Extending XACML to support Credential Based Hybrid Access Control

Nirmal Dagdee[1] and Ruchi Vijaywargiya[2]

**[1] S D Bansal College Of Technology,
Indore, MP, INDIA**

**[2] Department of Computer Science and Engineering, S D Bansal College Of Technology,
Indore, MP, INDIA**

Abstract

Various research efforts are in progress to enforce credential based access control using XACML standard. The current standard of XACML supports attribute based access control [4,5,9,19]. While XACML accepts certified attributes through digital certificates, it does not support credential based access control in which the access conditions are defined not only in terms of credential attributes but also in terms of types of credentials. Credential based hybrid access control[7,11,14,20,21] has been proposed for systems having diversified access control requirements. The use of various types of credentials in access control policy specification provides easy and immediate access to unknown user in open access environment. Fine grained access control in closed administrative domain is achieved using Identity Credential and the attributes associated with the credentials. In this paper, we propose extensions to the XACML standard that support credential-based hybrid access control. The XACML access policy language has been extended to define access policy in terms of heterogeneous credentials. Each credential is uniquely identified by associating a *category* and *type* with it. The access policy contains various conditions over credentials and the attributes associated with the credentials. Enhancement to XACML framework has also been proposed so that credential based hybrid access policies can be evaluated and enforced.
.

Keywords: *XACML, credential, access control.*

1. Introduction

With the development of digital certificate technology, credential based approaches are becoming popular for enforcing security and access control on shared resources. In credential based access control systems[1,2,3,4,6,8,9,12], the access decision depends on the properties possessed by the user and can prove by submitting one or more credentials. The user submits a set of credentials along with the data access request. Credential based hybrid access control approach [7,11,14,20,21] has been proposed for the environment having varied access control requirements. It provides easy and immediate access to unknown user in open access environment along with fine grained access control in closed administrative domain. The access policy is defined in terms of various types of credentials. Various types of credentials along with attributes associated with credentials, data sources and the environment affect the access decision.

XACML is an OASIS standard that provides a formal representation model of access control policies and mechanisms for their evaluation[15,16,19]. It defines a XML-based language for specifying and exchanging access control policies over the web. The XACML framework describes a modular architecture for the evaluation of policies and a communication protocol for message interchange between various modules. XACML includes standard extension points for the definition of new functions and data types that provide a great potential for enhancing the language for customized needs[13,15,16]. Although XACML enjoys a large adoption in industry due to its simplicity and powerful extension mechanism, it has few limitations[15,16,18,19]. The current XACML standard enforces attribute-based access control[4,5,9,19] in which the authorizations applicable to a user depends on the attributes that the user presents and the conditions that the user satisfies. While the XACML permits that the attributes can be presented by means of certificates and can accept certified attributes but this is limited however to conditions on attributes associated with the credentials[15,16,18,19]. The real support for credential based access control requires expressing the access control conditions not only on values of attributes associated with the credentials but also on the credentials themselves. Intuitively, XACML supports attribute-based access control but does not really support credential-based access control[19]. In this paper, we propose extensions to XACML framework so that it can support credential based hybrid access control.

The rest of this paper is organized as follows. Section 2 highlights the credential based hybrid access control and the XACML standard framework has been discussed in section 3. The proposed extension to XACML framework has been described in section 4. Few illustrative usecases from the medical domain are discussed in section 5. In section 6, we have summarized the salient advantages of the proposed work.

2. Credential based hybrid access control

Various credential based access control approaches [1,2,3,4,6,8,9,12,16,17] have been proposed that make use of digital credentials for authentication and authorization. For systems having varied access control requirements, credential based hybrid access control[7,11,14,20,21] has been proposed. This approach not only enables immediate and open access to shared information by competent users but also provides fine grained access control on domain confined data. The access control policy is defined in terms of heterogeneous credentials. Use of Identity credentials enable fine grained user specific accesses to known users. Attribute certificates are found suitable in open access environment where the resource provider and the resource requester are not known to each other and the access decision depends on the properties possessed by the user and can prove by submitting one or more attribute credentials. Recently, a new type of credential called as Standard credential[11, 14, 22] has also been proposed. Standard credentials are general purpose credential that grants a status to the bearer of the credential. A Credential *type* is associated with each credential and has a fixed set of attributes contained in it. The type of the credential identifies the attribute contained in the credential. Use of standard credentials provides easy and immediate access to resource in open access environment. Granular access control can be achieved by defining the conditions on attributes associated with the credentials, the resources and the environment. The authorization certificate overrides the existing access policies and grants immediate access to the desired resource. The main components of credential based hybrid access control policy are as follows:

1. Credentials: The credentials are the basic elements of the access control policy. The credentials are considered as a bundle of attributes. Access control policy is defined in terms of various credentials like identity credential, attribute credential, standard credential and the authorization credential. Every credential is uniquely represented in the policy and the policy is defined as a set of conditions over heterogeneous credentials.
2. Attributes: In credential based hybrid access control, the access policy is defined not only in terms of credentials but also in terms of various attributes associated with the credentials, the data source and the environment. The credential attributes are similar to any other attribute with a difference that the credential attributes are always associated with some credential.
3. Conditions: The credential based access policy is defined as a set of various conditions that must be satisfied in order to gain access on the resource. The conditions can be of two types: the credential condition and the attribute condition. The credential condition specifies a credential or a set of credentials that the user must submit. The attribute condition contains various attributes associated with the credential, user, resource or environment as operands. The conditions are evaluated and if found satisfied, access is granted else denied.

3. XACML Framework

3.1 XACML Policy Specifications

XACML[13,15,16] defines an XML-based access control policy language for specifying the access control policy. It provides a processing model for evaluating these policies on the basis of the given XACML access request. The access request is specified by means of attributes like which subject wants to perform which action on which resource. The XACML Request Context is embedded in the <Request> element. The <Request> element contains <Subject>, <Resource> and <Action> elements. The <Subject> element contains the various attributes associated with the user, the <Resource> indicates the attributes of the requested resource and the <Action> specifies the action which the user is requesting on the resource.

A XACML policy contains one <Policy> or <PolicySet> as root element, which is a container for other Policy or PolicySet elements. Element <Policy> consists of a <Target>, a set of <Rule>, an optional set of <Obligation>, and a rule combining algorithm. A <Target> element includes conditions on Subject, Resource, Action, and Environment. If a request satisfies the conditions specified in the Target, the corresponding policy applies to the request. A <Rule> corresponds to a positive (permit) or a negative (deny) authorization depending on its effect. It includes an element <Target> and an element <Condition> specifying further restrictions on subject, resource and action. Each condition can be defined through element <Apply> with attribute FunctionID denoting the XACML predicate (e.g., string-equal, integer-less-than, etc) and with appropriate sub-elements denoting both the attribute against which the condition is evaluated and the comparison value. The rule's effect is then returned whenever the condition evaluates to true. The <Obligation> element specifies an action that has to be performed in conjunction with the enforcement of the authorization decision. Each element <Policy> has attribute RuleCombiningAlgID specifying how to combine the decisions of different rules to obtain a final decision of the policy evaluation (e.g., deny overrides, permit overrides, first applicable, only one applicable, etc). According to the selected combining algorithm, the authorization decision can be permit, deny, not applicable (i.e., no applicable policies or rules can be found), or indeterminate (i.e., some information is missing for the completion of the evaluation process).

3.2 XACML Architecture

XACML standard[13,15,16] defines a modular architecture for the evaluation of policies and a communication protocol for message interchange between various modules. The architecture includes different functional components that interact to take an access control decision. The access request is received by the Policy Enforcement Point (PEP) that is responsible for the enforcement of the access decision. The Policy Decision Point (PDP) is the component that evaluates the access policy and produces the access decision by retrieving the applicable policies from the Policy Administration Point (PAP). The PAP provides functionalities to administer access control policies. The Context Handler(CH) provides the interface between the PDP and PEP. It buffers the attributes that were given to the PEP along with the request and provides them to the PDP on demand. In addition to what is included in the access request, for decision process the PDP may also need additional information related to subject, resource or environment. The context handler provides this information via the Policy Information Point (PIP) that acts as a source of attribute values. PIP retrieves the attribute values by interacting with the subject, resource and environment modules.

Data flow in the standard XACML model can be summarized as follows:

1. The requester sends an access request to the PEP module.
2. The PEP module sends the access request to the Context Handler that translates the original request into a canonical format, called XACML Request Context, and sends it to the PDP.
3. The PDP extracts the applicable policies by means of the PAP module and retrieves the required attributes and the resource from the Context Handler. The Context Handler relies on the PIP module to acquire the attribute values associated with the subject, resource and the environment. For this purpose, the PIP interacts with the Subject, Resource and the Environment modules.
4. The PDP evaluates the policies and returns the XACML Response Context, together with an optional set of obligations, to the Context Handler. The Context Handler translates the XACML response context to the native format of the PEP and returns it to the PEP. If some information is missing, the PDP cannot take the decision and returns an error (Indeterminate response).
5. The PEP fulfills the obligations and, if permitted, it gives access to the requester. Otherwise, the access is denied.

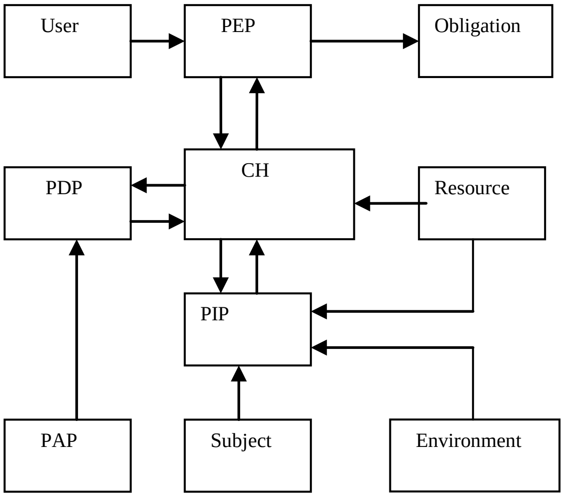

Fig 1: XACML Architecture

3.3 XACML and Credential based Hybrid Access Control

Although the XACML has been established as a solution for controlling access in a flexible way, it suffers from some limitations in supporting credential based access control. The current XACML standard enforces attribute-based access control[4,5,9,19] in which the access policy is defined in terms of various attributes. Here, the authorizations applicable to a user depend on the attributes that the user presents and the conditions that the user satisfies. While XACML permits that the attributes can be presented by means of certificates, the credential based access control requires the support for expressing the access control conditions not only on values of attributes contained in the credentials but also in terms of various types of credentials[10,13,15,16,18,19,20].

Several research efforts are in progress to extend the functionality of XACML to support credential based access control[10,13,15,16,18,19]. Representation of credential attributes in policy specification has been proposed in [16,19] but is limited to the evaluation of attributes like issuer, time and date only. Credential-based access control needs support for expressing the conditions on the types of the credentials in addition to the various attributes of the credentials. Need has been realized to uniquely identify each credential in the policy. The Standard credentials[11,14,22] have been proposed that have unique credential types. The credential type determines the attributes contained in the credential. The various credential types and the attributes contained in each credential is published and this facilitates the policy designers and the access requester in specifying and understanding the policy respectively. Recently, the similar concept of credential type has also been proposed in [10,13,16,19]. Credentials are realized as a bundle of attributes having a specific type. However, these works are more focused on addressing the problem of preserving the privacy of the various credential attributes. In [15], the author has pointed the need of having a common ontology of credential types and their attributes and to share this ontology between the policy makers and the user. Some recent proposals have investigated credential-based access control with anonymous credentials. In [23], an access control language has been introduced that is explicitly targeted to anonymous credentials. A card based access control system[10] has been proposed in which the card is similar to the credential. Rather that submitting various cards, depending on the policy, the user creates a claim from various cards. The claim contains a statement about the attributes of one or more of the cards along with their evidence. A card based access control requirements language (CARL) for technology-independent credential-based access control[10] has been introduced but the language follows a proprietary syntax, which makes deployment in real-world access control scenarios difficult. In [18,19], the author has focused on extending XACML to support credential based access control in web based scenarios. However their focus is on developing a mechanism for communicating the access policy to the access requester. In our work, we propose extensions to the XACML standard to enforce credential based hybrid access control. The extended XACML framework is suitable for providing open as well as closed access and thus handles diversified access control requirements.

4. Proposed Extension to XACML

In our work, we propose extension to XACML framework that supports credential based hybrid access control. Our proposal is divided into two parts. First we extend the access control policy specification language to support heterogeneous credentials, credential attributes and the conditions having credentials and attributes as operands. Second, we propose the functional enhancements in some of the architectural modules of XACML. The enhancements are designed in such a way that there is minimal impact on the policy language and the evaluation mechanism.

Following are the typical requirements of the hybrid credential based access control that entails enhancements in the XACML framework:

1. A Credential is the basic element of the access control policy. Access control policy is defined in terms of various categories of credentials like Identity credential, Attribute credential, Standard credential and Authorization credential. This is represented using C*redentialCategory* that specifies the category to which the credential belongs. Every standard credentials[11, 14, 22] has a unique type associated with it. Thus in addition to CredentialCategory, the standard credential also has a unique type identifier called as *CredentialType.* The credentials are uniquely represented in the policy using *CredentialCategory* and the *CredentialType*.

2. Every credential should be uniquely represented in the policy. A *CredentialId* is assigned to each credential that is being referred in the policy. The CredentialId is unique within the namespace of the policy.

3. As per the access policy, the user may have to submit one or more credentials. The access policy should support the specification of logical combination of credentials. For example, an access policy may expect the user to either submit the Standard Credential of Senior Citizen or an Attribute certificate issued by CityCorporation containing Date of birth as attribute.

4. The credential attributes are considered similar to other attributes. However, within the policy, the credential attributes are always referred along with the associated credential. The access policy should be able to define condition on credential attributes in addition to other types of attributes.
5. A standard credential is issued by the Standard Credential Authority and is easily verifiable using CIA hierarchy. Therefore the standard credential may not have issuer associated with it while being referred in the policy specification. Whereas the identity, attribute and authorization credentials can be issued by anyone and thus the policy should specify the issuer from whom these credentials can be accepted.

4.1 Extension to XACML Policy Specification

To support credential based hybrid access control, the extensions to XACML policy specification language have been proposed. The extensions are defined in such a way that they do not alter the semantics of existing elements or attributes. The proposed extensions are as follows:

4.1.1Representation of Credential

To represent credential in the access policy, we propose a new element <Credential> that corresponds to the definition of a credential and its attributes. To uniquely identify a credential with in the policy, <Credential> element has an attribute called as *CredentialId* whose value is the identifier by which the credential will be referred in the rest of the policy. CredentialId is unique within the namespace of the policy. The various attributes of the credentials are defined using <CredentialMatch> and <CredentialAttributeDesignator> elements that are similar to <SubjectMatch> and <SubjectAttributeDesignator>. For example the Standard Credential of Doctor can be represented as follows:

```
<Credential CredentialId = “SC1”>
          <CredentialMatch MatchId= “string-equal”>
                    <CredentialAttributeDesignator
                    DataType= “string”
                    AttributeId= “CredentialCategory”/>
          <AttributeValue> Standard </AttributeValue>
          </CredentialMatch>
          <CredentialMatch MatchId= “string-equal”>
                    <CredentialAttributeDesignator
                    DataType= “string”
                    AttributeId= “CredentialType”/>
          <AttributeValue>Doctor </AttributeValue>
           </CredentialMatch>
</Credential>
```

4.1.2 Logical combination of credentials

To specify logical combination of multiple credentials within the policy, we introduce a new element that corresponds to the specification of combination of credentials in the policy. We define <CredentialRequirements> within the <Target>. The <CredentialRequirements> contains one or more <Credentials> elements within it. The <Credentials> contain one or more <Credential> elements. Credentials expressed within one <Credentials> are considered in conjunction, whereas the <Credential> in different<Credentials> are considered in disjunction. The following code specifies that the user should submit either C1 and C2 or C3 in order to gain access where C1, C2 and C3 are defined as described in section 4.1.1.

```
<Target>
          <CredentialRequirements>
                    <Credentials>
                              <Credential>
                                        C1
                              </Credential>
                              <Credential>
                                        C2
                              </Credential>
                    </Credentials>
                    <Credentials>
                              <Credential>
                                        C3
                              </Credential>
                    </Credentials>
          </CredentialRequirements>
</Target>
```

4.1.3 Conditions involving Credential Attributes

In XACML, the operands appearing in conditions are typically the attributes associated with the subject, resource, action or environment. In our work, we extend XACML conditions to support credential attributes as well. Credential attributes can be treated just like any other attribute but the only difference is that they must be associated with a credential. For this purpose, we propose the inclusion of CredentialId attribute in the <CredentialAttributeDesignator> element. The CredentialId contains the value as defined in the <Credential> element for the credential in which the attribute being referred is contained. For example, the attribute “specialization” of the standard credential of doctor can be represented as follows:

```
<CredentialAttributeDesignator
          DataType= “string”
          CredentialId = “SC1”
          AttributeId=“specialization”/>
          <AttributeValue>orthopedics</AttributeValue>
```

4.1.4 XACML Request Context

The Context Handler generates the XACML Request Context which is sent to the PDP. The XACML Request Context has been extended so as to specify the various credentials submitted by the user. The standard Request Context contains <Subject><Resource> and <Action> element within the <Request> element. We propose one more element <Credentials> within the <Request> element. The <Credentials> element contains multiple <Credential> element one each for every credential submitted by the user. The <Credential> element contains details of all the attributes associated with the credential.

```
<Request>
```

```
        <Credentials>
                <Credential>
                <\Credential>
                ……….
                <Credential>
                <\Credential>
        <\Credentials>
        … Other Subject, Resource and Action
        elements…
<\Request>
```

4.2 Proposed XACML Architecture

Functional enhancements have been proposed to the XACML framework so as to make it suitable for enforcing credential based hybrid access control policies. The extensions have been proposed in such a way that the existing functionality of XACML is not altered and the extensions can be used in combination with the existing standard features. The modified XACML architecture is as shown in Figure 2. The greyed modules are modified as follows for the credential based access control.

- PEP: The PEP accepts the data access request from the user and sends it to the Context Handler. It has been modified to accept a credential or a set of credentials from the user. The PEP sends the credentials to the Context Handler.
- Context Handler(CH): The Context Handler is enhanced to have Credential Manager (CM) that verifies the credentials received from the PEP and extracts the values of various attributes associated with the credential. The CH then translates the access request into a canonical format, called XACML Request Context. The modified XACML Request Context is as described in section 4.1.4. The modified XACML Request Context is sent to the PDP.
- PAP: The PAP has been modified so as to administer the heterogeneous credentials based policies.
- PDP: The functionality of PDP has been extended to evaluate the credential based hybrid access control policies.

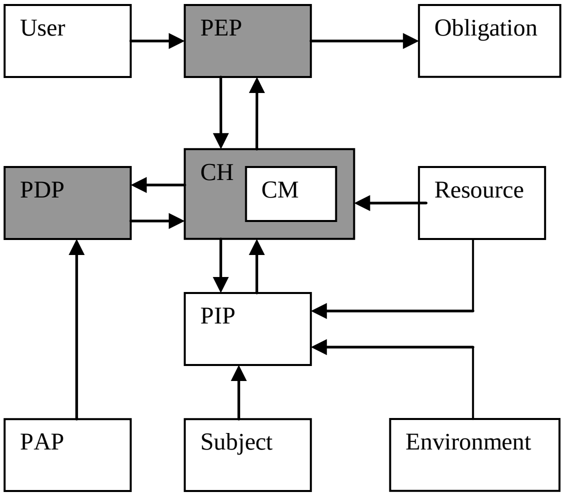

Fig 2: The Extended XACML Architecture

The data flow in the modified XACML model can be summarized as follows:

1. The requester sends an access request along with the set of credentials to the PEP.
2. The PEP accepts the credentials along with the request and sends them to the Context Handler.
3. The Context handler receives the credentials and the request from the PEP. The Credential Manager (CM) verifies the received credentials and extracts the values of various attributes associated with the credentials. The CH then translates the access request into a canonical format called XACML Request Context. The XACML Request Context is sent to the PDP.
4. The PDP identifies the applicable policies by means of the PAP module and retrieves the required attributes from the Context Handler. The context handler relies on the PIP module to retrieve the various attribute values.
5. The PDP evaluates the access policies. The PDP returns the XACML Response Context together with an optional set of obligations to the Context Handler. If some information is missing, the PDP cannot take a decision and returns an error (Indeterminate response).
6. The Context Handler translates the XACML Response Context to the native format of the PEP and returns it to the PEP.
7. The PEP fulfills the obligations and, if permitted, it gives access to the requester. Otherwise, the access is denied.

5. Illustrative Usecases

In this section, we have discussed few usecases from the medical domain. Some of the typical access control rules that exist in medical domain are as follows:
Rule 1: A Doctor can access details of only those patients that have disease same as his own specialization ie. an orthopedic doctor can access details of orthopedic patients, a gynecologist can access details of gynecology related patients, etc.
Rule 2: A known or registered doctor can access all details of his patients only.
Rule 3: A doctor can access the medical details of any patient.
Rule 4: A medical student who has enrolled in the university before year 2009 can access the medical details of patients between 9 AM and 5 PM.
Rule 5: A person can access all details of a specific patient if he brings an authority letter from the health secretary.

The XACML representation of the first two rules is shown below. The other rules can be defined similarly. The code snippet shown below is just an illustration of the various access control rules.

```
<Rule RuleId= “Rule1” Effect= “Permit”>
 <Target>
<CredentialRequirements>
<Credentials>
<Credential  CredentialId  = “SC1”>
          <CredentialMatch MatchId= “ string-equal”>
          <CredentialAttributeDesignator
                    DataType= “string”
                    AttributeId= “CredentialCategory”/>
                    <AttributeValue>              Standard
                    </AttributeValue>
          </CredentialMatch>
          <CredentialMatch MatchId= “string-equal”>
          <CredentialAttributeDesignator
                    DataType= “string”
AttributeId= “CredentialType"/>
          <AttributeValue>Doctor </AttributeValue>
          </CredentialMatch>
</Credential>
</Credentials>
</CredentialRequirements>
</Target>
<Condition >
          <Apply FunctionId= “string-equal”>
          <ResourceAttributeDesignator
                    DataType= “string”
                    AttributeId= “Disease”/>
          <CredentialAttributeDesignator
                   DataType= “string”
                    CredentialId = “SC2”
                    AttributeId= “Specialization”/>
          </Apply>
</Condition>
</Rule>
```

```
<Rule RuleId= “Rule2” Effect= “Permit”>
 <Target>
          <CredentialRequirements>
          <Credentials>
          <Credential  CredentialId  = “IC1”>
          <CredentialMatch MatchId= “ string-equal”>
          <CredentialAttributeDesignator
                    DataType= “string”
                    Issuer = “XYZ”
                    AttributeId= “CredentialCategory”/>
          <AttributeValue> Identity </AttributeValue>
          </CredentialMatch>
          </Credential>
          </Credentials>
          </CredentialRequirements>
</Target>
<Condition >
          <Apply FunctionId= “string-equal”>
          <CredentialAttributeDesignator
                    DataType= “string”
                    CredentialId = “IC1”
                    AttributeId= “Subject”
          <ResourceAttributeDesignator
                    DataType= “string”
                    AttributeId= “DoctorId”/>
          </Apply>
</Condition>
</Rule>
```

6. Conclusion

In this paper, we have presented extensions to the standard XACML framework for supporting credential based hybrid access control. The extended XACML policy language supports specification of various types of credentials. Fine grained access control is provided by defining conditions associated with credentials in addition to attributes associated with subject, resource and environment. The extended language does not alter the semantics of the existing elements and attributes of the policy language. The functionality of some of the modules of the XACML architecture has been enhanced to accept various types of credentials and to evaluate and enforce credential based policies defined using extended XACML language. To facilitate the adoption of our approach in existing XACML code bases, we have designed our extensions for minimal impact on the language and the XACML evaluation mechanism. The proposed extensions are suitable for developing access control systems that support open access as well as fine grained control in closed domain. Access to high priority user by overriding the existing policies is also possible.

References

1. Sudhir Agarwal, Barbara Sprick, Sandra Wortmann, Credential Based Access Control for Semantic Web Services, American Association for artificial Intelligence, 2004
2. Mary R. Thompson, Abdellilah Essiari, Srilekha Madumbai, Certification based Authorization policy in a PKI Environment, ACM Transactions on Information and System Security (TISSEC), Volume 6 , Issue 4, pages: 566 – 588, 2003
3. William Johnston, Srilekha Mudumbai and Mary Thompson, Authorization and Attribute Certificates for widely Distributed Access Control, IEEE WETICE, 1998
4. Damiani, E.,Di Vimercati, S.D.C.,Samarati, P., New paradigms for access control in open environments**,** Proceedings of the Fifth IEEE International Symposium on **S**ignal Processing and Information Technology, 2005
5. Shen Hai Bo, Hong Fan, An attribute based access control model for web services, proceeding of the seventh international conference on Parallel and Distributed Computing, Applications and Technologies, IEEE 2006
6. Mary Thompson, William Johnston, Srilekha Mudumbai, Gary Hoo, Keith Jackson, Abdelilah Essiari, Certificate-Based Access Control For Widely Distributed Resources, ACM, Proceedings of the 8th conference on USENIX Security Symposium - Volume 8, Pages: 17 - 17 , 1999
7. Nirmal Dagdee, Ruchi Vijaywargiya, Credential Based System for realizing Open and Domain confined accesses on Shared Data Sources, International Conf on data management, 2008
8. Ioannis Mavridis, Christos Georgiadis, George Pangalos, Marie Khair, Access Control based on Attribute Certificates for Medical Intranet Applications", Journal Of Medical Internet Research,Vol3, 2001
9. Sabrina De Capitanidi Vimercati, Pierangela Samarati, New Directions in Access control, www.spdp.dti.unimi.it/papers/nato.pdf, 2002
10. Jan Camenisch, A Card Requirements Language Enabling Privacy-Preserving Access Control, ACM 2010
11. Nirmal Dagdee, Ruchi Vijaywargiya: Credential based hybrid access control methodology for shared Electronic Health Records, IEEE International Conference on Information Management and Engineering, Kuala Lumpur, Malaysia, 2009
12. P. Bonatti and P. Samarati. A unified framework for regulating access and information release on the Web, Journal of Computer Security, 10(3):241–272, 2002
13. Jan Camenisch et al, Credential-Based Access Control Extensions to XACML, www.w3.org/2009/policy-ws/papers/Neven.pdf, 2009
14. Nirmal Dagdee, Ruchi Vijaywargiya, Policy architecture for credential based access control in open access environment, Journal of Information Assurance and Security 6, 039-047, 2011
15. Claudio A. Ardagna et al , Enabling Privacy-Preserving Credential-Based Access Control with XACML and SAML, www.zurich.ibm.com/~nev/ papers/credprofile.pdf, 2010
16. Claudio A. Ardagna, An XACML-Based Privacy Centered Access Control System", Conference on Computer and Communications Security", Proceedings of the first ACM workshop on Information security governance, Chicago, Illinois, USA, pages: 49-58, 2009
17. Pierangela Samarati et al, Enriching Access Control to Support Credential-Based Specifications, subs.emis.de/LNI/Proceedings/Proceedings19/GI-Proceedings.19-10.pdf, 2002
18. Claudio A. Ardagna et al, Extending XACML for Open Web-based Scenarios, http://www.w3.org/ 2009/policy-ws/papers/Samarati.pdf, 2009
19. Claudio A. Ardagna, Expressive and Deployable Access Control in Open Web Service Applications, IEEE Transactions on Services Computing, Vol. 4, No. 2, April-June 2011
20. Sonia Jahid et al, Enhancing Database Access Control with XACML Policy, www.sigsac.org/ccs/CCS2009/pd/ abstract_23.pdf, 2009
21. Mariemma I. Yagüe, Survey on XML-Based Policy Languages for Open Environments, Journal of Information Assurance and Security 1, pages 11-20, 2006
22. Dr. Nirmal Dagdee, Access control methodology for sharing of open and domain confined data using Standard Credentials, International Journal on Computer Science and Engineering Vol.1(3), 148-155, 2009
23. Claudio A. Ardagna, Exploiting cryptography for privacy enhanced access control, JCS, vol. 18, no. 1, 2010

Dr. Nirmal Dagdee has earned his BE, ME and PhD degrees in Computer Engineering. His major research interests are in the fields of Service oriented computing, Data security and Soft Computing. He has authored several research papers that are published in reputed journals and conference proceedings. Presently, he is Director of S. D. Bansal College of Technology, Indore, India. He is a member of IEEE and ACM.

Ruchi Vijaywargiya, a faculty of computer science in S.D. Bansal College of Technology, Indore, India has around 20 years of experience in academics and software industry. She has done BE and ME in Computer Engineering and is pursuing PhD under the supervision of Dr. Dagdee in the field of data security and access control. Her areas of interest are data security, computer networks and object oriented technology. She is a member of IEEE.