USAGE CONTROL MODEL SPECIFICATION IN XACML POLICY LANGUAGE

XACML Policy Engine of UCON

Um-e-Ghazia, Muhammad Awais Shibli, Rahat Masood, Muhammad Bilal

National University of Science and Technology, Islamabad, Pakistan

{10msccssghazia, awais.shibli, 10msccsmmasood, m.bilal}@seecs.edu.pk

Abstract. Usage control model (UCON) is one of the emerging and comprehensive attribute based access control model that has the ability of monitoring the continuous updates in a system making it better than the other models of access control. UCON is suitable for the distributed environment of grid and cloud computing platforms however the proper formulation of this model does not exist in literature in any policy specification standard. It is for this reason that UCON is not widely adopted as an access control model by industry, though research community is now paying attention to make standard policy specification for this model. In this paper we are suggesting the interpretation of UCON model in extensible access control markup language (XACML) which is an OASIS standard of access control policies. We also highlight UCON model features by explaining its core processes and characteristics with respect to the case study of financial application.

Keywords: Access Control: Authorization: Obligation: Condition: Policy: Attribute

1 Introduction

Access control models play vital role in protecting digital resources in a way to control and mediate access from unauthorized users. These models assure the security requirements of different applications improving their protection from unauthorized access. They secure digital information and resources until the permission is granted to user and are suitable for closed domain comprising static entities. Continuous evolution of computing platforms demands advanced security procedures and mechanisms to handle the consequences of unauthorized access.

Usage control model (UCON) has been proposed by Park and Sandhu that caters heterogeneous and dynamic environment conditions [1]. It is an attribute based access control that judges three factors: authorization, obligations and conditions for access decision. Two distinctive characteristics are attribute mutability (updates) and access

decision continuity which augments UCON model than other traditional access control models. In UCON model usage session is maintained that is categorized into three parts: pre access, ongoing access and post access phases. When the subject S is accessing resource R, UCON decision factors i-e. attributes, obligations and conditions are evaluated before granting access which is called pre access phase. Evaluation is also performed during the time when R is in use by S called as ongoing access phase. In some cases, it is required to evaluate attributes and obligations after the completion of usage session which is known as post access phase. Decision is continuously evaluated during these three phases to monitor the changes in attribute values. In addition to it, UCON model is comprehensive enough to cover the traditional access control models. So it can be used to provide the better protection of system resources in a collaborative and dynamic environment. Despite of all the excellent features, UCON is not widely adopted as an access control model to restrict the unauthorized access. The major reason for this is that the model features are not been translated in any of the standard policy language to offer the proper formulization of model..

XACML [17] being a generic policy language of OASIS standard is suitable to represent the aforementioned features of UCON model. It describes the platform independent access control request/ response mechanism and policy specification. Its interoperability feature makes it widely used for various platforms and environments. XACML offers the extension points by introducing new data types, identifiers, elements and functions to offer a generic policy structure. Since UCON can facilitate the diverse range of applications like digital rights management (DRM), health care systems and social networking, it is highly encouraged to provide the formal specification of model in generic policy language like XACML. There is a need to define the separate profile of UCON in XACML that will enable organizations to adopt this flexible model. Also to guarantee the accurate access decision in different deployment scenarios, it is mandatory to propose the required alterations and additions in generic policy language of XACML which is not developed so far. We wish to propose the implementation of UCON model in XACML by incorporating additional elements and specify the information flow of different UCON processes in this paper.

Organization of paper is as follows: Section 2 presents the detailed UCON model and its core features, Section 3 includes the profile of UCON model in XACML, use of this profile is explained in Section 4 by taking the scenario of posting vouchers service in financial applications and Section 5 concludes the paper and present future directions.

2 Usage Control Model

UCON being an attribute based access control model accommodates the security requirements by the addition of more than one decision factors which makes it more reliable and flexible [3]. This model primarily restricts the usage of digital objects and provides the efficient mechanism to include the traditional access control models. Previous access control models only encompass authorization rules in making access decision; rather UCON model also consider the obligations and environmental condi-

tions. Moreover collaborative environments demand the need of enhanced provisioning and controlled access to digital resources. In addition to the immutable attributes that are explicitly modified by the administrator, system controlled mutable attributes are also managed by constant monitoring throughout the stages of usage session.

UCON model identifies three types of subjects; consumer, provider and identifiee. Consumers are the subjects who make request to perform certain action on object. Providers are the individuals who own services and issue the rights to the requesting party. Identifiee is the entity whose confidential information is incorporated within digital object. It is an optional group of subjects which may or may not be present depending on system requirements however it is always present in case of systems having users' confidential information. Depending on the job functions of subjects, three types of rights (actions) are specified namely consumer, provider and identifiee rights which indicates the set of actions or privileges on digital objects [1]. Apart from these, there are other actions as well that fall in the category to perform updates in attributes values during the phases of usage session which are termed as usage control actions [2].

UCON model also classify the objects as privacy sensitive and privacy non sensitive objects that determines whether the object contains critical information of identifiee subject or not. Improper management of privacy sensitive objects cause security breaches which results in data disclosure to unauthorized users and compromising data integrity. There is another phenomenon of UCON model called as reverse UCON in which the position of consumers and providers are inverted depending on the scenario. Complete classification of UCON subjects, objects and rights is shown in Figure 1.

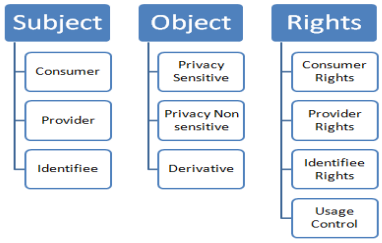

Fig. 1. UCON classification of subject, object and rights

All the traditional models include authorization rules that need to be satisfied before the using the resources called as rights related authorization rules. UCON identify additional obligations related authorization rules which are the set of actions related to access request to be completed before granting permissions of resources. Obligations are first time addressed by UCON model to improve the accuracy of access decision by enforcing users to perform certain actions before access. Obligations also act as functions that check whether the obligatory actions are fulfilled by the requesting entity or not. In order to further increase the accuracy of authorization decision, another factor to be considered important in UCON model are the environmental conditions such as ip addresses, current date or time. Conditions can be of two types;

dynamic (stateful) and static (stateless). Dynamic conditions have constantly changing information so they needs to be evaluated for every update and static conditions do not have to be checked for each update during usage session [1].

2.1 Authorization models of UCON

UCON Pre authorization models are carried out in the same way as for traditional authorization models. In this model, user credentials and resource attributes are checked before granting permission for the requested resource. Pre authorization models can be with immutable and mutable attributes (pre, ongoing and post updates) in which attributes either remain same during access phase or their values change before, during or after the access phase. Ongoing authorization models evaluate the attribute values during the resource usage by the user to perform the continuous verification. Ongoing authorization models can also be with immutable and mutable attributes as with pre authorization models. The effect of ongoing authorization evaluation in access control model is that the access rights can be revoked during the session as certain attribute value is changed.

2.2 Obligation models of UCON

Obligation models comprise the pre and ongoing obligation monitoring of access request. They improve the accuracy of access decision in a way that it requests user to perform the access related mandatory actions first and then access would be granted.. Obligations that are used before and during the usage session are termed as pre and ongoing obligations respectively. Post obligations are introduced in order to execute actions after access session is finished i-e access fulfillment notifications to service provider. These post obligations can affect the decision of future usage sessions by generating request to policy repository for policy modification [14]. Pre, ongoing and post obligation models with immutable and mutable attributes perform the obligation checking before, during or after the access phase.

2.2 Condition models of UCON

UCON has two condition models; *pre and ongoing condition with immutable attributes* to cater the environmental constraints and system related parameters. Conditions are evaluated before and during the session in the same way as authorization rules. Rather condition models do not consider the mutable attributes such as the changing location of a subject.

3 UCON MODEL SPECIFICATION IN XACML

In order to provide UCON model specification in XACML, we are going to introduce additional identifiers and attribute values to incorporate the features of UCON model. XACML has identifiers for subject categories like `access-subject, reci-`

pient-subject, intermediary subject. They are used under the tag of AttributeDesignator that is one of the methods of attributes retrieval in XACML. Subject-type identifier is introduced for UCON subject categories and their values might be consumer, provider and identifiee. UCON subject identifiers are used along with XACML subject category access-subject to further specify the type of accessing subject as follows.

```
<AttributeDesignator
Category=urn: oasis: names: tc: xacml: 1.0: subject-category:
access-subject
Type=urn: oasis: names: tc: xacml: 3.0: subject-type: consumer>
```

In the same way, action-type identifier is introduced with the XACML attribute category of action to reflect the UCON categories of action i-e consumer, provider, identifiee and usage control actions.

```
<AttributeDesignator
Category=urn: oasis: names: tc: xacml: 3.0: attribute-category:
action
Type=urn: oasis: names: tc: xacml: 3.0: action-type: usage-
control>
```

For demonstrating UCON objects, resource category identifier is created under attribute category of resource that has the value of privacy-sensitive, privacy-nonsensitive or derivative.

```
<AttributeDesignator
Category=urn: oasis: names: tc: xacml: 3.0: attribute-category:
resource
Type=urn: oasis: names: tc: xacml: 3.0: resource-category: de-
rivative>
```

Since the UCON model consider both mutable (updating values) as well as immutable (constant values) attributes, new identifier attribute-class is constructed to differentiate between them. Mutable attributes are then further narrow down into pre-mutable, ongoing mutable and post mutable. This general classification of attributes is used with all of the attribute categories of subject, resource, action and environment.
Rights related authorization rules can be mapped in XACML as general rules but the time of evaluating these rules needs to be managed. So the pre and ongoing element of authorization rules is explained by the rule-id attribute of the rule element.

```
RuleId="urn: oasis: names: tc: xacml: 3.0:pre-authorization"
```

Obligation element is specified in XACML for mandatory actions required to be performed by the subject that can also handle the obligation related authorization rules of

UCON. Obligation expression element includes arguments that are required to execute the obligation. We have proposed the new attribute namely `Fulfill-phase` for specification of pre and ongoing obligations. It indicates the access phase during which obligation must have to be satisfied by PEP, so it may have the values of `pre-access, ongoing-access` that reveals the pre and ongoing obligations.

```
<ObligationExpression
ObligationId="urn: oasis: names: tc: xacml: ucon-example: obli-
gation: license-agreement
Fulfill-phase="pre-access">
```

`Condition` element in XACML contain single expression element which includes functions to be evaluated. We have introduced additional attribute `condition-type` under the condition element to present the UCON dynamic and static conditions. Furthermore pre and ongoing element of condition models are expressed by introducing new attribute called as `evaluation-phase` under the condition element. It can have the value of pre-access or ongoing access.

```
<Condition
Condition-type = "urn: oasis: names: tc: xacml: 3.0: condition-
type: dynamic
Evaluation-phase= ongoing-access">
```

3.1 UCON Access Control Framework

We are proposing an access control framework which has the policy information flow modules of XACML like PDP, PEP, PIP and PAP. Policy repository is a unit in XACML that resides between the PAP and PDP containing the access control policies of corresponding model. Generally PAP creates the access control policies and pushed them into policy repository to be used for evaluation by PDP. We have proposed additional module integrated with PAP that has the capability to interpret the newly created UCON identifiers, attributes and their values. This interpretation will help to determine whether the processing is to be performed in before, ongoing or after phase of usage session. This module is called as *UCON policy builder* that incorporates UCON model features in policy specification, when PAP accepts the inputs from policy administrator to formulate the generic UCON policy (Fig. 2).
In addition to it, *UCON policy engine* module is incorporated with PDP that provides the main features of UCON model like attribute mutability and decision continuity. *UCON policy engine* has sub modules like *attribute mutability (AM)* and *decision continuity (DC)* as shown in Figure3. *AM* handles the updating of attribute values in three access phases; before, during and after access. As a result of attribute values modification, *DC* monitors the continuous policy evaluation of three access decision factors; authorization, obligation and condition. So the pre and ongoing models of authorization, obligation and conditions are deployed accordingly with mutable and immutable attributes.

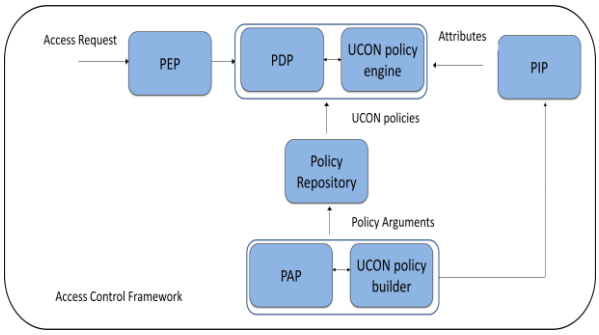

Fig. 2. UCON Access Control Framework

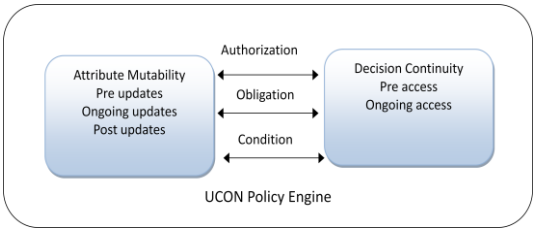

Fig. 3 UCON Policy Engine

UCON policy engine in conjunction with PDP evaluates the combined effect of target, condition, rules and obligations of policy or policy set and applies the specified combining algorithms to generate the final response for user. Functions for continuous policy evaluation are embedded in policy engine to carry out the constant monitoring of attributes, obligations and conditions in order to support the ongoing feature of UCON model. *UCON policy engine* in combination with *policy builder* will perform different functions such as obligation monitoring, attributes management, maintaining history of updated events of certain access request, notifies user about the current policy status for request.

4 Example Scenario

UCON model can be widely adopted in distributed environment to manage the controlled access for applications like health care systems, database management systems, resource sharing systems etc. We are considering the financial application to demonstrate the proposed specification of UCON model in XACML. Managing financial data is one of the key challenges in large enterprises. Financial data is comprised of financial entries regarding its employees, customers, vendors, assets, inventory, products, cost and profit centers etc hence play a vital role in enterprise progress and development. Often the highly secret and confidential in-formation related to

enterprise and their employees also reside in these applications, thereby it is required to provide security in terms of data privacy, data loss and data access. General Ledger (GL) is a key component of financial applications that acts as a central repository of company's accounting transactions. All the business transactions are posted to GL in the form of journal vouchers having debit and credit card entries. These entries are further classified into assets, liabilities, revenues, expenses, capital/owner equity. Core modules of GL includes chart of accounts, financial calendars, journal entries, ledgers, trial balance, balance sheet, profit and loss statement, cash flow statement and user defined report writer.

In this section, we will demonstrate our proposed methodology through the journal entries module. Journal entries are broadly categorized into five types like bank payment voucher, cash payment voucher, bank receipt voucher, cash receipt voucher and journal voucher. Every user of application is not authorized to create/post journal entries rather a user with specific attributes like *name*, *department* and *role* can create/post voucher for certain amount. For authorized users, they are not allowed to create/post entries for all accounts. Moreover an authorized user with permissions to create/post journal entries further requires quota/limit on journal entry amount for the permissible account. Different accounts act as object and the attributes allotted them are *accessing list* (specifies which user can access this account), *limit of total amount* (represents value that a user can post for a voucher). In this scenario policy is formulated as; user *John* having a role of *Director General* belonging to *Administration* department cannot post vouchers beyond the limit of 50,000$ in a day time. Further he can only reference *City Bank Account # 345678B, 657463C* in the credit entries of all vouchers. This scenario workflow is based on pre authorization and ongoing authorization, pre obligation and ongoing obligation and pre condition that are described below according to aforementioned scenario and their corresponding XACML code snippets.

Before providing access to journal entry management service, above specified subject and object attributes are verified according to policy specification that corresponds to pre authorization mechanism. UCON policy engine will represent this pre authorization mechanism for the attribute *subject-id* "John" as follows.

```
<Rule RuleId="urn: oasis: names: tc: xacml: 2.0: ucon-example:
ongoing-authorization"...>
     <AttributeDesignator Category="urn: oasis: names: tc:
     xacml:   1.0: subject-category: access-subject"
     Type="urn: oasis: names: tc: xacml: 3.0: subject-type:
     consumer"
     AttributeId="urn: oasis: names: tc: xacml: 1.0: subject:
     subject-id"
     DataType="http://www.w3.org/2001/X   MLSchema#string"
     Class="urn: oasis: names: tc: xacml: 3.0: attribute-class:
     immutable"/>
```

Since *voucher limit* of *John* is fifty thousand for a day, posting voucher service is revoked as limit reaches before a day time. Request is no more facilitated and a mes-

sage is prompt to user that he is not eligible to post a voucher in present day. It is an example of *ongoing authorization model* of UCON which is shown below.

```
<Rule RuleId= "urn: oasis: names: tc: xacml: 3.0: ongoing-
authorization"...>
     <AttributeDesignator Category="urn: oasis: names: tc:
     xacml: 1.0: subject-category: access-subject"
     Type="urn: oasis: names: tc: xacml: 3.0: subject-type: con-
     sumer"
     AttributeId="urn: oasis: names: tc: xacml: 1.0: subject:
     subject-voucher-limit"
     DataType="http://www.w3.org/2001/X   MLSchema#double"
     Class="urn: oasis: names: tc: xacml: 3.0: attribute-class:
     ongoing-mutable"/>
```

Accessing list attribute of account object is updated after the usage session which is the example of post update. This attribute keeps record of users that are accessing certain account, so it is classified as privacy-sensitive object. It is also useful for auditing and management purposes.

```
<Rule RuleId= "urn: oasis: names: tc: xacml: 3.0: ongoing-
authorization"...>
      <AttributeDesignator Category="urn: oasis: names: tc:
      xacml: 3.0: attribute-category: resource"
      Type="urn: oasis: names: tc: xacml: 3.0: resource-
      category: privacy-sensitive"
      AttributeId="urn: oasis: names: tc: xacml: 1.0: subject:
      accessing-list"
      DataType="http://www.w3.org/2001/X   MLSchema#string"
      Class="urn: oasis: names: tc: xacml: 3.0: attribute-class:
      post-mutable"/>
```

Obligations are categorized as system related obligations and subject related obligations [14]. System related obligations are those actions that are executed by the service provider in order to ensure the verification of requesting party. On the other hand subject related obligations are performed by the requesting subject which is enforced by the service provider. In the present situation, subject related pre obligation is to accept the application terms and conditions before accessing journal management service. This subject related obligation is performed just once when the subject requests to access the service for the first time.

```
<ObligationExpression ObligationId="urn: oasis: names:  tc:
xacml: ucon-example: obligation: license-agreement"
Fulfill-phase="pre-access">
        <AttributeAssignmentExpression
        <AttributeDesignator Category="urn: oasis: names: tc:
        xacml: 1.0: attribute-category: resource
```

```
        AttributeId="urn: oasis: names: tc: xacml: 1.0: re-
        source: window-id"
        </AttributeAssignmentExpression>
        <AttributeAssignmentExpression
        <AttributeDesignator Category="urn: oasis: names: tc:
        xacml: 1.0: subject-category: access-subject"
        AttributeId="urn: oasis: names: tc: xacml: 1.0: sub-
        ject: subject-id">
        </AttributeAssignmentExpression>
</ObligationExpression>
```

As the specific limit of voucher is reached for particular user, insert option for voucher becomes invisible to that user. This is also an example of *ongoing obligation* performed by service provider. *Voucher limit* for *John* is fifty thousand in current scenario, after that the post voucher option is disable for him.

```
<ObligationExpression ObligationId="urn: oasis: names: tc:
xacml: ucon-example: obligation: disabling-post-voucher"
Fulfill-phase="ongoing-access">
           <AttributeAssignmentExpression
           <AttributeDesignator Category="urn: oasis: names: tc:
           xacml: 1.0: attribute-category: resource
           AttributeId="urn: oasis: names: tc: xacml: 1.0: re-
           source: disable-post-voucher-option">
           </AttributeAssignmentExpression>
</ObligationExpression>
```

Further conditional parameters might be incorporated as pre condition in the form of user specific ip-address which helps to determine the user location before access. In this case, policy `condition` element states that specific ip-addresses of "x.x.x.x and "y.y.y.y" (which falls in the organization subnet) can access the service of posting journal voucher.

```
<Condition Condition-type = "urn: oasis: names: tc: xacml: 3.0:
condition-type: static"
Evaluation-phase= pre-access"
FunctionId="urn: oasis: names: tc: xacml: 1.0: function: not"
        <Apply
        FunctionId="urn: oasis: names: tc: xacml: 1.0: func-
        tion: string-at-least-one-member-of">
        <AttributeDesignator
        AttributeId="urn: internetexplorer: names: internetex-
        plorer: 2.1: environment: httpRequest: clientIpAddress"
        DataType="http://www.w3.org/2001/XMLSchema#string"/>
        <Apply FunctionId="urn: oasis: names: tc: xacml: 1.0:
        function: string-bag"
```

```
        <AttributeValue
        Data-
        Type="http://www.w3.org/2001/XMLSchema#string">127.0.0.
        1
        </AttributeValue>
        <AttributeValue
        Data-
        Type="http://www.w3.org/2001/XMLSchema#string">128.84.1
        03.11
        </AttributeValue>
        </Apply>
        </Apply>
</Condition>
```

5 Conclusion and Future Work

For the applicability of UCON model features in XACML, some of the corresponding modifications are proposed in paper to cater the authorization requirements of distributed environment. Our main objective was to develop the comprehensive UCON framework in XACML that should be reliable, flexible and scalable. XACML being a generic policy language can better translate the UCON model features. It will provide model specifications in a formal manner to be deployed as an access control model by practical application environment. This framework provides the policy administration interface that will help enterprises in implementing UCON model within their applications. It will be generic and consistent to accept the arguments according to scenario and improve the accuracy of access decision.

Future directions in this domain include the identification of issues and problems in UCON model with respect to distributed and collaborative platforms like grid and cloud computing. In addition to this, detailed performance analysis, usability and interoperability issues of UCON model need to be addressed. UCON models of authorization, obligations and conditions can further be investigated to integrate them with each other and to show interactions between parallel usage sessions. Moreover, UCON access control framework can be further extended to provide the main feature of extensibility. Different access control models can be incorporated within this framework to offer uniformity and consistency across applications. Depending on application security requirements, organizations can select any model with much more flexibility and ease. Distributed environments such as cloud computing can adopt this generic access control framework to provide better resource protection.

References

1. Jaehong Park, Ravi Sandhu: Towards Usage Control Models: Beyond Traditional Access Control. In: SACMAT ‘02 Proceedings of 7th ACM Symposium on Access Control Models and Technologies (2002)

2. Xinwen Zhang: Formal model and analysis of usage control, PhD Thesis. George Mason University, Fairfax, USA (2006)
3. Aliaksandr Lazouski, Fabio Martinelli, Paolo Mori Moore: Usage control in computer security: A survey. In: ELSEVIER journal of Computer Science Review Volume 4, Issue 2 (2010)
4. Jaehong Park, Ravi Sandhu: The UCON ABC Usage Control Model. In: Journal of ACM Transactions on Information and System Security Volume 7, Issue 1 (2004)
5. Xinwen Zhang, Masayuki Nakae, Michael Covington, Ravi Sandhu: Toward a Usage-Based Security Framework for Collaborative Computing Systems. In: Journal of ACM Transactions on Information and System Security Volume 11, Issue 1 (2008)
6. Ponnurangam Kumaraguru, Lorrie Faith Cranor: A Survey of privacy policy languages. In: SOUPS'07 Proceedings of third symposium on usable privacy and security (2007)
7. Antonios Gouglidis, Ioannis Mavridis: On the Definition of Access Control Requirements for Grid and Cloud Computing Systems. In: GridNets'09 Third International ICST Conference (2009)
8. Xinwen Zhang, Francesco Parisi-Presicce, Jaehong Park, Ravi Sandhu: A Logical Specification of Usage Control. In: SACMAT'04 ACM Transactions on Information and System Security (2000)
9. Jianfeng Lu, Ruixuan Li, Vijay Varadharajan, Zhengding Lu, Xiaopu Ma: Secure Interoperation in Multidomain Environments Employing UCON Policies. In: ISC '09 Proceedings of the 12th International Conference on Information Security (2009)
10. Subashini, Kavitha: A survey on security issues in service delivery models of cloud computing. In: ELSEVIER journal of Network and Computer Applications Volume 34, Issue 1 (2010)
11. Diala Abi Haidar, Nora CuppensBoulahia, Frederic Cuppens, Herve Debar: An Extended RBAC profile in XACML. In: SWS '06 Proceedings of the 3rd ACM workshop on Secure web services (2006)
12. Chen Danwei, Huang Xiuli, Ren Xunyi: Access Control of Cloud Services Based on UCON. In: CloudCom '09 Proceedings of the 1st International Conference on Cloud Computing (2009)
13. Tamleek Ali, Mohammad Nauman, Fazl-e-Hadi, Fahad bin Muhaya: On Usage Control of Multimedia Content in and through Cloud Computing Paradigm. In: 5th International Conference on Future Information Technology (2010)
14. Basel Katt, Xinwen Zhang, Ruth Breu, Michael Hafner, Jean-Pierre Seifert: A General Obligation Model and Continuity-Enhanced Policy Enforcement Engine for Usage Control. In: SACMAT '08 Proceedings of the 13th ACM symposium on Access control models and technologies (2008)
15. eXtensible Access Control Markup Language (XACML) Version 2. Standard, OASIS, April 2009
16. Core and hierarchical role based access control (RBAC) profile of XACML v3.0. Standard, OASIS, August 2010
17. A Brief Introduction to XACML, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html