Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 16 Issue 1 Version 1.0 Year 2016
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

# Exploring Predicate based Access Control for Cloud Workflow Systems

By B. Srinivasa Rao & Dr. G. Appa Rao
*GITAM University, India*

*Abstract-* Authentication and authorization are the two crucial functions of any modern security and access control mechanisms. Authorization for controlling access to resources is a dynamic characteristic of a workflow system which is based on true business dynamics and access policies. Allowing or denying a user to gain access to a resource is the cornerstone for successful implementation of security and controlling paradigms. Role based and attribute based access control are the existing mechanisms widely used. As per these schemes, any user with given role or attribute respectively is granted applicable privileges to access a resource. There is third approach known as predicate based access control which is less explored. We intend to throw light on this as it provides more fine-grained control over resources besides being able to complement with existing approaches. In this paper we proposed a predicate-based access control mechanism that caters to the needs of cloud-based workflow systems.

*Index Terms:* *workflow systems, authorization, predicate based access control, fine-grained access control.*

*GJCST-B Classification :* *D.4.6 H.4.1*


*Strictly as per the compliance and regulations of:*

# Exploring Predicate based Access Control for Cloud Workflow Systems

B. Srinivasa Rao[α] & Dr. G. Appa Rao[σ]

*Abstract-* Authentication and authorization are the two crucial functions of any modern security and access control mechanisms. Authorization for controlling access to resources is a dynamic characteristic of a workflow system which is based on true business dynamics and access policies. Allowing or denying a user to gain access to a resource is the cornerstone for successful implementation of security and controlling paradigms. Role based and attribute based access control are the existing mechanisms widely used. As per these schemes, any user with given role or attribute respectively is granted applicable privileges to access a resource. There is third approach known as predicate based access control which is less explored. We intend to throw light on this as it provides more fine-grained control over resources besides being able to complement with existing approaches. In this paper we proposed a predicate-based access control mechanism that caters to the needs of cloud-based workflow systems. Enterprise wide business processes are executed in coordinated and controlled fashion with our comprehensive authorization and access control mechanism. This approach is based on analysis of application level resources and access policies for controlling users from accessing resources. This novel approach considers data content, users, processes, tasks, objects and roles thus making it a holistic approach in the application level access control. We built a prototype application to demonstrate the proof of concept. Our implementation of predicate based access control mechanism has shown more fine-grained control. We believe that it can be incorporated in real world workflow systems with diverse access control needs.

*Index Terms: workflow systems, authorization, predicate based access control, fine-grained access control.*

## I. Introduction

Users of an application play different role in an organization. Based on their role they have previleges to gain access to application resources. The role is convenient way in managing users in large scale and controlling access to resources in better way. Authorization is a term that refers to an information secuirty mechanism that deals with access rights in order to deny or authorize a user to access particular resource. This is based on access policies and the criticality of resources. Authorization is the part of overall computer or information security which is synonymous to real world thinking of humans with respect to access control. For instance a user in manager role is privileged to perform certain action and the same is denied to a user in clerk role. This is what reflects the real world though process that is captured greatly with access control mechanisms. After authentication of a user which deals with finding whether user is genuine (identity of user), authorization is crucial for controllling the authenitcated user in accessing resources. To reiterate, the process of denying or granting access to resources is known as authorization. Figure 1 shows overview of different authorization As can be seen in Figure 1, it is evident that the three models have different approaches in controlling access to resources. Stated differently, though resource is same, the users are controlled to access it differently. According to Jin [41] role based access control (RBAC) has its drawbacks as described here. Explosion of roles parameters, privileges makes it complex. It is difficult to design roles and managing them. It is cumbersome to grant/revoke privileges to/from roles. Making changes based on global or local factors is difficult. And RBAC does not support a custom extension to it. Attribute based access control (ABAC) overcomes these drawbacks and provides a flexible means of granting access rights through attributes. Here attribute is a key/value pair. However, it can be a set of key/value pairs to which access rights can be granted to authorized users. Access implications when user's attributes are changed and reaching consensus on the meaning of attributes are the drawbacks in ABAC as discussed in [42].mechanism.

*Author α: Assistant Professor, CSE Department GIT, GITAM University Visakhapatnam. e-mail: ugandarsrinu@gmail.com*
*Author σ: Professor, CSE Department GIT, GITAM University Visakhapatnam. e-mail: apparao_999@yahoo.com*

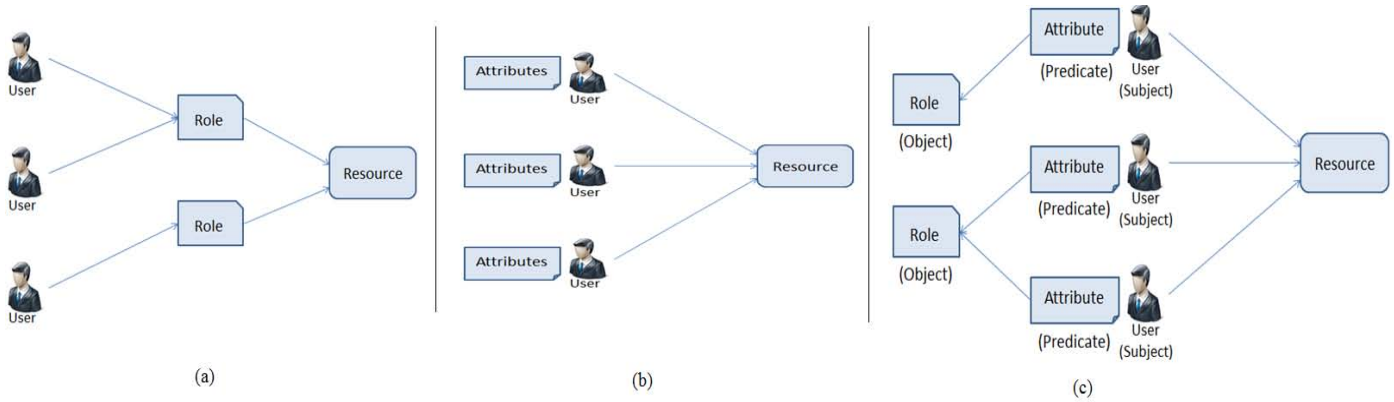


*Figure 1:* Overview of access control models. (a) Role-based (b) Attribute based (c) Predicate based

The third approach which is less explored is predicate based access control (PBAC) which can simplify the access control further besides complementing the other mechanisms. In other words, it can have synergic advantages of the other two access control mechanisms. In this paper we explore PBAC with cloud-based workflow systems. Table 1 show acronyms used in the paper. Our contributions in this paper include the design and implementation of PBAC mechanism with a case study. This research paves way

*Table 1:* Acronyms

| Acronym | Description |
|---|---|
| IaaS | Infrastructure as a Service |
| PaaS | Platform as a Service |
| SaaS | Software as a Service |
| AaaS | Authorization as a Service |
| SOA | Service Oriented Architecture |
| DDoS | Distributed Denial of Service |
| ACaaS | Access Control as a Service |
| CP-ABE | Cipher text-Policy Attribute Based Encryption |
| ABE | Attributed-Based Encryption |
| ABE-AL | Attributed-Based Encryption with Attribute Lattice |
| XML | Extensible Mark up Language |
| CRiBAC | Community Centric Role Interaction Based Access Control |
| OSN | Online Social Network |
| RBAC | Role Based Access Control |
| ABAC | Attributed Based Access Control |
| PBAC | Predicate Based Access Control |

for exploring PBAC for workflow systems of different domains. The remainder of the paper is structured as follows. Section II reviews related literature. Section III provides details of the proposed PBAC in detail. Section IV provides results of the research while section V concludes the paper and gives directions for future work.

## II. Related Works

This section reviews literature on different kinds of authentication systems such as role based authentication, attribute based authentication and predicate based authentication. Leandro *et al*. [1] proposed a multi-tenancy authorization system for cloud computing. It is based on Shibboleth without using a trusted third party. Similar kind of work is done in [2], [14] for cloud architectures. Reeja [3] focused on co-operative secondary authorization that is a method of role based access control mechanism with a recycling approach. Khalid *et al*. [4] proposed a protocol for authorization and authentication for cloud that supports anonymous communication. Birgisson *et al*. [5] employed cookies with contextual caveats for authorization in cloud. This mechanism is decentralized in nature with delegation of principals. Gonzalez *et al*. [6] credentials based authorization and authentication for cloud computing. Continuous authorization re-evaluation method is proposed by Marcon *et al*. [7]. Lang [8] proposed authorization as a service (AaaS) for cloud computing and Service Oriented Architecture (SOA) applications for reliable security. Chen *et al*. [9] proposed authentication mechanisms for high quality applications that deal with multimedia.

Zareapoor *et al*. [10] focused on data security model for safe cloud. Kumar and Sharma [11] proposed mechanisms for protecting cloud systems from Distributed Denial of Service (DDoS) attacks. Ryoo *et al*. [12] focused on secure mechanisms in cloud with auditing services. Masood *et al*. [13] proposed an access control framework for cloud computing. They proposed a service layer for cloud known as Access Control as a Service (ACaaS). This is a generic solution for authenticaiotn and authorization. Zhu and Gong [15] proposed fuzzy authorization scheme based on Cipher text-Policy Attribute Based Encryption (CP-ABE). It works fine with multiple clouds besides enabling fuzziness in authorization. For multi-platform clouds an authorization frameworks is proposed in [16]. Rather

and Vida [17] proposed two-step authentication for cloud which is based on de-duplication which ensures privacy and integrity of data. Akimbo *et al*. [18] focused on securing PaaS layer of cloud. Other authorization and authentication schemes can be found in [19] and [20].

Other mechanisms found in the literature include identity based encryption [21] and other mechanisms as described here. Popa *et al*. [22] proposed Cloud Policy for access control in cloud which is hypervisor based and proved to be robust. Ruj *et al*. [23], [26] proposed a privacy preserving mechanism for access control in a decentralized fashion. She *et al*. [24] proposed a rule bsed information flow control for cloud with fine-grained access control. Zhu and Ma [25] proposed a role based access control for cloud that exploits Attributed-Based Encryption with Attribute Lattice (ABE-AL). Sun *et al*. [27] presented multi-keyword text search with secure authentication and authorization. Sun and Wang [28] focused on purpose-based access control for XML databases. Bauer *et al*. [29] proposed logic-based access control with credentials and constraints for robust security. Similar work was done in [34]. Tu *et al*. [30] proposed a fine-grained access control mechanism which also supports revocation of credentials. Ababneh *et al*. [31] focused on the policy – based dialog for protecting systems with physical access control.

Jung and Joshi [36] proposed Community Centric Property Based Access Control (CPBAC) which is an extension to Community Centric Role Interaction Based Access Control (CRiBAC) for Online Social Networks (OSNs). Service Level Agreement (SLA) based security risk analysis is explored in [37]. Dara [38] explored cryptography challenges in cloud. Jana and Bandyopadhyay [39] explored controlled privacy in mobile cloud for protecting system from different threats. Yadav and Wanjari [40] proposed an authentication mechanism for smart grid besides exploring its secure access to smart grid in real time environment. In this paper our focus is on the predicate based access control mechanisms for improved security in cloud.

## III. Predicate based Access Control Mechanism

In this section we provide a generic framework that can be used for any workflow system. Any workflow system needs data to be captured and protected besides giving controlled access to its legitimate users. Instead of giving a domain-specific solution, we provide a generic framework that can be adapted to different application domains. There are certain things common across domains. This is the basis for the generic framework. Every workflow system has to deal with data. Therefore the central point of discussion is the record or tuple that needs to be given controlled access to users. Therefore we considered the record or tuple as basis to which many aspects are associated with. The record is a master record that might have associated tuples in different relations based on the transactions made. However, the master record is very important as it does not generally subjected to frequent changes. Figure 2 shows the generic framework that is further extended in Figure 3. The framework shows different aspects such as instance-based user-group, task-based privileges, privilege propagation, role, instance-based predicate and dynamic authorization. All these aspects are related to the record or tuple with respect to access control.

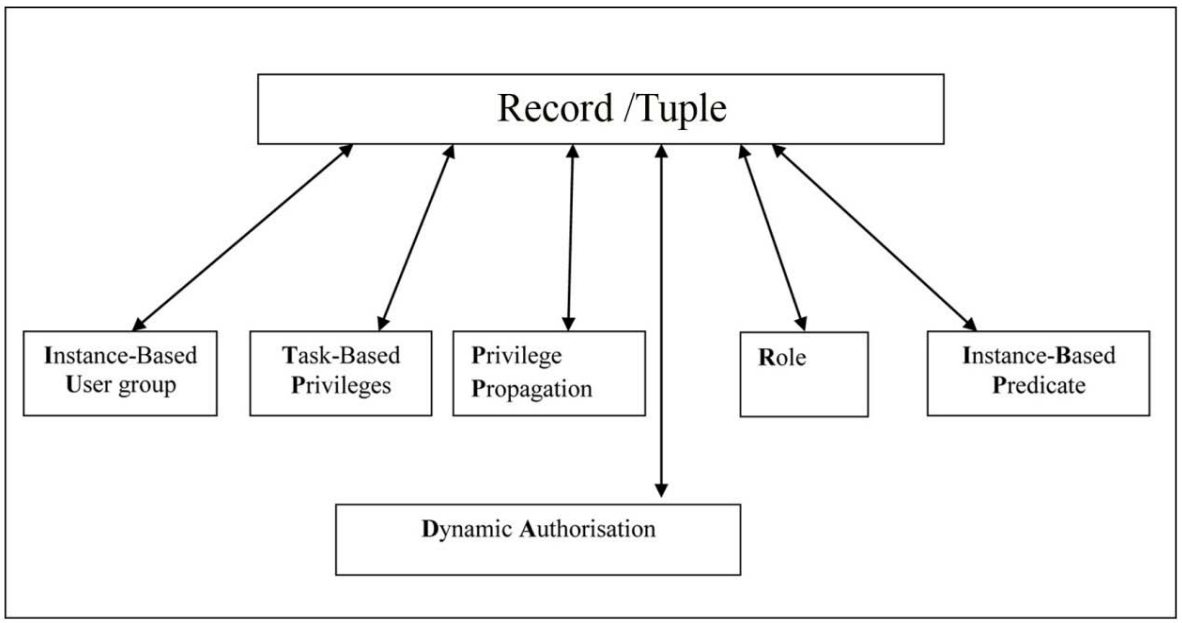


*Figure 2 :* Generic framework required for predicate-based access control model

*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 a 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.

## IV. Components of Access Control Model

Predicate based access control model, we presented in this paper is generic in nature and can be adapted to different domains with required changes. Apart from the aspects associated with master tuple shown in Figure 2, there are five components associated with predicate-based access control model. They are subject, task, object, constraint and privilege. These components are used with certain notations to have a comprehensive predicate-based access control model.

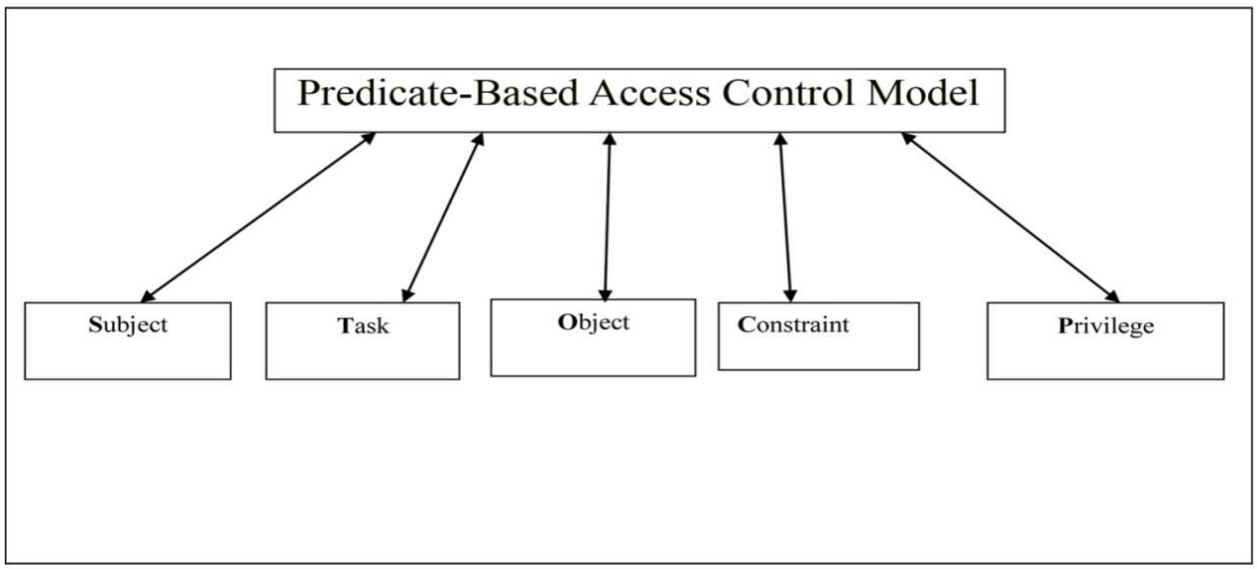


*Figure 3 :* Components of predicate-based access control model

Prior to describing the components, les us discuss some of the important notations used. A runtime instance is nothing but the ID of master record and its associated data. Different master records are distinguished by using unique ID. The state of runtime instance is represented using some variables. They are presented in Table 2.

*Table 2 :* Important system variables of the model

| Variable | Description |
|---|---|
| #This.ID | It represents current runtime instance of master record. It is the instance to which user is associated with. |
| #This.TaskName | It denotes the current task being performed by an authorized user. |
| #This.RoleName | It represents the role name to which the authorized user belongs. |
| #This.UserID | It represents the unique ID of the user who accesses runtime instance of master record. |

Apart from these variables which can be called as system variables, designers of application can create domain specific variables. These variables are accessible throughout the workflow system.

*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. In an organization, roles are hierarchically organized as shown in Figure 5. **R** denotes a set of roles. .

$\mathbf{R} = \mathbf{r}_i$ $(1 \le i \le n)$ and $<_R$
$\mathbf{r}_i, \mathbf{r}_j \in \mathbf{R}$
$\mathbf{r}_i$ precedes $\mathbf{r}_j$ in the hierarchy ($\mathbf{r}_i <_R \mathbf{r}_j$)

The runtime instance based user group denotes a set of users (individuals) who were involved when the master tuple is created. For instance in a health care workflow system (case study is given in the subsequent section) a patient is served by Doctor, Nurse, and Receptionist. In this case these three users are known as runtime instance based user group. And these three should be able to access the record as per privileges and roles. There is many to many relationship between users and roles. And the instance user group is dynamic and new users may be included at runtime.

*Task:* The task is a component. A set of components of workflow is represented as a tree. An example is shown in Figure 7. Let **T** represent set of tasks.

$\mathbf{t}_i, \mathbf{t}_j \in \mathbf{T}$
$\mathbf{t}_i$ includes $\mathbf{t}_j$ in the hierarchy ($\mathbf{t}_i <_T \mathbf{t}_j$) if $\mathbf{t}_i$ has a subtask $\mathbf{t}_j$

*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 diversifie set of files of different kinds such as audio, video, .exe, instance of Java classes, a relation instance, a database, set of relations and so on. **O** represents set of objects.

$\mathbf{O} = \{o_1, o_2, \ldots, o_n\}$
For every $\mathbf{o} \in \mathbf{O}$ set of security attributes are defined **security-attri**(**o**)
For each object $\mathbf{o} \in \mathbf{O}$ object represents data of different domains like **outside, historical and current**

The data generated by the current runtime instance of record can be of two types such as current and historical. Historical refers to the past runtime instance of the same kind produced data. Current refers to the data produced by the current runtime instance of the master record. Outside indicates that the data comes from outside of the workflow process to which the predicate based access control is employed.

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

<Boolean-expression> ::= <condition1> {OR <condition2>}
<condition > ::= < predicate1> {AND <predicate2>}
< predicate> ::= <left-value> <operator> <right-value>
<left-value> ::= <security-attribute-variable>
<right-value> ::= <constant> | <workflow-system-environment-variable>
| < security-attribute-variable >
Possible operators are:
<operator> ::= '=' | '!=' | '<>' | '>' | '<' | '>=' | '<='
**rel(c)** represents all objects whose security attributes **security-attri**(**o**) are part of the constraint c
A constraint is valid if it holds true for the following conditions:
(a) $\exists \mathbf{o}\ (\mathbf{o} \in \mathbf{O} \wedge \mathbf{o} \in \mathbf{rel}(\mathbf{c}))$
(b) $\neg\exists(o_1, o_2)\ (o_1 \in \mathbf{O} \wedge o_2 \in \mathbf{O} \wedge o_1 \neq o_2 \wedge \{o_1, o_2\} \subseteq \mathbf{rel}(\mathbf{c}))$
In any constraint $\mathbf{c} \in \mathbf{C}$, only one object's security attributes should appear

*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.

## V. Case Study – Health Care Work Flow System

Cloud computing has emerged as a new model of computing which provides pool of computing resources in pay as you use fashion. Any cloud based workflow system (or even without cloud) can make use of the proposed predicated based access control model. Figure 4 shows a general work flow of the health care system. Many details are not considered for making it simple. However the flow can provide required functionalities that can be used to demonstrate the access control mechanisms.

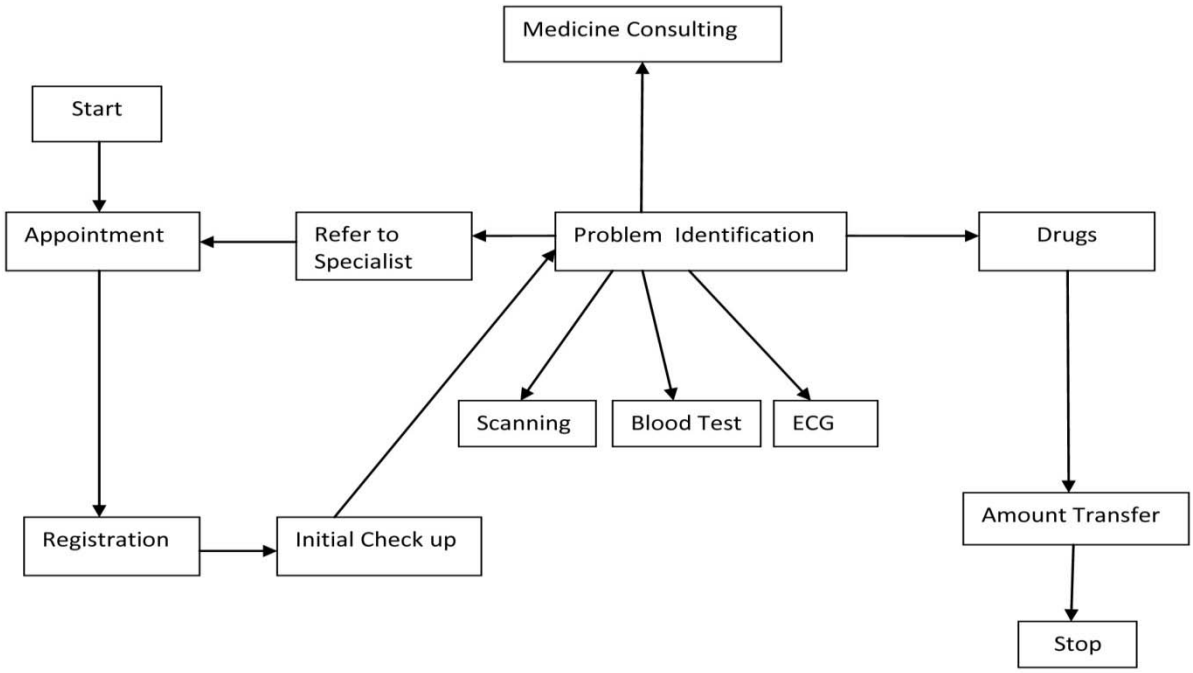


*Figure 4 :* General work flow of a healthcare system

As shown in Figure 4, the flow starts with an appointment. On requesting appointment registration of the patient is completed. Then health service provider will check for any symptoms or temperature, blood pressure and so on in order to identify the problem. Sometimes, it is possible that investigation is made with different tests and problem is identified.

Once the problem is identified either medicine is prescribed or referred to a specialist doctor. After taking medicine, the patient will pay money. This is the flow which actually reflects a typical, though not elaborate, scenario in every healthcare unit.

## VI. Roles in the Health Care System

The roles in any workflow system are hierarchical in nature. Healthcare system is no exception. It has many roles and some roles depend on other roles. Figure 5 shows roles in hierarchical fashion.

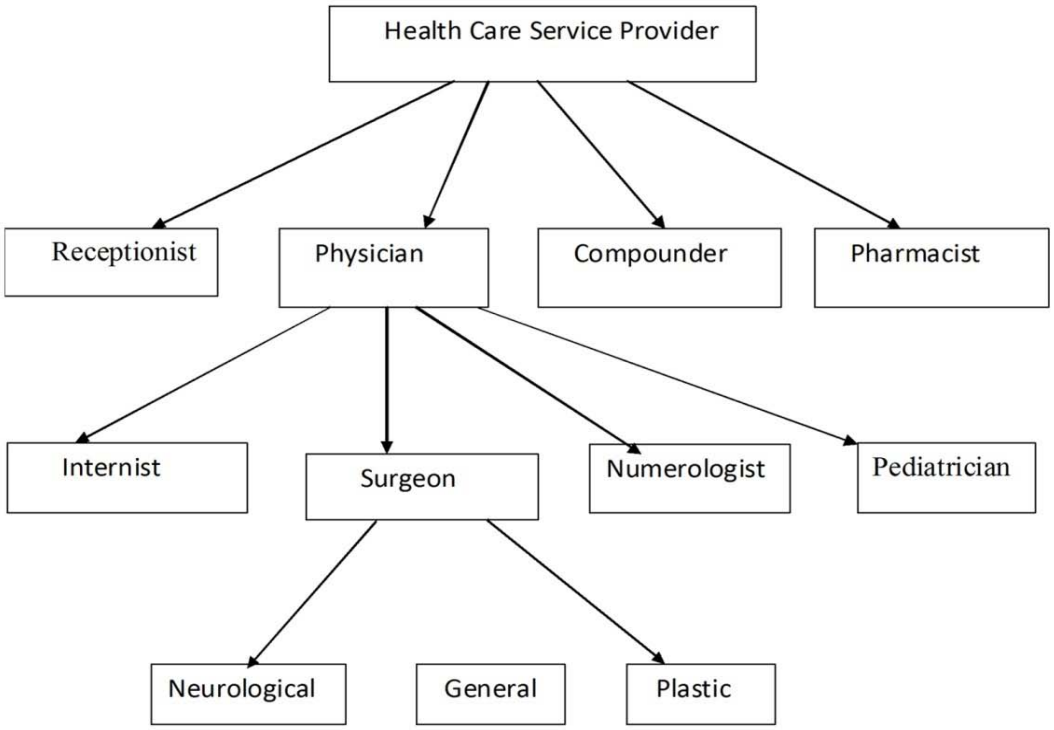


*Figure 5 :* Roles in healthcare workflow

As shown in Figure 5, the roles include receptionist, physician, compounder or nurse, and pharmacist. The physician role can have sub roles such as internist, surgeon, numerologist, and paediatrician. Again the surgeon role has sub roles such as neurological surgeon, general surgeon and plastic surgeon. These roles are used in the access control system to have controlled access to various stakeholders of the system.

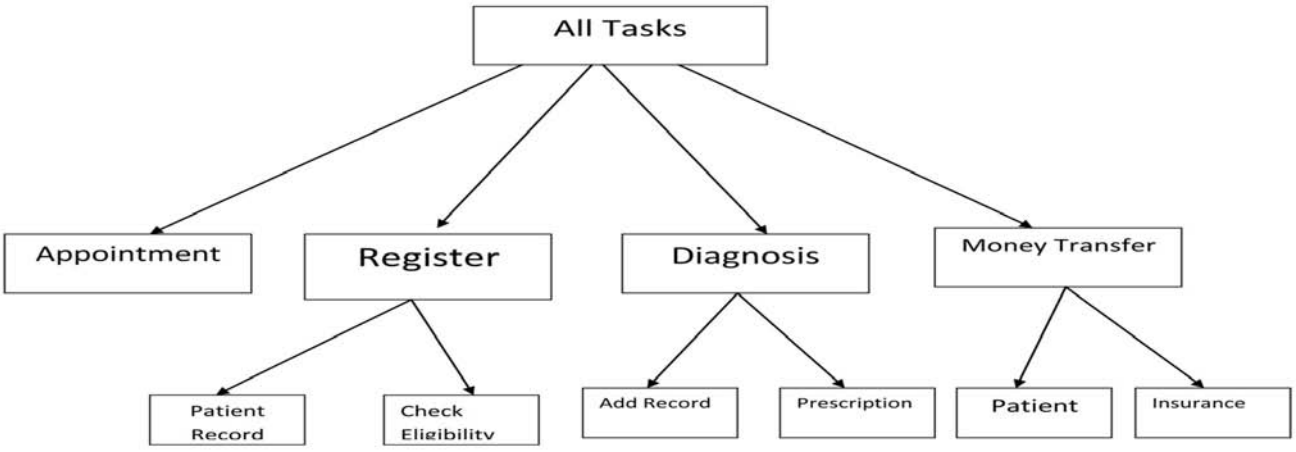


*Figure 6 :* Tasks hierarchy involved in healthcare workflow system (some tasks omitted to simplify the workflow)

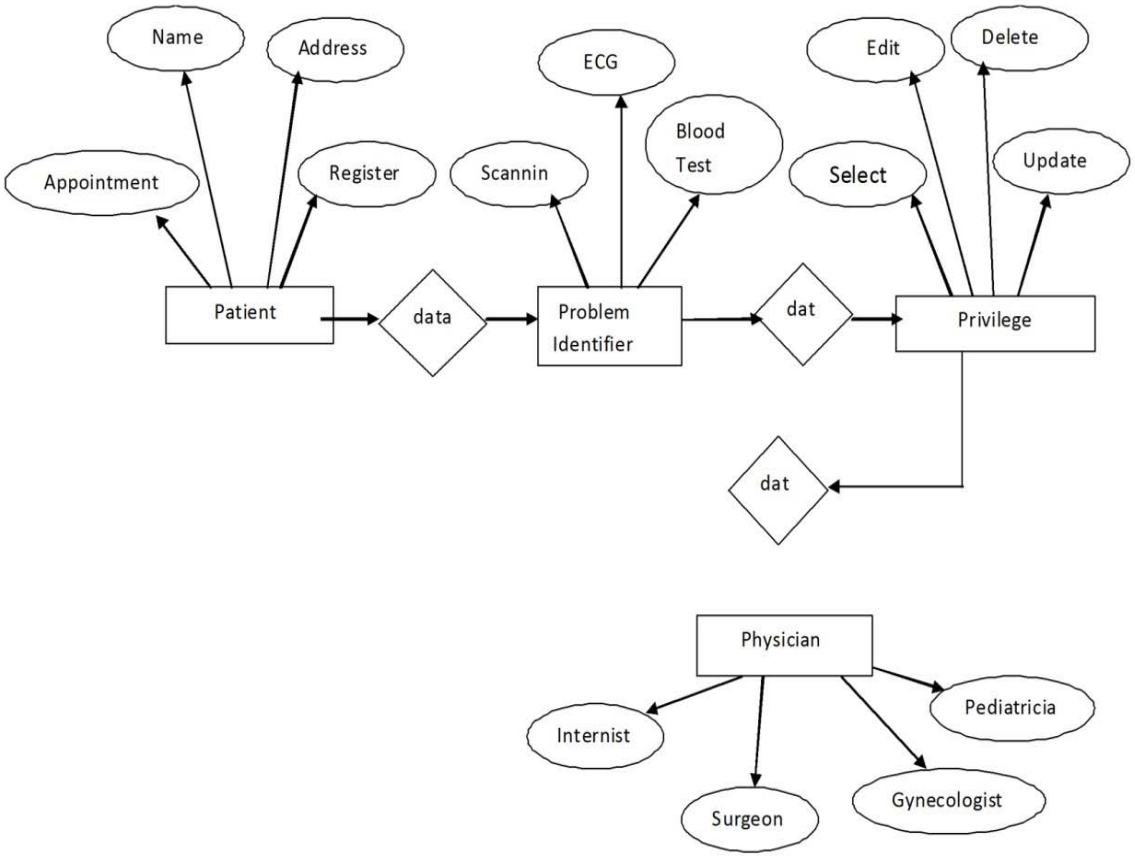


*Figure 7 :* Entity relationship diagram for healthcare workflow system (with simplified relations)

As shown in Figure 7, the workflow repository contains many entities and attributes. These entities, attributes and relationships are mapped to related tables in relational database. Patient, problem identifier, privilege and physician are the entities with different attributes involved. The repository is not completely provided and the cardinality is not shown in the diagram.

As shown in Figure 6, there are many tasks involved in the healthcare system. The main tasks considered are appointment, registration, diagnosis, and money transfer. The registration process contains two sub tasks such as patient record, checking eligibility. Diagnosis has two sub tasks such as adding record and prescription. Money transfer has two sub tasks such as one related to patient and other one related to insurance.

## VII. Access Control Model Employed to Healthcare Workflow System

The following components and relationships are considered to have a formal access control system for the healthcare workflow system.

U, R, O, T, C, P represent User, Role, Task, Object, Constraint and Privilege

RoleHierarchy $\subseteq R \times R$ represents partial order on R representing relationship known as role dominance $<_R$

TaskTree $\subseteq T \times T$ represents partial order on T representing relationship known as task inclusion relationship $<_T$

UserRoleAssignment $\subseteq U \times R$ represents assignment of user to role with many to many relationship

RoleTaskAssignment $\subseteq R \times T$ represents authorization of role to task with many to many relationship

$o \in O$ can be of historical, current or outside

ObjectPrivilege $\subseteq O \times P$ represents possession of object to privilege with many to many relationship

PermissionAssignment1 $\subseteq$ RoleTaskAssignment$\times$ObjectPrivilege$\times$C represents permission relationship role to task and task to object and access privilege could be select, read, delete, edit, update and destroy

PermissionAssignment2 $\subseteq$ RoleTaskAssignment$\times$ObjectPrivilege represent permission relationship from role to task and task to object and access privilege could be select, read, delete, edit, update and destroy

| Authorization Rule | Description |
|---|---|
| p = insert (new) | All users of given role can insert new records into a relation or create a new document. |
| p = select (read), or update (edit), or delete (destroy) | All users who play given role an perform the privileged operations on either database or files. |
| Role authorization | Role authorization propagates to all roles that precede **r** in the role hierarchy. |
| Task authorization | Authorization to role on task propagates to all sub tasks as well. |

*Authorization Rules* This sub section provides different authorization rules and description of them. Here an authorization can be considered to be a 4-tuple or 5-tuple. (r, **t**, **o**, **p**) is a 4-tuple representation indicating the user of given role can perform given task on specified object with given privileges. The 5-tuple representation (**r**, **t**, **o**, **p, c**) is similar to that of 4-tuple except the fact that it supports constraints as well.

*Some Examples*

Select ID(**o**) From **meta-object** (**o**) Where **c'** and (**$c_1$** or **$c_2$** or … or **$c_n$**);

In this query o is either a relational table or set of files that can be used to retrieve data. Here c' represents either privilege propagation or runtime-instance based access control based on the runtime situations. The union of privileges is used based on the constraints given for authorized access to the data. Once query operation is finished, the object IDs that satisfy predicate based access control are retrieved. Then further processing carried out. If the o belongs to a relation, join operation can be used to combine results. If not name and category of files can be used. Even if the o is a special data, that external interface is invoked to access it. Data can be migrated from current domain to historical domain. The object o' is used to represent historical object. The following operations complete the migration process.

```
Select *
From o Where ID = #This.InstanceID;
Delete from o where ID = #This.InstanceID;
```

## VIII. Experimental Results

We built a prototype application that caters to the needs of a healthcare workflow system. Then we applied the predicate based access control which combines the features of roles and attributes as well and obtains synergic effect in controlling access to application resources. The application has proved to be useful for the real world applications as it was able to provide controlled access with high flexibility and utility. The results of application with respect to the attributes, constraints and are presented here.

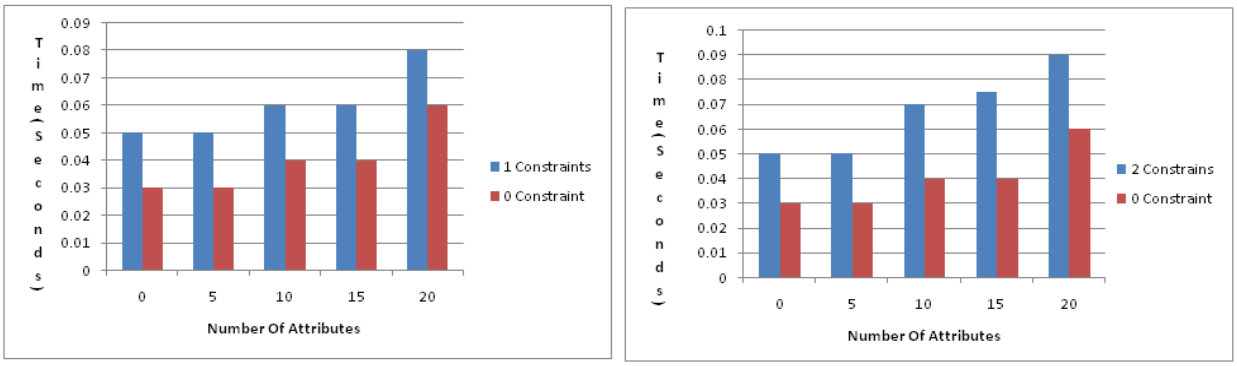


*Figure 8 :* Shows the time taken when 1 and 2 constraints are used

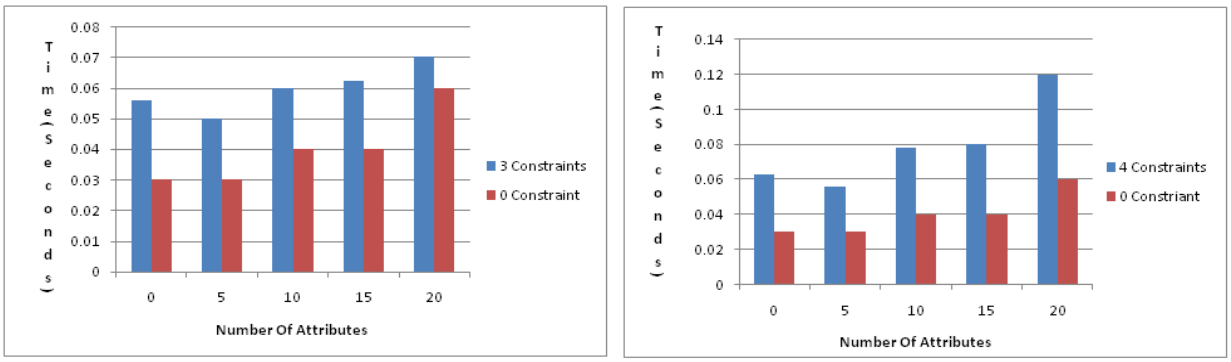


*Figure 9 :* Shows the time taken when 3 and 4 constraints are used

As can be seen in Figure 8 and Figure 9, it is evident that the horizontal axis represents number of attributes while the vertical axis represents the time taken. The results reveal the difference in time when constraints are applied while performing the proposed access control mechanisms.

## IX. Conclusions and Future Work

In this paper, we studied different kinds of access control mechanisms. We found that there are two widely used access control mechanisms. They are RBAC and ABAC. The RBAC depends on the roles that represent set of privileges that can be assigned to users who belong to the role. RBAC has its drawbacks as described here. Explosion of roles parameters, privileges makes it complex. It is difficult to design roles and managing them. It is cumbersome to grant/revoke privileges to/from roles. Making changes based on global or local factors is difficult. And RBAC does not support a custom extension to it [41]. Access implications when user's attributes are changed and reaching consensus on the meaning of attributes are the drawbacks in ABAC [42]. We focused on the third alternative known as predicated based access control model which can also complement to the features of role and attributed based models. We proposed a generic model for predicate based access control that can be applied to any workflow system including cloud based workflow systems. Afterwards we applied the model to a case study "healthcare workflow system". We built a prototype application to demonstrate the proof of concept. The empirical results revealed that the proposed application is flexible and effective in controlling access to application resources. In future we intend to improve the PBAC and adapt it to different workflow systems.

## References References Referencias


1. Marcos A. P. Leandro, Tiago J. Nascimento, Daniel R. dos Santos, Carla M. Westphall, Carlos B. Westphall. (2012). Multi-Tenancy Authorization System with Federated Identity for Cloud-Based Environments Using Shibboleth. *The Eleventh International Conference on Networks*, p.32-44.
2. Jorge Bernal Bernabe a, Juan M. Marin Perez b, Jose M. Alcaraz Calero b, Felix J. Garcia Clementec, Gregorio Martinez Perez a, Antonio F. Gomez Skarmetaa. (2012). Semantic-aware multi-tenancy authorization system for cloud architectures. *ELsevier*, p.213-313.
3. Reeja S L. (2012). Role Based Access Control Mechanism In Cloud Computing Using Co - Operative Secondary Authorization Recycling Method. *International Journal of Emerging Technology and Advanced Engineering*. 2 (10), p.25-34.
4. Umer Khalida, Abdul Ghafoor, Misbah Irum, Muhammad Awais Shibli. (2013). Cloud based Secure and Privacy Enhanced Authentication & Authorization Protocol. *ELsevier*. 22, p.32-44.
5. Arnar Birgisson,Joe Gibbs Politz,U lfar Erlingsson, Ankur Taly. (2014). Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud. *ACM*, p.56-60.
6. Nelson Mimura Gonzalez∗, Marco Antônio Torrez Rojas, Marcos Vinícius Maciel da Silva, Fernando Redígolo, Tereza Cristina Melo de Brito Carvalho∗, Charles Christian Miers†, Mats Näslund‡ and Abu Sho. (2013). A framework for authentication and authorization credentials in cloud computing. *IEEE*, p.213-313.
7. Arlindo Luis Marcon Jr., Altair Olivo Santin,Maicon Stihler, and Juliana Bachtold Jr. (2014). A UCONABC Resilient Authorization Evaluation for Cloud Computing. *IEEE*. 25 (2), p.12-17.
8. Ulrich Lang. (2010). OpenPMF SCaaS: Authorization as a Service for Cloud & SOA Applications. *IEEE*, p.56-60.
9. Jiann-Liang Chenz, Szu-Lin Wuy, Yanuarius Teofilus Larosa, Pei-Jia Yang, and Yang-Fang Li. (2011). IMS Cloud Computing Architecture for High-Quality Multimedia Applications. *IEEE*, p.25-34.
10. Masoumeh Zareapoor, Pourya Shamsolmoali, and M. Afshar Alam. (2014). Establishing Safe Cloud: Ensuring Data Security and Performance Evaluation. *International Journal of Electronics and Information Engineering*. 1 (2), p.32-44.
11. Naresh Kumar,Shalini Sharma. (2013). Study of Intrusion Detection System for DDoS Attacks in Cloud Computing. *IEEE*, p.12-17.
12. Jungwoo Ryoo , Syed Rizvi , William Aiken , and John Kissell . (2013). Cloud Security Auditing: Challenges and Emerging Approaches. *IEEE*, p.213-313.
13. Rahat MASOOD, Muhammad Awais SHIBLI, Yumna GHAZI, Ayesha KANWAL, Arshad ALI. (2014). Cloud authorization: exploring techniques and approach towards effective access control framewo rk. *Springer-Verlag Berlin Heidelberg*, p.56-60.
14. Zhaohai Zhang, Qiaoyan Wen. (2012). AN AUTHORIZATION MODEL FOR MULTI-TENANCY SERVICES IN CLOUD. *IEEE*, p.23-33.
15. Shasha Zhu and Guang Gong. (2013). Fuzzy Authorization for Cloud Storage. *IEEE*, p.213-313.
16. Primož Cigoj & Borka Jerman Blaži(2015). An Authentication and Authorization Solution for a Multiplatform Cloud Environment. *Information Security Journal: A Global Perspective*, p.80-90.
17. Manohar Vasantrao Rathod, Prof.S.G.Vaidya. (2015). Two-step authentication with data de-duplication in Cloud. *International Journal of*

*Advanced Research in Computer Engineering & Technology*. 4 (4), p.56-60.

18. A. Akinbi, E. Pereira, C. Beaumont. (2013). Identifying Security Methods and Controls for Secure PaaS Cloud Environments. *IEEE*, p.12-17.
19. Jannu. Prasanna Krishna, Macha. Ganesh Kumar. (2015). An Authorized Duplicate Check Scheme for Removing Duplicate Copies of Repeating Data in The Cloud Environment to Reduce Amount of Storage Space. *International Journal & Magazine of Engineering, Technology, Management and Research*. 2 (4), p.760–769.
20. Alexander Stanik, Patrick Bittner, Marvin Byfield, Fridtjof Sander, Daniel Sch¨oder. (2013). Local Authentication and Authorization System for Immediate Setup of Cloud Environments. *IEEE*, p.32-44.
21. Xuan Hung Le, Terry Doll, Monica Barbosu, Amneris Luque, Dongwen Wang. (2013). Evaluation of an Enhanced Role-Based Access Control model to manage information access in collaborative processes for a statewide clinical education program. *ELsevier*, p.213-313.
22. Lucian Popa, Minlan Yu, Steven Y. Ko. (2010). Cloud Police: Taking Access Control out of the Network. *ACM*, p.56-60.
23. Sushmita Ruj, Milos Stojmenovic, Amiya Nayak. (2012). Privacy Preserving Access Control with Authentication for Securing Data in Clouds. *IEEE*, p.23-33.
24. Wei She, I-Ling Yen, Bhavani Thuraisingham,. (2011). Rule-Based Run-Time Information Flow Control in Service Cloud. *IEEE*, p.32-44.
25. Yan Zhu, Changjun Hu, Di Ma, Jin Li. (2013). How to Use Attribute-Based Encryption to Implement Role-based Access Control in the Cloud. *ACM*, p.23-33.
26. Sushmita Ruj, Milos Stojmenovic, Amiya Nayak. (2014). Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds. *IEEE*. 25 (2), p.25-34.
27. Wenhai Sun, Bing Wang,Ning Cao, Ming Li, Wenjing Lou, Y. Thomas Hou. (2013). Privacy-preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking. *ACM*, p.12-17.
28. Lili Sun and Hua Wang. (2011). A purpose-based access control in native XML databases. *John Wiley & Sons, Ltd*, p.56-60.
29. Lujo Bauer, Limin Jia,Divya Sharma. (2010). Constraining Credential Usage in Logic-Based Access Control. *IEEE*, p.32-44.
30. Shan-shan Tu, Shao-zhang Niu and Hui Li. (2012). A fine-grained access control and revocation scheme on clouds. *John Wiley & Sons, Ltd*, p.23-33.
31. Mohammad Ababneh,Duminda Wijesekera,James Bret Michael. (2012). A Policy-Based Dialogue System for Physical Access Control. *IEEE*, p.56-60.
32. Mohammad Ababneh, Duminda Wijesekera, James Bret Michael. (2012). A Policy-Based Dialogue System for Physical Access Control. *IEEE*, p.23-33.
33. Rafael Teigao, Carlos Maziero, Altair Santin. (2011). Applying a usage control model in an operating system kernel. *Elsevier*. 24, p.213-313.
34. Lujo Bauer, Limin Jia, Divya Sharma. (2010). Constraining Credential Usage in Logic-Based Access Control. *IEEE*, p.23-33.
35. Md. Fakhrul Alam Onik, Syed Sabir Salman-Al-Musawi, Khairul Anam, Nafiul Rashid. (2012). A Secured Cloud based Health Care Data Management System. *International Journal of Computer Applications*. 49 (12), p.12-17.
36. Youna Jung a, James B.D. Joshi. (2014). CPBAC: Property-based access control model for secure cooperation in online social networks. *ELsevier*. 41, p.32-44.
37. Matthew L. Hale Rose Gamble. (2012). Sec Agreement: Advancing Security Risk Calculations in Cloud Services. *IEEE*, p.213-313.
38. Sashank Dara. (2013). Cryptography Challenges for Computational Privacy in Public Clouds. *ACM*, p.23-33.
39. Debasish Jana, Debasis Bandyopadhyay. (2015). Controlled Privacy in Mobile Cloud. *IEEE*, p.213-313.
40. Nageshwar Dev yadav,Prof. Akash Wanjari. (2014). Cloud Based Smart Metering Security Access and Monitoring System in the Real Time Environment. *International Journal of Engineering Research & Technology*. 3 (2), p.56-60
41. Xin Jin (2014). Attribute Based Access Control and Implementation in Infrastructure as a Service Cloud. Accessible at: http://www.profsandhu.com/dissert /Dissertation_Xin_Jin.pdf
42. Alan H. Karp, Harry Haury, Michael H. Davis (2009). From ABAC to ZBAC: The Evolution of Access Control Models. Accessible at: http://www. hpl.hp. com/techreports/2009/HPL-2009-30.pdf

This page is intentionally left blank