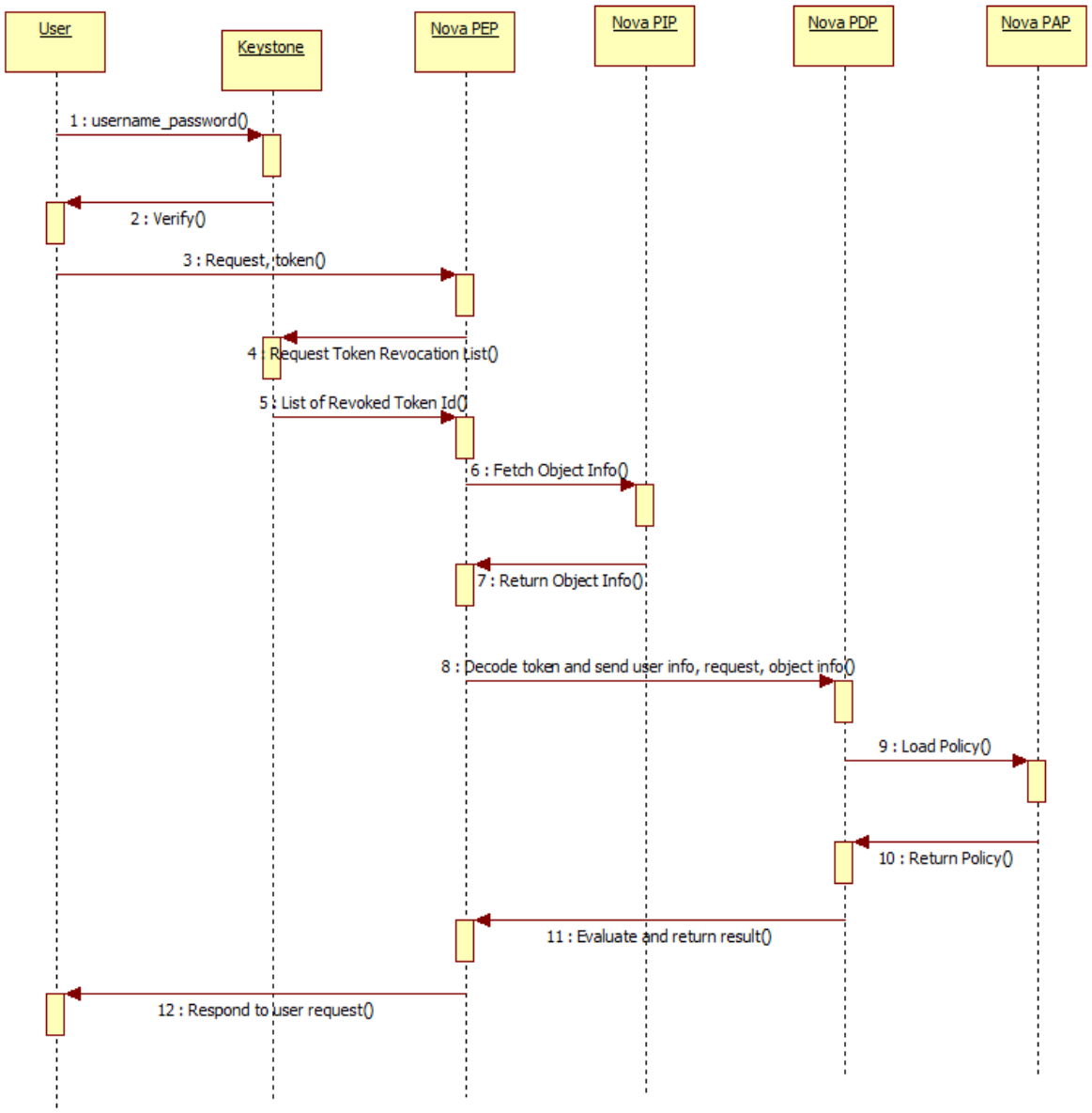

Figure 5 – Authorization mechanism in OpenStack using asymmetric keys

As shown in Figure 5, a user sends his credentials to Keystone. Keystone verifies the credentials and generates a token. Keystone also does this along with signed user data. Then Keystone sends the token back to user along with service end points. Then user is able t send request to Nova's PEP component where token is verified and validated. PEP gets object details from local PIP and decodes the verified token with its public key. Then user data and object data are sent to PDP which gets policies to evaluate request. Finally true or false is returned which determines whether service needs to be provided or not.

III. PROPOSED ACCESS CONTROL MODEL FOR OPENSTACK

One size for all kind of policies enforced by OpenStack is not feasible when customers want to have different access policies on their data. Therefore we proposed a generic framework that is meant for fine grained access control. It is known as PBAC.

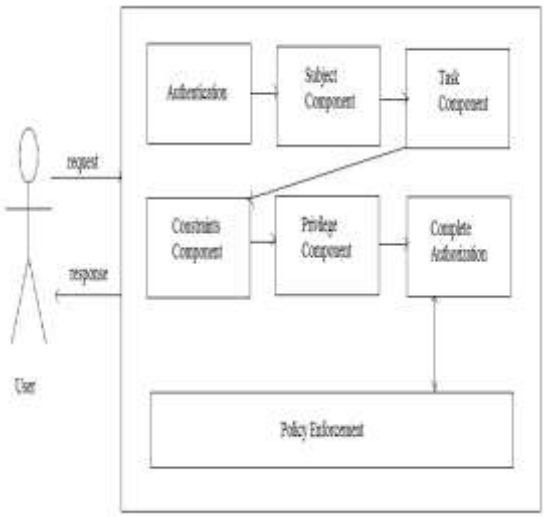

Figure 6 – Framework for PBAC

As shown in Figure 6, there are many components involved in the framework. After authentication, all the components are in place as described in one of the previous sections. Policy enforcement is done based on the PBAC mechanism.

The modelling of these components is illustrated in Figure 7.

Figure 7 – Access control modelling as per PBAC mechanism

As shown in Figure 7, there are formal notations used in the predicate control mechanism. The notations and the aspects like role hierarchy, task tree, user role assignment, role task assignment, object privilege, permission assignment are clearly defined. These are used in the implementation PBAC with OpenStack.

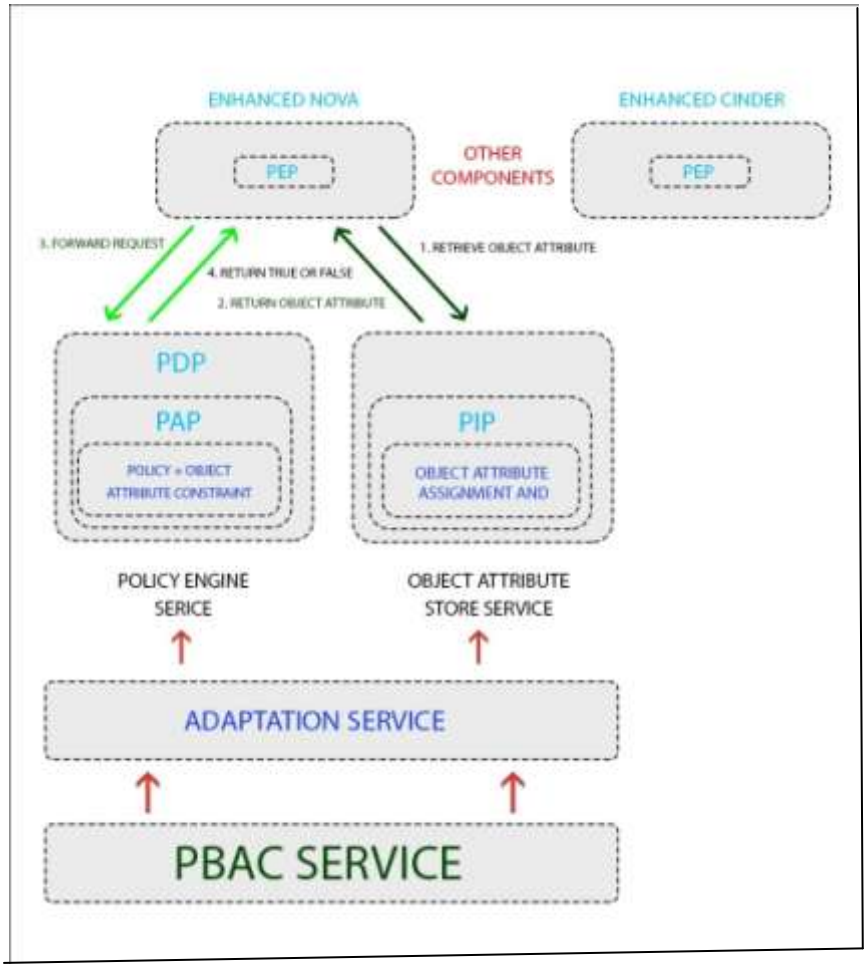

Figure 8 - Proposed PBAC enforcement model

As shown in Figure 8, the PBAC enforcement model incorporated in OpenStack contains additional services such as PBAC service and Adaptation service. These services work in tandem with the existing components like Nova, Cinder, and other services available in OpenStack.

IV.EXPERIMENTAL RESULTS

We built a private cloud using OpenStack. Four physical machines are involved in the cloud. Out of them one node acts as controller, one node is networking node, and two nodes are compute nodes. The controller and network nodes have configuration such as 24 cores CPU, 1TB disk and 24 GB RAM. For Nova compute nodes the configuration is 16 cores CPU, 1TB disk and 98 GB RAM. Out experiments are made in terms of average time taken for token generation in Keystone, average time taken for Nova communicating with PolicyEngine, time taken in presence of different number of constraints used in proposed PBAC in OpenStack.

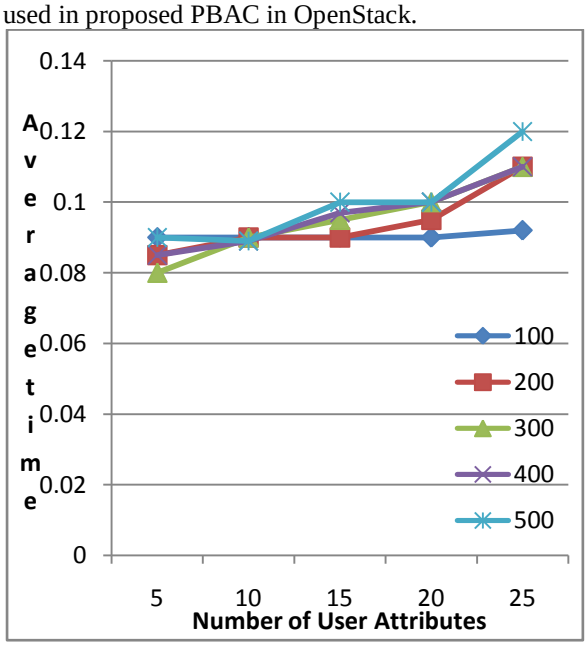

Figure 9 – Average time for token generation in Keystone

As shown in Figure 9, it is evident that there is increase in the time taken when number of user attributes is increased. The experiments are made with different user data length like 100, 200, 300, 400 and 500. The time taken is the average time for token generation in Keystone. There is another observation that the user data length has its influence on the average time taken.

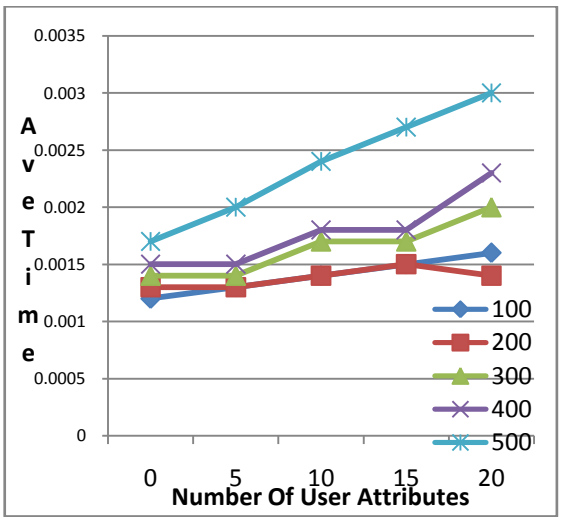

Figure 10 – Average time for Nova communicating with PolicyEngine

As shown in Figure 10, it is evident that there is increase in the time taken when number of user attributes is increased. The experiments are made with different user data length like 100, 200, 300, 400 and 500. The time taken is the average time for Nova communicating with PolicyEngine. There is another observation that the user data length has its influence on the average time taken.

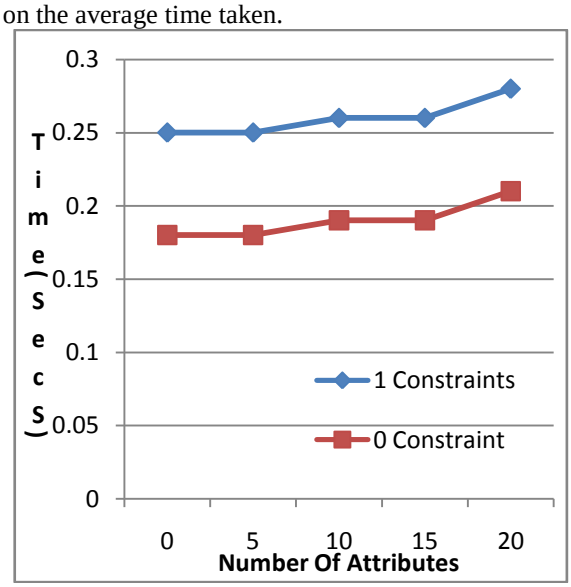

Figure 11 – Show time taken when 1 constraint is used

As can be seen in Figure 11, it is evident that there is increase in time taken when number user attributes is increased. Another observation is that when there is a constraint the time taken is more.

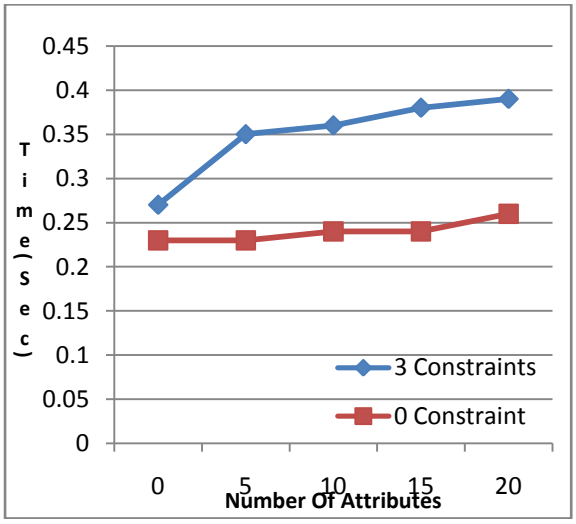

Figure 12 – Show time taken when 2 constraints are used

As can be seen in Figure 12, it is evident that there is increase in time taken when number user attributes is increased. Another observation is that when there are two constraints the time taken is more.

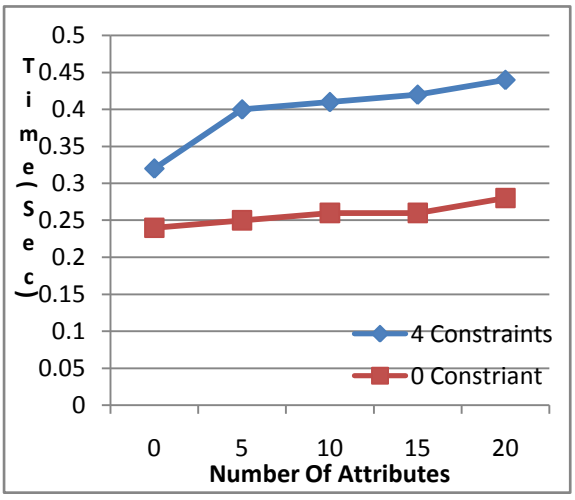

Figure 13 – Show time taken when 3 constraints are used

As can be seen in Figure 13, it is evident that there is increase in time taken when number user attributes is increased. Another observation is that when there are three constraints the time taken is more.

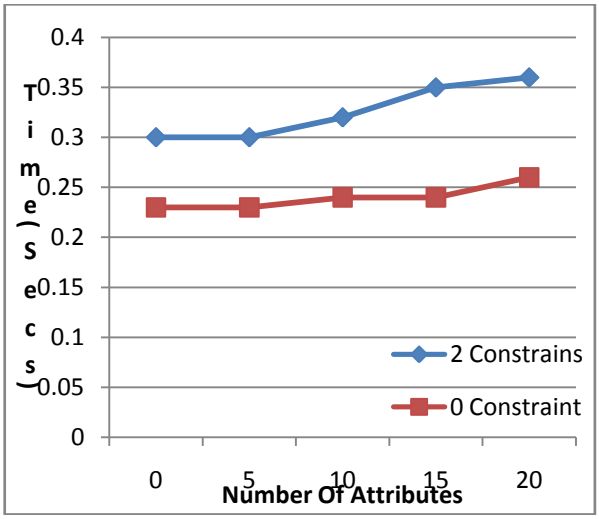

Figure 14 – Show time taken when 4 constraints are used

As can be seen in Figure 14, it is evident that there is increase in time taken when number user attributes is increased. Another observation is that when there are four constraints the time taken is more.

V. CONCLUSIONS AND FUTURE WORK

Cloud computing has proved to be the computational model which will go a long way in the IT strategy of enterprises. Especially Infrastructure as a Service (IaaS) is the cloud service which is widely used. However the users of cloud are concerned with security and flexible access control to their data. Many access control mechanisms came into existence. Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC) have limitations as they cannot provide fine grained access control to the possible extent. A promising access control mechanism that provides fine grained access control is known as PBAC. In this paper we proposed PBAC generic framework and then adapted to IaaS cloud. We implemented the framework in OpenStack as it is open source and supports IaaS service. OpenStack follows One Size Fits for all approach in the policies of access control. However, in the real world, users are expecting different access policies to control the outsourced data. To overcome this problem we proposed and implemented a framework within the IaaS service components of OpenStack. Our empirical results revealed that our approach can provide fine grained and flexible access control to cater the needs of different users. It is a significant step forward in exploring such policy enforcement in the confines of PBAC. In future we investigate the feasibility of adapting our PBCA framework to Big Data access control.

REFERENCES

[1] Xuan Hung Le, Sungyoung Lee, Young-Koo Lee , Heejo Lee , Murad Khalid , Ravi Sankar . (2010). Activity-oriented access control to ubiquitous hospital information and services. *Elsevier*. 180, p.20-30.

[2] Maria Krotsiani, George Spanoudakis, Khaled Mahbub. (2013). Incremental Certification of Cloud Services. *Research paper*. p1-10.

[3] Luokai Hu, Chao Liang, Ying Lu, Yan Zeng. (2014). A defeasible policy based access control approach for semantic web services composition. *Research paper*. p1-7.

[4] Tejeddine Mouelhi, Donia El Kateb, Yves Le Traon. (2015). Inroads in Testing Access Control. *Elsevier*. p1-26.

[5] Yongzhi Wang and Jinpeng Wei. (2015). Toward Protecting Control Flow Confidentiality in Cloud-Based Computation. *Research paper*. p1-32.

[6] Canh Ngo , Yuri Demchenko, Cees de Laat. (2015). Multi-tenant attribute-based access control for cloud infrastructure services.*Elsevier*. p1-20.

[7] Rafael Teig~ao, CarlosMaziero , AltairSantin. (2011). Applying ausagecontrolmodelinanoperatingsystemkernel. *Elsevier*. p1-11.

[8] HuaWanga,LiliSuna,ElisaBertinob. (2014). Buildingaccesscontrolpolicymodelforprivacypreservingandtestingpolicyconflictingproblems.*Elsevier*. p1-11.

[9] Aliaksandr Lazousk, Fabio Martinelli, Paolo Mori. (2010). Usage control in computer security: A survey. *Elsevier*, p1-19.

[10] Ruixuan Li , Zhiyong Xub, Wanshang Kanga, Kin Choong Yowc, Cheng-Zhong Xuc. (2014). Efficient multi-keyword ranked query over encrypted data in cloud computing. *Elsevier*. 30, p1-9.

[11] ChingHsu. (2013). Extensible access control mark up language integrated with Semantic Web technologies. *Elsevier*. p1-19.

[12] Nicoletta Dess, Gabriele Milia, Emanuele Pascariello, Barbara Pes. (2015). COWB: A cloud-based framework supporting collaborative knowledge management within biomedical communities. *Research paper*. p1-40.

[13] Pietro Colomboa, Elena Ferraria. (2015). Privacy aware access control for Big Data: a research roadmap. *http://dx.doi.org/10.1016/j.bdr.2015.08.001*. p1-13.

[14] Marco Crasso , Cristian Mateos, Alejandro Zunino, Marcelo Campo. (2011). SWAM: A logic-based mobile agent programming language for the Semantic Web. *Elsevier*. p1-15.

[15] W. Li, C.Yang, D.Nebert, R.Raskin, P.Houser, H.Wua, Z.Li . (2011). Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure. *Elsevier*. p1-11.

[16] D. Gregor,S.L.Toral n, T.Ariza,F.Barrero. (2012). An ontology-based semantic service for cooperative urban equipments. *Elsevier*. p1-14.

[17] Andrzej Goscinski, Michael Brock. (2010). Toward dynamic and attribute based publication, discovery and selection for cloud computing. *Elsevier*. p1-24.

[18] Younis A. Younis, Kashif Kifayat, Madjid Merabti. (2014). An access control model for cloud computing. *Elsevier*. p1-16.

[19] Md. Whaiduzzaman , Mehdi Sookhak , AbdullahGani , Rajkumar Buyya b Q.(2013).A survey on vehicular cloudcomputing. *Elsevier*, p1370-1381.

[20] Pankaj Deep Kaur and Inderveer Chana. (2014). Cloud based intelligent system for deliveringhealth care as a service. *Elsevier*. p1-14.

[21] Haralambos Mouratidisa, Shareeful Islama, Christos Kalloniatisb, Stefanos Gritzalis. (2013). A framework to support selection of cloud providers based on security andprivacy requirements. *Elsevier*. p1-18.

[22] Saurabh Kumar Garg a, Steve Versteeg b, Rajkumar Buyyaa. (2013). A framework for ranking of cloud computing services. *Elsevier*. p1-12.

[23] Mehdi Sookhak a,n, HamidTalebian a, EjazAhmed a, AbdullahGani a, Muhammad KhurramKhan. (2014). A review on remote data auditing in single cloud server: Taxonomy and open issues. *Elsevier*. p1-20.

[24] Florin Pop, Radu-Ioan Tutueanu, Ciprian Barbieru. (2016). Adaptive Resource Allocation in Cloud Computing Based on Agreement Protocols. *Springer*. p1-21.

[25] Luis Rodero-Merino, Luis M. Vaqueroa, Victor Gil , Fermín Galán, Javier Fontán, Rubén S. Monteroc, Ignacio M. Llorente. (2010). From infrastructure delivery to service management in clouds.*Elsevier*. p1-15.

[26] Wenjuan Fan, Harry Perros. (2014). A novel trust management framework for multi-cloud environments 4 based on trust service providers. *Elsevier*. p1-15.

[27] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani and Rajkumar Buyya. (2013). Heterogeneity in Mobile Cloud Computing:

Taxonomy and Open Challenges. *IEEE*. p1-24.

[28] Peter Bloodsworth, Raihan Ur Rasoolb, Kamran Munir, Omer Rana. (2015). Cloud Market Maker: An automated dynamic pricing marketplace for cloud users. Barkha Javed a *Manuscript*. p1-50.

[29] Mohammad Shorfuzzaman, Abulhameed Alelaiwi, Mehedi Masud, Mohammad Mehedi Hassan, M. Shamim Hossain. (2014). Usability of a cloud-based collaborative learning framework to improve learners' experience. *Elsevier*. p1-10.

[30] Jing Du, Mohamed El-Gafy, Palden Lama . (2016). A Cloud-based shareable library of cooperative behaviors for Agent Based Modeling in construction. *Elsevier*. p1-12.

[31] Philip Churcha, Andrzej Goscinski, Christophe Lefèvre. (2015). Exposing HPC and sequential applications as services through the development and deployment of a SaaS cloud. *Elsevier*. p1-14.

[32] Gangyan Xu, George Q. Huang , Ji Fang. (2015). Cloud asset for urban flood control. *Elsevier*. p1-11.

[33] Indrajit Sinha and Milind Kumar Sharma. (2015). Cloud computing in small and medium sized enterprises: an architectural model.*Inderscience Enterprises Ltd.*. 6 (3), p1-22.

[34] Pei-Chi Chao, Hung-Min Sun . (2013). Multi-agent-based cloud utilization for the IT office-aid asset distribution chain: An empirical case study. *Elsevier*. p1-21.

[35] Dimitrios Kourtesis, Jose María Alvarez-Rodríguez, Iraklis Paraskakis. (2014). Semantic-based QoS management in cloud systems: Current status and future challenges. *Elsevier*. p1-17.