Challenging Issues of UCON in Modern Computing Environments

Christos Grompanopoulos
Department of Applied Informatics, University of Macedonia
156 Egnatia Street
Thessaloniki, Greece
groban@uom.gr

Ioannis Mavridis
Department of Applied Informatics, University of Macedonia
156 Egnatia Street
Thessaloniki, Greece
mavridis@uom.gr

ABSTRACT

Usage CONtrol (UCON) is a next generation access control model enhanced with capabilities presented in trust and digital rights management. However, modern computing environments are usually introducing complex usage scenarios. Such a complexity results in involving a large number of entities and in utilizing multi party contextual information during the decision making process of a particular usage. Moreover, usage control is demanded to support novel access modes on single or composite resources, while taking into account new socio-technical abstractions and relations. In this paper, a number of challenging issues faced when UCON is applied in modern computing environments are highlighted through the utilization of representative usage scenarios. The results of this study are revealing various limitations in contextual information handling, lack to support complicated usage modes of subjects on objects, and weaknesses in utilizing information concerning previous or current usages of system resources.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; D.4.6 [**Operating Systems**]: Security and Protection—*Access controls*

General Terms

Security

Keywords

Information Security, Access Control, Usage Control, UCON

1. INTRODUCTION

The constantly decreasing size of computing devices, combined with the increment of networking capabilities offered by wireless technologies, provides users with abilities to access information anywhere and anytime. Despite the benefits of such an ubiquitous information sharing, a number of security challenges are also raised. Moreover, traditional access control models, responsible for granting access to resources, are incapable to support the new requirements of modern computing and communication environments [1, 14, 2].

Usage CONtrol (UCON) is a next generation access control model enhanced with capabilities presented in trust and digital rights management [8]. It associates subjects and objects with attributes and also supports three decision factors, namely authorizations, obligations and conditions. Additionally, two significant innovations to access control models, viz. continuity of decision and attribute mutability, are introduced in UCON [9].

Despite UCON's virtues, supporting complex usage scenarios in modern computing environments remains a challenging procedure [14, 2]. The involvement of a large number of entities together with the utilization of contextual information originating from a number of sources during the usage decision making process, results into an increased complexity of usage scenarios. Additionally, the support of novel access modes on resources, along with new socio-technical abstractions and relations, which are created during the usage process [14] is another significant issue for usage control in modern computing environments. In fact UCON's attribute mutability provides limited knowledge about the usages exercised on system resources and further complicates the administration process. Furthermore, obligation management required by UCON policies creates an additional complexity to the operations of the access control system.

The remainder of this paper is organized as follows: The UCON model is described briefly in Section 2. In the following sections a number of UCON's limitations are located in the context of given usage scenarios and solution approaches are discussed. More specifically, shortcomings in UCON's contextual information handling are presented in Section 3. In Section 4 the lack of UCON to support the complicated usage modes that are required in modern computing environments is revealed. The limitations caused by the utilization of the attribute mutability mechanism are included in Section 5. An alternative approaches to managing obligation enforcement are examined in Section 6. Section 7 discusses the criteria utilized in UCON's decision making process and resumes their weaknesses. Finally, we conclude

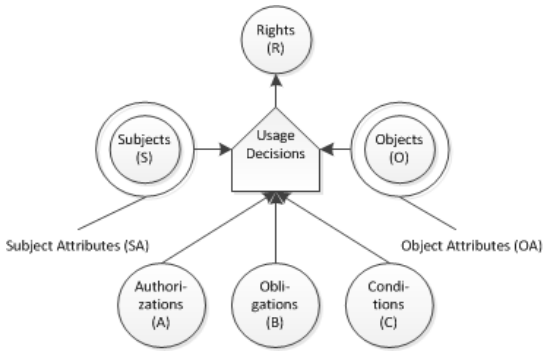

Figure 1: Components of the UCON model [9].

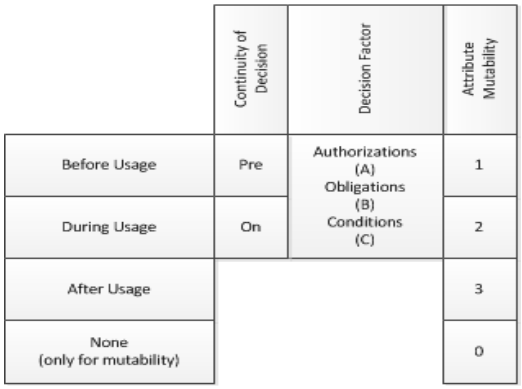

Figure 2: Family of UCON sub-models.

the whole discussion in Section 8.

2. A BRIEF INTRODUCTION TO UCON

The UCON model [9] consists of six components, viz. subjects, objects, rights, authorizations, obligations and conditions. Subject is the entity that requests the usage of another entity (object), both introduced from the primary access control models [5]. However, subjects and objects in UCON are defined and represented with security relevant characteristics called attributes. The utilization of subject and object attributes provides the capability for fine-grained usage control on resources and the ability to support users without knowledge of their identity [4]. The usage modes that a subject can exercise on an object are represented in UCON with rights. Rights are specific and they are not associated with attributes, in contrast to subjects and objects.

The allowance of a subject to exercise a right on an object is not predetermined, but it is decided at the time of the usage request, with the evaluation of a decision factor. UCON introduces three decision factors, named Authorizations (A), oBligations (B) and Conditions (C). Authorizations are functional predicates that utilize the values of subject and object attributes. An obligation is one or more activities that must have been performed by the subject as a requirement for the requested usage to be allowed. The selection of the required obligations is based on the attribute values of the subject and the object involved in the usage. Moreover, the subject and object pair in an operation required by an obligation, may differ from the subject and object pair involved in the usage under consideration. Conditions are also functional predicates, like authorizations, but they utilize only system related information contained in special system variables, named condition variables. The UCON components and their relationships are depicted in Figure 1 as originally presented in [9].

Two innovative concepts introduced in UCON is continuity of decision and attribute mutability [9]. More specifically, the exercise of a right from a subject in an object, is not a spontaneous action, but it may last some time and include several sub-actions. With the utilization of continuity of decision, authorizations, obligation and conditions are categorized into pre and ongoing (required before and during the usage, respectively). Moreover, attribute mutability updates the attributes of the subject or the object as a consequence of an allowed usage. The attribute update procedure can be executed before (1), during (2), after (3) the exercise of an allowed usage, or not executed at all (0). Figure 2 is a graphical representation of the family of UCON sub-models [9] that encompasses decision factors, continuity of decision and attribute mutability.

3. CONTEXTUAL INFORMATION HANDLING

The utilization of contextual information in the decision making process is an essential feature of access control in modern open computing environments [1]. Contextual information in UCON is stored into special system variables, called condition variables [9]. Condition variables in turn, are utilized by UCON's functional predicates called conditions. However, there are cases where the utilization of multiple condition variables, during the decision making process in UCON, can lead to a number of challenges. An usage scenario where multiple sources of contextual information are required for the creation of the usage decision is the following:

Usage Scenario 1. The personnel of a company is categorized to employees and managers. Every employee is supervised by a corresponding manager. Whenever an employee, named Alice, sets a request to execute a crucial operation, e.g. open a vault, a policy directive requires the physical coexistence of Alice's manager, named Bob.

In usage scenario 1, a policy rule requires the comparison between two locations represented by two separated condition variables. The problem arises from the fact that Bob is neither the subject nor the object of the requested usage. In such a case, specifying in UCON the particular condition variable that represents Bob's location seems to be impossible. Even if UCON can represent with an Alice's attribute the fact that Bob is her manager, it is not possible to link at the same time Bob with a condition variable that represents his location. A solution could be provided by utilizing a number of condition variables that represent contextual information, which is irrelevant to the subject or the object of the usage (e.g. "subjectsAdminLocation" may be the condition variable that represents the location of Bob in our example). However, in a system with a large number of

condition variables, such an implementation could result in a very complicated usage control system.

Another issue is related to the condition evaluation, which in UCON is a complicated process. Instead of authorizations, where only a predicate is evaluated, condition evaluation is a two-stage process [9]. Firstly, the get-condition operation is executed gathering the required conditions based on subject and object attributes. Secondly, the selected conditions are evaluated using the corresponding condition variable values. As a result, a significant delay in the decision making process can be introduced due to this complexity.

A solution to the above issues could be the assignment of contextual information to subject and object attributes, which was also proposed in [2]. This modeling decision is also based on the notice in [9], stating that there is a vague line differentiating which information should be assigned to attributes and which to condition variables. For example, if a user tries to access a server located in a different country, there are two time value candidates for the same condition variable. Instead of defining two condition variables, the addition of an attribute *time* to both the subject and the object of the usage is proposed. By assigning contextual information to subject and object attributes, all the participating entities in the usage control system are directly associated with their specific context.

4. SUPPORTING COMPLICATED USAGE MODES

Rights in UCON are described as "privileges that a subject can hold and exercise on an object" [9]. However, rights are not described by attributes, as opposed to subjects and objects. Such a modeling decision seems to be adequate for simple and straight-forward rights, like reading or writing a file. However, in modern computing environments subjects may need to access objects with novel and complicated access modes. A usage scenario that requires a complicated usage mode is the following:

Usage Scenario 2. Electronic banking transactions must confirm with a government policy directive requesting that players of on-line betting companies must be adults. More specifically, whenever a customer attempts to make a money transfer to a betting company's account, his age must be evaluated. In addition, the amount of money transfer must not overcome the customer's account balance.

In the usage scenario 2, the customer's account is the subject of the usage, whereas the money transfer is the right and the account of the betting company is the object. The necessity for associating the amount of money with the money transfer results in the need for enhancing rights with a more detailed description.

A possible solution in UCON, to overcome the lack of right attributes, might include decomposing the transaction of usage scenario 2 into two different rights. In the first right, the customer's account is the subject, whereas the money transfer is the object. In the second, subject is the money transfer and object is the bank account of the betting company. For each one of these rights, a respective UCON policy rule is created. However, utilizing these two UCON policy rules has significant drawbacks. Firstly, UCON modeling does not depict that these two policy rules are related with each other. Only the policy administrator is aware that these two rules are correlated and must be checked in a particular order. Secondly, and more importantly, is the fact that UCON's modeling is quite different from the way security policies are expressed, usually through natural languages, e.g. the same way assembly differs from SQL programming. As a result, UCON introduces additional complexity to the policy administrator tasks. Thirdly, the fact of using only simple rights (e.g. read, write, etc.) limits the policy administrator's ability to create a rule that controls either all the rights or only a specific one.

The replacement of UCON's *right* component with a new one, called *action*, could provide a solution to the previous described UCON right shortcomings. Actions can be described, like subjects and objects, with attributes. It is worth mentioning that the necessity for further enrichment of UCON with right attributes was also proposed in [17]. Moreover, actions can support efficiently complicated rights. The previous example with the banking transaction can now be modeled by utilizing an action having attributes that describe the transaction amount, the purpose of money transfer, etc. Actions, along with their attributes, can contribute to the increasing of policy language expressiveness. In addition, action attributes can reduce the gap between UCON modeling with multiple policy rules and expressing security policies in almost natural language. Action attributes can also enable the creation and management of possible existing right hierarchies, as presented in [9].

5. KEEPING INFORMATION ABOUT SYSTEM USAGES

Traditional access control models can be replaced with UCON's authorizations only [8]. In addition, by utilizing attribute mutability, UCON becomes capable to support consumable rights, history - based access control and dynamic constraints, like cardinality or separation of duty [16]. However, authorizations combined with attribute mutability are quite restrictive, due to the fact that the decision for allowing or not future usages with a subject A and an object B can be based only on previous usages with subject A and/or object B. The aforementioned feature is caused by the fact that authorizations utilize only the attribute values of subject A and object B, while the values of attributes of A and B may be updated only as a result of previous usages with subject A and/or object B. Nonetheless, in modern computing systems it is likely that a usage control decision should be based on previous usages with subjects and/or objects that are not related with the ones of the current usage request, like the one introduced in the following usage scenario.

Usage Scenario 3. In a research institute, a presentation room is equipped with both an interactive board and a media player. A policy rule requires that an employee is permitted to access the media player only if there is no other presentation in progress in the same room.

In order for UCON to support usage scenario 3, the authorization rule for the usage of the media player should utilize the attributes of interactive board, which however is not involved in the usage under consideration.

Attribute mutability in UCON may also introduce difficulties into the policy administrator tasks. In usage scenario 3, by analyzing the policy requirements, the administrator must recognize that the allowance for using the media player is depended on the knowledge for the current use of the interactive board. As a result, a new attribute should be created for recording whether the interactive board is currently in use or not. However, in modern computing systems there may be hundreds or thousands of different device usages. To cope with them, the policy administrator must recognize and record, through attribute mutability, all the probably correlated usages. In addition, as mentioned earlier, a completed usage may result in updating the value of an attribute that another usage may utilize in its authorization rules. Thus, attribute mutability can produce dependencies between different usages. Therefore, policy administrators must be very careful in modifying policy rules of a usage in order to avoid side effects to other usages. The above facts are contributing to the conclusion that attribute mutability complicates the policy administration task. Furthermore, updates of attribute values must be executed for each one allowed usage, resulting so in prolonging the execution time of the usage control system. In usage scenario 3, every time the interactive board is used, both pre and post update procedures [9] will be executed. However, it is possible that the rule that utilizes interactive board's attribute value will never be evaluated. Even if nobody requests the media player, the attribute value must be updated, resulting so in overloading the overall system performance.

In UCON, attribute values are updated only after a usage is allowed. Therefore, a requested but denied usage can not be recorded by the system. In addition, in an ongoing authorization, a usage can be discontinued either by a subject's request (completed) or by system's termination (stopped), due to an ongoing authorization rule's violation. UCON will use the same attribute update procedure, no matter if the usage was completed or terminated. However, as stated in [16], a discrimination between the ongoing usages, which have been completed or stopped must be supported by the usage control model.

In order to address the above mentioned issues, the complete recording of the system usages is required maybe with the introduction of a new entity, called *use*. A use is created, when a subject requests to exercise a usage to an object through an action. Subsequently, as the usage continues, the use records its evolution. A use is actually a materialized view of a usage in progress. By searching uses, the policy administrator gains full knowledge of the previous system usages without the need for utilizing attribute mutability. Moreover, uses enable the usage control system to handle knowledge for every status of usage evolution: requested, allowed, denied, completed and stopped. For example, an administrator in UseCON is able to find all the (requested, activated, denied, completed or stopped) usages that are related to a particular object. However, using UCON he is restricted into retrieving the value of the object attribute that contains only the number of its previous (completed or terminated) usages (as proposed in [9]). Use entities can also facilitate the fine-grained definition and proper association of attributes to various system entities. More specifically, according to UCON, information characterizing an object-right combination, like the price of a service, must be contained into object's attributes [9]. However, the price of a service is actually derived from the subject-right-object combination, resulting so in a differentiated pricing for the same pair of right-object requested by various subjects. Through uses existence, information characterizing the subject-action-object combination to be associated with the proper use attributes.

A core benefit from the utilization of use attributes is the capability to store information characterizing the relation between usages, e.g. those belonging to the same transaction. For example, in an accounting office, a transaction that updates the files of a customer may be composed by a number of usages. However, all those usages, as long as they concern the same customer, should share a common attribute with the same value that represents the transaction wherein the usages belong. Subsequently, a policy rule can utilize this value, and permit all the usages of the transaction with the same privacy agreement.

6. MANAGING OBLIGATION ENFORCEMENT

One of the three UCON's decision factors is obligations [9], which are functional predicates verifying the fulfillment of necessary operations on objects. The selection of the required obligations is based on the attribute values of the subject and the object involved in a usage. Moreover, the subject and object pair involved in an operation required by an obligation may differ from the subject and object pair of the usage under consideration. An usage scenario that requires the utilization of an obligation is the following:

Usage Scenario 4. An educational software system provides the capability to display question's solution. However, a solution could be revealed to a student only if the student previously tried at least two times to answer the question.

In UCON, such a policy rule is modeled as a pre-obligation rule with an attribute update. Each time a student tries to answer a question the update of the appropriate attribute value is caused. After the second attempt the question's solution is available to be read by the same student.

In [16] the authors argue that UCON's obligation operations (named actions) are activities that must be performed as a requirement for the allowance of a usage request. Furthermore, obligation operations are not dependent on permissions and therefore, they can always be performed. However, a mechanism is required to be implemented in order for the UCON decision making engine to recognize whether an obligation operation was performed or not [6]. Such a mechanism is proposed in [15], which associates attributes with obligation operations. Each time an obligation operation is performed, the associated attribute value is updated. Consequently, the fulfillment of obligations can be verified by checking the values of associated attributes.

Table 1: Usage decision criteria in UCON

Usage Decision Criteria	UCON mechanism	UCON sub-model	Comments
Security Characteristics of Entities	Subject and Object attributes	A	Attribute values are updated through an administrative process
Contextual Information	Condition variables	C	Conditions may utilize contextual information only directly related with the entities of the usage requested.
Past usages of entities involved in access request and/or usages of other entities	Attribute mutability	A	Complicated policy administration process Limited knowledge of previous usages
	Obligations management	B	Absence of an obligation fulfillment enforcement mechanism "Obscure" discrimination between obligation operations and normal usages

An alternative obligation modeling approach to handle obligation operations as common usages, as also proposed in [3]. Hence, the execution of obligation operations is managed by the usage control system. Moreover, a search in the previous uses of the system can confirm whether an obligation operation was executed or not.

7. UTILIZING DECISION MAKING CRITERIA

The above discussion concludes that the decision for allowing or not an access request is a complicated process in modern computing environments. Classical access control models utilize only a single criterion for the allowance of an access request, which is related to the security characteristics of the subject and the object involved in the access request [11]. More specifically, whenever a subject requests to access an object, the clearance of a subject and the classification of an object are utilized in Mandatory Access Control (MAC) models [13]. The identity of both subject and object are evaluated in Discretionary Access Control (DAC) models [10] and the active role from the set of assigned roles to the subject and the identity of the object are utilized in Role Based Access Control (RBAC) models [12]. Attribute based access control approaches [7] provide enhanced flexibility compared to the previous mentioned access control models, by utilizing a number of security related characteristics of subjects and objects, which are expressed in the form of attributes, in order to provide fine-grained access control. However, modern computing environments require the utilization of a number of factors for the creation of the access control decision. UCON employs the following criteria during the process of decision making (Table 1):

Security Characteristics of Entities: Security characteristics of system entities are limited to subject and object attributes. These attributes are utilized by UCON authorization predicates in order to create a usage decision. A UCON's authorization may utilize attributes not only related to subject and object of the requested usage but also to other entities of the system. The values of these attribute are updated only manually by administrators and not automatically by the attribute mutability mechanism.

Contextual Information: Contextual information in UCON is associated with special system variables, named condition variables. However, using UCON modeling results in significant difficulties when attempting to determine the particular condition variable that represents the contextual information that is related with a given system entity, as already described in usage scenario 1. Consequently, the process of decision making on a request of a subject S to use an object O utilizes only contextual information that is related to S or/and O.

Past usages of entities involved in access request and/or usages of other entities: Attribute mutability in UCON is implemented by storing information about allowed system usages in the values of subject or object attributes e.g. every time a user listens to a music file an attribute value of is updating. Therefore, the values of these attributes do not represent security characteristics of the involved entities and can be updated only by the attribute mutability mechanism. As a result, future UCON authorizations are able to utilize information about allowed usages for the decision making process of a requested usage. However, as already mentioned in Section 4, attribute mutability faces a number of limitations. More specifically, attribute mutability provides limited knowledge of the system usages (only those containing attribute updates) and further complicates the policy administration process. Another mechanism that utilizes information about previous usages of the subject and the object of the requested usage for the allowance of future usage requests is obligations. UCON obligation operations also represent usages exercised by subjects on objects. However, these obligation operations are discriminated from normal system usages and are considered to be always doable usages that are not supervised by the usage control system.

In modern computing environments it is more likely that the allowance of a usage of an object by a subject may be dependent on past usages of other entities of the system. Such usage scenarios can be implemented through UCON obliga-

tions. However, the lack of a feasible obligation fulfillment mechanism is mentioned in the literature [6]. Moreover, the treatment of obligation operations as normal system usages could provide an efficient solution to the obligation management issue.

8. CONCLUSIONS

In this paper, a number of challenging issues that are revealed when applying UCON in open and dynamic computing environments were discussed in the context of suitable usage scenarios. Through the analysis of the usage scenarios, a number of new requirements were specified that result in the need for a new usage control approach with enhanced capabilities in handling information related to context, usage modes and history. More specifically, the relationship between contextual information and corresponding entities should be utilized. In addition, the additional information encompassed in complicated usage modes of subjects on objects in modern computing environments present should be supported by attributes of rights. Moreover, a comprehensive knowledge of the system usages augmented with that offered by the attribute mutability mechanism is expected to provide efficient support in complicated usage scenarios by utilizing various access control related criteria (attributes, context or usages) of involved and third-party entities (subjects and objects). Finally, an alternative obligation modeling approach, similar to the one presented in this paper, could efficiently address the issues regarding the obligation management process in UCON. In our future work, we intend to propose a new usage control model that will incorporate all the aforementioned capabilities in order to support the access control needs of modern computing and communication paradigms.

9. REFERENCES

[1] E. Damiani, S. D. C. di Vimercati, and P. Samarati. New paradigms for access control in open environments. In *SIGNAL PROCESSING AND INFORMATION TECHNOLOGY*, pages 540–545, 2005.

[2] C. Grompanopoulos and I. Mavridis. Towards differentiated utilization of attribute mutability for access control in ubiquitous computing. *Informatics, Panhellenic Conference on*, 0:118–123, 2010.

[3] H. Janicke, A. Cau, and H. Zedan. A note on the formalisation of UCON. In *Proceedings of the 12th ACM symposium on Access control models and technologies*, SACMAT '07, pages 163–168, New York, NY, USA, 2007. ACM.

[4] L. Kagal, T. Finin, A. Joshi, and S. Greenspan. Security and privacy challenges in open and dynamic environments. *Computer*, 39(6):89 –91, june 2006.

[5] B. W. Lampson. Protection. *SIGOPS Oper. Syst. Rev.*, 8:18–24, January 1974.

[6] A. Lazouski, F. Martinelli, and P. Mori. Usage control in computer security: A survey. *Computer Science Review*, 4(2):81 – 99, 2010.

[7] OASIS. Oasis extensible access control markup language (xacml) tc, 2011.

[8] J. Park and R. Sandhu. Towards usage control models: beyond traditional access control. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, SACMAT '02, pages 57–64, New York, NY, USA, 2002. ACM.

[9] J. Park and R. Sandhu. The UCON abc usage control model. *ACM Trans. Inf. Syst. Secur.*, 7:128–174, February 2004.

[10] L. Qiu, Y. Zhang, F. Wang, M. Kyung, and H. R. Mahajan. Trusted computer system evaluation criteria. In *National Computer Security Center*, 1985.

[11] P. Samarati and S. D. C. d. Vimercati. Access control: Policies, models, and mechanisms. In *Revised versions of lectures given during the IFIP WG 1.7 International School on Foundations of Security Analysis and Design on Foundations of Security Analysis and Design: Tutorial Lectures*, FOSAD '00, pages 137–196, London, UK, UK, 2001. Springer-Verlag.

[12] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *Computer*, 29(2):38 –47, feb 1996.

[13] R. S. Sandhu. Lattice-based access control models, 1993.

[14] R. K. Thomas and R. Sandhu. Models, protocols, and architectures for secure pervasive computing: Challenges and research directions. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW '04, pages 164–, Washington, DC, USA, 2004. IEEE Computer Society.

[15] X. Zhang, M. Nakae, M. J. Covington, and R. Sandhu. Toward a usage-based security framework for collaborative computing systems. *ACM Trans. Inf. Syst. Secur.*, 11:3:1–3:36, February 2008.

[16] X. Zhang, F. Parisi-Presicce, R. Sandhu, and J. Park. Formal model and policy specification of usage control. *ACM Trans. Inf. Syst. Secur.*, 8:351–387, November 2005.

[17] X. Zhang, J. Park, F. Parisi-Presicce, and R. Sandhu. A logical specification for usage control. In *Proceedings of the ninth ACM symposium on Access control models and technologies*, SACMAT '04, pages 1–10, New York, NY, USA, 2004. ACM.