ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
UDC/UDK 004.738.056

A CONTEXTUAL USAGE CONTROL MODEL

Xiaofeng Luo, Lin Li, Wanbo Luo

Original scientific paper

The usage control model (UCON) is the latest major enhancement of traditional access control models. It enables subject and object attributes mutability and usage control continuity. However, with the model access permission may be denied as a result of the environmental changes even though the authorization and obligation requirements are met, thus causing disruptions to users. Contextual UCON (CUC) was proposed to overcome this major weakness of UCON. In CUC context was introduced to replace the conditions component in UCON. And management module was added to manipulate the subject and object and context attributes. CUC seamlessly combines control and management modules and has the ability to dynamically adapt the changes in context, and is truly attribute-based. An algebra approach was employed to describe CUC syntax and semantics formally.

Keywords: *access control, computer security, context, IT (Information Technology), UCON*

Kontekstualni model praćenja uporabe

Izvorni znanstveni članak

Model praćenja uporabe (UCON) je najnovije veliko poboljšanje tradicionalnih modela za praćenje pristupa. On omogućava promjenljivost atributa subjekta i objekta i kontinuitet praćenja uporabe. Međutim, taj model može zabraniti pristup zbog promjena u okolini čak i ako su zadovoljeni zahtjevi autorizacije i obveze te tako korisnicima stvoriti prekide. Predložen je kontekstualni UCON (CUC) kako bi se prevladala ta osnovna slabost UCONa. U CUC-u se uvodi kontekst kao zamjena za komponentu uvjeta u UCON-u. Dodaje se modul upravljanja za manipuliranje atributima subjekta, objekta i konteksta. CUC izravno kombinira module praćenja i upravljanja i može dinamički prilagođavati promjene u kontekstu te je uistinu baziran na atributima. Primijenjen je algebarski pristup za opis sintakse i semantike CUCa.

Ključne riječi: *IT (Informacijska tehnologija), kontekst, praćenje pristupa, sigurnost računala, UCON*

1 Introduction

Access control is one of the basic and critical technologies for information security. It works with other security services in information systems to provide information security. Access control is basically an everyday phenomenon and has a long history. A pair of lock and key, for example, is a typical form of access control. Modern access control technology was impregnated in late 1960's and early 1970's. Lampson first gave a formal description for the access control mechanism by introducing the concepts of the subject, object and access matrix [1]. Over the past 40 years, a lot of access control models were developed, such as Bell-La Padula (BLP) Model, Harrison-Ruzzo-Ullman (HRU) Model, Biba Integrity Model, Clark-Wilson Integrity Model, Chinese Wall Model, Clinical Information Systems Security Policy Model, Role-Based Access Control model (RBAC), Usage CONtrol model (UCON), and so on [2].

The UCON proposed by Park et al. is an important advance since RBAC [3 ÷ 6]. It was recognized as the latest major enhancement of the classic access control models, and drew considerable interests and attentions. For example, by the time this paper was written, about 3000 records related with UCON were returned by the keywords "usage control" (in Chinese) or UCON on China Knowledge Resource Integrated Database (CNKI). After they surveyed the literatures on usage control over the period 2002 to 2010, Lazouski et al. found that research on UCON is still active, and some research issues are open and need more work [7].

The original UCON is made of six core components: subjects (with attributes), objects (with attributes), rights, authorizations, obligations, and conditions [8]. The first three components are inherited from traditional access control models, and have similar meanings. The last three are new components for usage control decisions. Authorization permits or rejects a subject request based on the evaluation of the subject and/or the object attributes under conditions. Obligations are requirements a subject has to perform before or during his/her access. Conditions are system and environment constraints for decision, and independent of both subjects and objects attributes [4, 5]. However, as [4, 5] pointed, condition variables of UCON are not mutable. UCON aims to enable subject and object attributes mutability and control continuity. Because of the decision continuity, the access permission may be denied as a result of the environmental changes even though the authorization and obligation requirements are met. This major weakness is rooted at the static conditions variables. As the rapid development of modern computing and information technology, the application environments become more and more complex and varied. For example, in a cloud computing environment, the equipment, the systems, the platforms or even the organizations are virtualized. To make an access control effective, it is necessary to dynamically adapt the changes in environments, so as to ensure continuity of usage.

Some recent researches were implemented to improve the UCON model. For example, Zhao et al. introduced a time variable into UCON, called as TUCON (Times-based Usage Control) [9]. Zhang Hong-jun added geospatial factor into UCON to ensure the spatial characteristics of location-aware application security [10]. Bai et al. proposed ConUCON (context-aware usage control model) for data and resource protection in mobile computing environment [11], and applied it to improve security protection for WoT (Web of Things) [12]. Almutairi and Siewe employ context information to make UCON context-aware in pervasive computing systems

[13]. However, these attempts did not completely solve the issue.

In this paper we introduce contexts to replace the conditions component in UCON to overcome the inadequacy of the original UCON model. This approach is different from those [9 ÷ 13]. In the rest of this paper, we first describe the concept and principle of the context component. Then the improvement of UCON is illustrated, followed by the components framework and a formal description by an algebra approach. Then is the comparison of CUC. Finally some conclusions and further work are presented.

2 Context

The word "context" derived from the Latin words *con* and *texere* originally means something weaved with or together. There are some definitions of context. In linguistics, for example, context is the parts of a written text or spoken discourse that surround a particular word or passage and clarify its meaning. For an event, context is the parts of something that perform the event setting. In Chinese it is also known as the origin and development of things to happen.

In computer science and technology area, research on context and context-aware has been done in many literatures since Schilit et al. first introduced the concept of context-aware [14 ÷ 16]. The widely-accepted definition of context was provided by Dey [17].

Dey's definition is focused on entity. In a usage transaction, it is natural to concern the events or actions. In this paper action-oriented context can be explained below.

Definition 1 (**Context**) Context is any information that can influence the situation of an action. Any information includes conditions and factors that impact, constrain, or relate to the occurrence, development, existing, and change of the action.

From the temporal view context information falls into two categories: static and dynamic. In administrative view it can be divided as mutable and immutable. Administrative action can change the values of both mutable and immutable portions, whereas access can change values of the mutable portions only. Examples of static context are system platform, operation system. Network bandwidth is immutable, it is initially determined by hardware configuration, and can be reconfigured by administrator. But the current available bandwidth is mutable because both current online users and devices can change it.

We need both static and dynamic (or immutable and mutable) portions for information security.

The so-called usage context is a context in a usage process. For instance, in a typical access control, in order to determine the access permission we should consider subjects, objects, time, location, environment, reasons, and so on. These issues are all related to the context of the access control.

Considering a QoS system, when the users are less, all users can get high quality service. As users increase to a certain extent, the system response speed is likely to decline, the network available bandwidth will become smaller. The information should be dealt with promptly and corresponded timely. The contexts can cap with it, but the immutable conditions of UCON cannot.

In linguistics Frege brought out two important principles: (1) the context principle. Words have meaning only in the context of a sentence; and (2) the principle of compositionality. The meaning of a sentence is determined by the meanings of its constituents [18]. We try to apply Frege's principles in computer security, called as security context principle.

Security context principle: The security without application context is nonsense.

This means that the definition of security is always relative to an application context. Without the context we cannot know if the information is secure or not. Information considered secure in a particular application context may be in a dangerous state in another context.

3 Contextual usage control model

This section presents the proposed contextual usage control model (CUC), which explicitly includes context component and has capability of dynamic adaptation to the changes in the environmental context. Like the subjects and objects in UCON, the usage context component will be characterized in attributes.

3.1 The system components

Fig. 1 is a system components diagram of CUC. There are seven components around the usage transaction: subjects set S with attributes set SA, objects set O with attributes set OA, contexts set X with attributes set XA, rights set R, authorizations set A, and management set M. Here SA and OA are identical to $ATT(S)$ and $ATT(O)$ in [4] respectively. So XA can be denoted by $ATT(X)$.

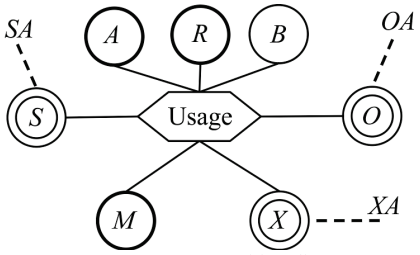


Figure 1 CUC composition diagram

Definition 2 (CUC System) A CUC system is a 7-tuple:

$CUC ::= (S, O, R, X, A, B, M)$

where S is the set of subjects with attributes; O set of objects with attributes; X set of contexts with attributes; R set of rights; $A: R \times O \mid (X,B) \rightarrow S$, set of authorizations, means to grant R to S to use O under the conditions X and B; $B: S \times R \mid (B,X) \rightarrow O$, set of obligations, means S apply R to O under the conditions B and X; $M ::= (M \| A, M \| B)$ is a set of management functions, where $M \| A$ means the M loosely couples A, and $M \| B$ the M loosely coupling B.

For S, O, R, A, and B, more detailed descriptions are in [4, 5] because they are inherited directly from UCON. Following is a brief explanation of the components.

Subjects set *S*

S is a set of subjects. A subject is an entity which takes actions on objects. A subject is characterized by its attributes. Examples of subject attributes are identity, role, reputation, credits, and so on.

Objects set *O*

O is a set of objects. An object is the entity that subjects can hold rights on. An object is characterized by its attributes. Examples of object attributes are identity, value, role permission, and so on.

Rights set *R*

R is a set of rights. A right is a privilege held or applied by subjects for objects. Rights consist of usage functions enabling subjects to access objects. From the viewpoint of access control, the right can enable subjects to access objects in a specific mode, such as to read or write.

For original UCON this is the moment to determine if subjects hold a right when the subjects try to access. Unlike UCON, CUC clearly distinguishes between authorization and application of rights, so that it supports directly the visualization of rights.

Contexts set *X*

X is a set of usage contexts. A context is characterized by its attributes. Context attributes are properties or constraints that can influence a usage transaction. Context attributes are divided as either static and dynamic, or mutable and immutable.

From administrative view contexts did not only inherit the conditions element of UCON, but also added new portions, e.g. the mutable attributes. The mutable attributes of contexts are just-in-time information that is relevant to usage transaction. So the CUC decision based on contexts can be more reasonable and adaptive, and easily offer fine-grained control.

In the QoS scenario the system current states as contexts attributes should be used to make usage decision, the service quality got by low-level users will decline, but the service quality of high-level users will be unchanged.

In UCON, as decision factors, conditions exclude the mutable information, so do the individual subject and object which are involved in the usage transaction. So the usage cannot be controlled.

Authorization set *A*

A is a set of predicate authorizations. Authorization is to bind subjects and rights in a usage context.

Obligations set *B*

B is a set of predicate obligations. Obligations which can be done by a subject either before, during, or after the access are requirements that the subject as a principal in a usage context has to do.

Management set *M*

M is a set of management functions. *M* is divided into two categories: one is for attributes management, and another for rules management. The functions for management of attributes of subjects, objects and contexts include addition of new attributes, removal of existing attributes, updating and evaluation of values of attributes, store of critical attributes values, and so on. Changes of attributes values can occur in three stages: before, ongoing and after a usage transaction. The functions for management of the usage control rules are used for the rules of authorizations decisions and right exercise decisions.

From the operational view, *M* can also fall into either administrative functions, or run-time functions. The addition/remove of attributes, for example, is administrative function, and evaluation of attributes values run-time function.

In original UCON the management functions are implicit, and only focus on evaluation of values of the subjects and objects attributes. The subjects, objects, and their attributes are static (predefined), and the values of attributes can be changed only. The component *conditions* of UCON model is separated from the management, and dealt with as administrative issues [5].

On the contrary, CUC defines and enhances the management functions explicitly so that it can co-ordinately combine the access control and management. The combination of control and management is a trend of information security. Of course it needs more work on it.

3.2 The system architecture

Fig. 2 is an alternative diagram of CUC which shows the system architecture. The architecture of CUC consists of two functional blocks: the management block (MB), and the control block (CB). MB is made of four components: *S* with *SA*, *O* with *OA*, *X* with *XA*, and the management set *M*. CB consists of three components and two decision points: *A*, *B*, *R*, the authorization decision point (ADP), and the right exercise decision point (EDP). There is loose coupling between the management and control blocks. In MB subjects, objects, contexts, and their attributes are managed by *M*. The loose coupling means any change of either *S*, or *SA*, or *O*, or *OA*, or *X*, or *XA* should not bring on changes of the access control rules and algorithms in CB, and vice versa.

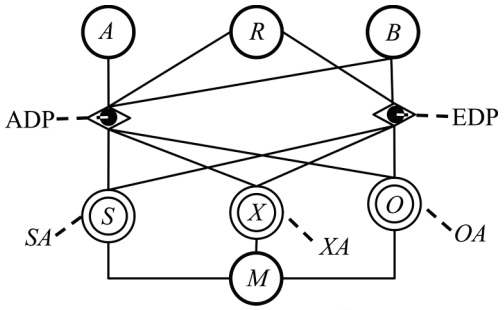
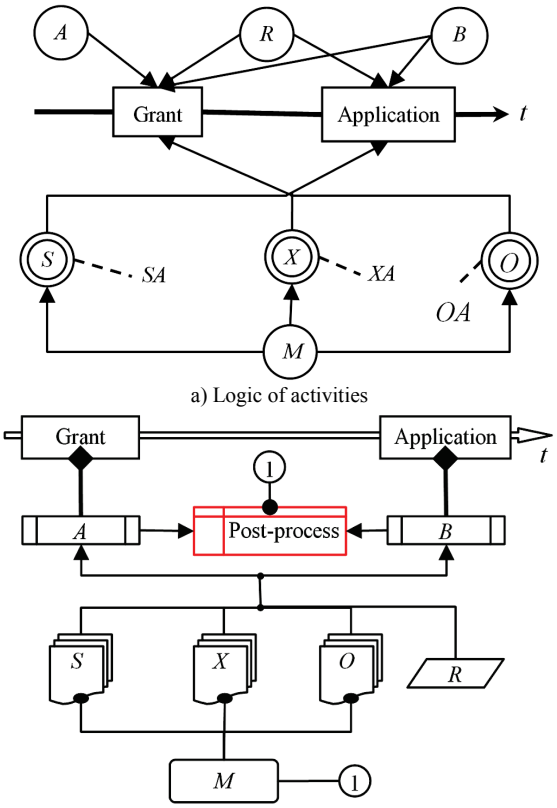


Figure 2 CUC architecture diagram

3.3 The activities

For convenience we redraw CUC model in Fig. 3. The timeline at the middle of Fig. 3a indicates the direction of time-series activities, and the components at the two layers (upper and lower) are activity decision factors. Fig. 3a emphasizes the activities of right grant and application based on the attributes of subject, object and context, and Fig. 3b the architecture of decision-point and policy-enforcement-point.



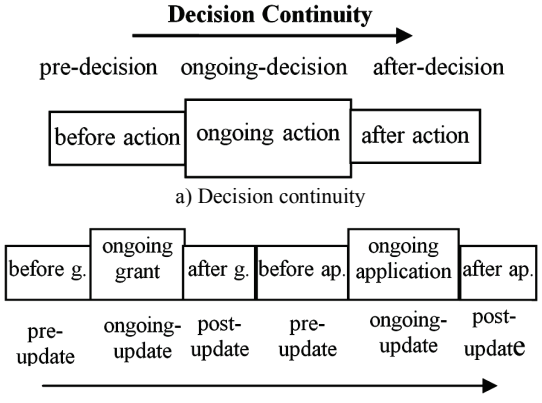
a) Logic of activities

b) Decision-activities

Figure 3 CUC action relationship model

3.4 Decision continuity and attributes mutability

Continuity and mutability are two important properties that information systems require. In UCON, the decision continuity and attributes mutability were shown in Fig. 2 of [8].



a) Decision continuity

b) Attributes mutability

Figure 4 CUC's continuity and mutability

CUC supports continuity and mutability as well. The decision continuity in CUC is similar in UCON (Figure 4a), but the attributes mutability in CUC is different from UCON's (Fig. 4b). As shown in Fig. 4b, the attributes mutability applies in both right grant and application stages. Because there are three classes of attributes, belonging to the subjects, objects and contexts, the attribute-update function should support the update of these three classes of attributes in two stages.

3.5 The CUC family

The UCON classification was based on the decision factors (authorizations, obligations, and conditions), attributes mutability, and decision continuity. Park and Sandhu enumerated 16 basic members of UCON family [5].

The criteria of CUC classification are similar to UCON except that the attributes of contexts can allow updates, and the time series is divided into two stages: right grant and application (see Tab. 1). In Tab. 1 a decision process denoted 0 means that no updates are needed (this case is corresponding to immutable attributes). Process denoted 1, 2, or 3 means updates are possible before, ongoing, or after the right grant and application respectively (these are corresponding to mutable attributes). Besides that, cases marked Y in Tab. 1 mean they are likely useful in practice, otherwise marked N. It has more members of family than UCON because CUC's usage process consists of two distinguished stages, the grant and application, and updates can be made in both processes.

We can formally define the CUC family members as UCON does. In this way, for example, CUC_{preA00} model is identical to UCONpreA0 defined in [5].

Definition 3 (CUCpreX00 model) The CUC_{preX00} model is made of the following components:
S, O, X, R, $ATT(S)$, $ATT(O)$, $ATT(X)$, and $preX$;
$Allowed(s, o, x, r) \Rightarrow preX(ATT(s), ATT(o), ATT(x), r)$.

Where $ATT(y)$ is the notation of attributes of entity y, and $y \in (S, O, X, s, o, x)$. The functional predicate $preX$ employs $ATT(s)$, $ATT(o)$, $ATT(x)$, and r to make a usage decision. And $allowed(s, o, x, r)$ indicates that right r is granted for subject s to apply to object o under the context x.

For completeness there should be many definitions. We give only a few definitions in this paper because it is straightforward to derive the rest by the similar way.

Table 1 The CUC family

Factors	Grant Mutability				Application Mutability			
	Immutable (0)	Update before (1)	Update ongoing (2)	Update after (3)	Immutable (0)	Update before (1)	Update ongoing (2)	Update after (3)
preA	Y	Y	N	Y	Y	Y	N	Y
onA	Y	Y	Y	Y	Y	Y	Y	Y
preB	Y	Y	N	Y	Y	Y	N	Y
onB	Y	Y	Y	Y	Y	Y	Y	Y
preX	Y	N	Y	N	Y	N	Y	Y
onX	Y	N	Y	N	Y	N	Y	Y

4 Formal description

Generally spoking, formal description needs to use certain language and methodology. Zhang et al. first conducted an UCON formal description using extended TLA (temporal logic of action) [8]. Then, Helge et al. used ITL (interval temporal logic), and Martinelli et al. used the Policy Language based on Process Algebra, respectively. For these methods each has its advantages [7].

[19] employed an algebra approach based on those introduced in [20] and [21] to describe formal syntax and semantics of attribute-based access control policy. Their method is simple, direct, and easy to understand and translate into computer algorithms language. We also adopt their methods.

4.1 Basic concepts

To formally describe the CUC model we first define some basic terms.

Definition 4 (Entity) An entity is an existing or real thing as a particular and discrete unit.

Definition 5 (Object) All entities to be the goal or end of an effort or activity in a system are called objects (passive entities).

Definition 6 (Subject) A subject is an entity that requests or executes access right on object(s) (active entity, e.g. user and process).

Definition 7 (Access Right) All usage permissions of objects that are granted to a subject in a system are called access rights (e.g. read, write, execute).

Definition 8 (Attribute) An attribute is a quality proper to a particular event or thing.

Each subject has a set of attributes, and each attribute of a subject has a value. So does each object and each context.

Assuming an entity named as *eName* with an attribute named as *atb*, the value of *atb* is denoted as *eName.atb* and $eName.atb \in domain(eName)$, where $domain(eName)$ is the value domain of *eName*, and $null \notin domain(eName)$. That *eName.atb* is equal to *null* indicates that *eName* does not have the attribute *atb*, or the value of *atb* has not to be assigned after *eName* creation.

4.2 The CUC Model

In this section for convenience the following notations are adopted:

S – Set of subjects, s – a particular subject;
O – Set of objects, o – a particular object;
X – Set of contexts, x – a particular context;
R – Set of rights, r – a particular right;
D – Set of decision, d – a particular decision, and $d \in (A, B)$, where A is set of authorizations, B set of obligations.

Definition 9 (Attribute-value Pair) An attribute-value pair is an attribute and a value pair linked by a operator in the form $a\_id \circ val$, where a_id is the attribute identifier, val the alpha-numerical value, and the operator $\circ$ an element in the set of

$\{<, \leq, =, \geq, >, \in, \notin, \subset, \subseteq, \not\subset, \supset, \supseteq, dominate\}$.

Definition 10 (Evaluation circumstance) An evaluation circumstance e is a quadruple $(S_{avp}, O_{avp}, X_{avp}, r_{req})$, where Y_{avp} is a set of attribute-value pairs of entity Y, $Y \in (S, O, X)$, and r_{req} is the requested right.

Definition 11 (CUC policy(s)) A CUC policy $p = (S, O, X, R, d)$ specifies that a rights set R applied by a subjects set S to an objects set O is improved by the decision d according to the policy p in a contexts set X. Where S, O, and X are specified by multiple-attribute sets respectively, and decision $d \in (A, B)$

where, the multiple-attribute set was defined in [19]. The evaluation result of an entity multiple-attribute set Y in an evaluation circumstance e can be represented as $\|Y\|_e$.

Definition 12 (Applicable policy) A CUC policy $p = (S, O, X, R, d)$ is applicable in an evaluation circumstance e if: $\|S\|_e \wedge \|O\|_e \wedge \|X\|_e \wedge \|R\|_e = true$.

Definition 13 (Policy(s) evaluation result) For a given evaluation circumstance e, a set of CUC policy, denoted as $P = \{p_1, p_2, \ldots, p_n\}$, returns a decisions set D as the evaluation results: $D = \|P\|_e = \{ d \mid p_i = (S, O, X, A, d), i=1,\ldots,n; \|S\|_e \wedge \|O\|_e \wedge \|X\|_e \wedge \|R\|_e = true \}$.

All policies in the CUC policy set P are evaluated against e. A policy is applicable to e if all multiple-attribute sets of subjects, objects and contexts are within those specified by e and the requested right falls within the policy's right set of e. The decisions set of applicable policies is returned as the result of the policy set evaluation.

Definition 14 (Core CUC) A core CUC model is a system that can form CUC policies.

Definition 15 (Management function) The CUC management function, denoted by $m(S, O, X)$, is an attributes function mainly to update or meter attributes of subjects, objects and contexts, and can be called before, ongoing, and after exercising a CUC policy $p = (S, O, X, R, d)$.

As an option, the management function can be employed for post-process of usage control such as to analyze and store the decision in practice, the rationale of decision, and other information.

CUC management set $M = \{m_1, m_2, \ldots, m_k\}$ is a set of management functions $m_1, m_2, \ldots, m_k$.

Definition 16 (Complete CUC) A complete CUC model is a system that can form CUC policies and has management functions.

In this section by using algebra approach and beginning from semantics we define CUC system to form CUC policies. So the description should be called policy description too.

4.3 Model application

Definition 17 (Usage) A usage $U = (s, o, r, x)$ is a quadruple specifying the subject s apply right r to object o in the context x.

Definition 18(Usage control) Usage control is the full control of processes including getting and applying right r on object o in context x for subject s in a usage $U = (s, o, r, x)$.

A usage process consists of two stages: the authorization (also called right grant), and the application

of right. So does the usage control: decision of authorization, and decision of right application.

The process of authorization decision is a practice of a CUC policy $p = (S, O, X, R, d)$, where the d is a decision of authorization. The procedure is as follows:

Step 1: (before authorization) to update attributes of subject, object and context by calling management functions.

Step 2: (ongoing authorization) to authorize by using authorization predicate, and meter attributes by calling management functions.

Step 3: (after authorization) to update attributes of subject, object and context by calling management functions. If authorized, go into right application; else, quit.

The process of right application decision is a practice of a CUC policy $p = (S, O, X, R, d)$, where the d is a decision of right application. The procedure of single right-application decision is as follows:

Step 1: (before application) to update attributes of subject, object and context by calling management functions.

Step 2: (ongoing application) to authorize by using authorization predicate, and meter attributes by calling management functions.

Step 3: (after application) to update attributes of subject, object and context by calling management functions.

5 Comparison

5.1 UCON

Comparing UCON, in CUC there are three improvements: replacement of UCON conditions, two stage decision, and management function.

That the contexts introduced by CUC substitute the conditions of UCON is significant. One of the benefits is dynamic. Contrary to the immutable conditions of UCON, the contexts of CUC are dynamic or mutable. As a consequence of the activities of subjects and objects in usage transaction, the situation may be changed and reflects as context information which can be employed in usage decision process. CUC is more in conformity with the actual situations. In UCON, a decision of usage is based on the individual subject and object plus systemic or environmental conditions (limitations or constraints), that makes UCON insensitive to context; whereas in CUC, the conditions for making decision are checked involving multiple subjects and objects, and the evaluation of systems and environment takes more current circumstance into account, and it better adapts to the dynamic changes.

Another benefit is that the descriptions of CUC are similar to UCON's, so many of research results of UCON are suitable for CUC directly or after a slight modification.

Example 1. Multilevel security policies

Multilevel security policies are also called as BLP policies. CUC_{preA00} can be employed to represent this typical mandatory access control (MAC) policies.

Let S be the set of subject s, O the set of object o, and L a security labels lattice with dominance relation $\geq$. The presentation of multilevel security policies is identical to realization in [5] by using $UCON_{preA0}$.

Example 2. ACL policies

The ACL policies can be presented by CUC_{pre00} and also identical to realization made in using $UCON_{preA0}$ [5].

Finally, Conditions in UCON are not characterized by its attributes, and contexts in CUC are. CUC is truly attributes-based.

Usage decision is divided into two stages, the authorization stage and the application of right stage. The two-stage control is more meticulous, and its logic more clear. For example, let's consider a case of usage. A file server supports 1000 files download concurrently with a limit that one user can only download 3 files at the same time. Supposing 400 users are downloading 999 files, the 401st authorized user tries to download 2 files, and the second one will fail. It is a system capability issue in CUC. But in original UCON, it may be explained as a system capability issue or a problem of insufficient user rights.

In CUC the authorization and right-application are ordered, that is, the authorization is implemented first, and after the successful authorization right-application kicks in. It truly meets the objectives and requirements of information security: confidentiality, integrity and availability. Users must be authorized first, because confidentiality, integrity and availability are for authorized users only [19].

The management block makes the attributes-update explicit which is implicit in UCON. It not only separates the functions of the modules more clearly, but also combines the management and control to some extent. It is consistent with the development trend of information security.

5.2 Others

In this section we try to compare the CUC with some approaches proposed by current researches to improve the UCON model.

Zhao et al. (2007) proposed a new access model called as TUCON (Times-based Usage Control) for prevention of digital resources abuse [9]. In TUCON a time variable is introduced into UCON, and maximum times defined as consumption constraints. Zhao et al. approach is implemented easily by defining a new attribute of context in CUC.

Zhang Hongjun (2009) proposed Geography Usage Control (GEO-UCON) model to deal with GEO DBMS access control [10]. In GEO-UCON a geospatial factor is added into UCON to ensure data security in location-based services and mobile applications. Zhang's approach can be dealt with by adding geospatial attribute of context into CUC.

Bai et al. (2011) extended usage control model to context-aware in mobile computing environments [11]. Bai et al. introduced two new components into UCON model: contexts and states. The new model called as ConUCON takes these new components plus obligations on access decisions. CUC model is simpler and more general-purpose than ConUCON model. Situations information is included by context information, and the states are the results of contexts evaluation. As a model, CUC can do these ConUCON can do.

Almutairi and Siewe (2011) proposed another context-aware usage control model in pervasive

computing systems [12]. Employing context information, CA-UCON can invoke some special actions to adapt to the environments changes. The approach for CA-UCON to employ contexts is different from CUC, and not general-purpose.

6 Conclusion and future work

In this paper an important model of the access control, UCON, was analysed in-depth, and its deficiency to capture the situation and reflect the dynamic nature pointed. The concept of action context and the principle of security context were introduced, and the generalized system and environment in UCON replaced by the context to solve the dynamic deficiency; management functions were also introduced to combine the management and control. The extended UCON is contextual usage control (CUC), and consists of two functional blocks, the management block manages attributes of subjects, objects and contexts, and the control block governs the usage. There are some loose couples between the management and control blocks. After the conceptual description of CUC, an algebra approach was used for formal description of CUC syntax and semantics.

For CUC more work is needed. Firstly, it is a concept model, its properties of CUC models are in open, and the formal description is essential. Secondly, there still are some restrictions. For example, the management functions can be used to remove existing attributes, but the loose couple restrict the changes do not bring on changes of the access control rules and algorithms in CB. Finally, the implementation issues also need to be further studied.

7 References

[1] Lampson, W. Protection. // Proceedings of the 5th Princeton Conference on Information Sciences and Systems, 1971, pp. 437-443.

[2] Luo, W.; Liu, J.; Dai, Z.; Han, L. Foundation of Information Security Applications. Chongqing University Press, Chongqing, 2005. (in Chinese).

[3] Park, J.; Sandhu, R. Towards usage control models: Beyond traditional access control. // Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, 2002, pp. 57-64.

[4] Park, J. Usage control: A unified framework for next generation access control. // PhD Thesis, George Mason University, 2003, pp. 45-46.

[5] Park, J.; Sandhu, R. The UCONABC usage control model. // ACM Transaction on Information and System Security, 7, 1(2004), pp. 128-174.

[6] Sandhu, R.S.; Park, J. Usage control: A vision for next generation access control. // LNCS 2776, 2003, pp. 17-31.

[7] Lazouski, A.; Martinelli, F.; Mori, P. Usage control in computer security: A survey. // Computer Science Review, 4, 2(2010), pp. 81-99.

[8] Zhang, X.; Parisi-Presicce, F.; Sandhu, R. Formal Model and Policy Specification of Usage Control. //ACM Transactions on Information and System Security, 8, 4(2005), pp. 351-387.

[9] Zhao, B.; Sandhu, R.; Zhang, X.; Qin, X. Towards a Times-Based Usage Control Model. // LNCS 4602, 2007, pp. 227-242.

[10] ZHANG Hong-jun. Study and application of special access control model based on UCON. // Master Thesis, Jiangshu University, 2009, pp. 24-26. (In Chinese)

[11] Bai, G.; Gu, L.; Feng, T.; Guo, Y.; Chen, X. Context-Aware Usage Control for Android. // LNICST 50, 2010, pp. 326-343.

[12] Bai, G.; Yan, L.; Gu, L.; Guo, Y.; Chen, X. Context-aware usage control for web of things. // Security and Communication Networks. The Web version (2012), 5. http://onlinelibrary.wiley.com/doi/10.1002/sec.424/full.

[13] Almutairi, A.; Siewe, F. CA-UCON: a context-aware usage control model. // Proceedings of the 5th ACM International Workshop on Context-Awareness for Self-Managing Systems, 2011, pp. 38-43.

[14] Soylu, A.; Causmaecker, P.; Desmet, P. Context and Adaptivity in Pervasive Computing Environments: Links with Software Engineering and Ontological Engineering. // Journal of Software, 4, 9(2009). pp. 992-1013.

[15] Baldauf, M.; Dustdar S.; Rosenberg, F. A survey on context-aware systems. // International Journal of Ad Hoc and Ubiquitous Computing 2, 26(2007), pp. 53-60.

[16] Zimmermann, A.; Lorenz, A.; Oppermann, R. An Operational Definition of Context. // LNAI 4635, 2007, pp. 558-571.

[17] Dey, A. Understanding and Using Context. // Personal Ubiquitous Computing, 5, 1(2001), pp. 4-7.

[18] Frege, G. The Foundations of Arithmetic. Trans. J. L. Austin. Second Revised Edition. Northwestern University Press, Illinois, 1980.

[19] Shu, C.; Yang, E.; Arenas, A. Detecting Conflicts in ABAC Policies with Rule- reduction and binary-search Techniques. // IEEE International Symposium on Policies for Distributed Systems and Networks, 2009, pp. 182-185.

[20] Bonatti, P.; Vimercati, S.; Samarati, P. An Algebra for Composing Access Control Policies. // ACM Transactions on Information and System Security, 5, 1(2002), pp. 1-35.

[21] Wimmer, M.; Kemper, A.; Rits, M.; Lotz, V. Consolidating the Access Control of Composite Applications and Workflows. // LNCS 4127, 2006, pp. 44-59.

Authors' addresses

PhD. Xiaofeng Luo
Mathematics College, Jingjiang College, Sichuan University
Chengdu 610064, China
E-mail: xluo_f@aliyun.com

PhD. Lin Li
Computer Science College, Sichuan University
Chengdu 610064, China
E-mail: lilin@scu.edu.cn

Prof. Wanbo Luo
Computer Science College, Sichuan University
Chengdu 610064, China
E-mail: wbluo@scu.edu.cn