On Mutual Support of Modern and Traditional Access Control Models with UCON and BLP as Case Study

Hui Feng, Wenchang Shi, Zhaohui Liang and Bin Liang
School of Information, Renmin University of China, Beijing 100872, China
Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China), Ministry of Education, Beijing 100872, China

Abstract—Access control is essential to computer security, especially in an open, distributed, networked communication environment. Modern access control model such as UCON aims at accommodating general requirements. Traditional one such as BLP focuses on specific properties, e.g. confidentiality. Both of these two realms have their limitations. Taking UCON and BLP as case study, this paper explores mutual support of modern and traditional access control models. It investigates BLP’s adaptable characteristic in the UCON perspective. First, it constructs properties in the UCON language to manifest the BLP adaptability, which shows that the BLP adaptability can be ensured to function correctly by the UCON framework. Further, it proposes a formal specification for the BLP adaptability under the UCON framework with the Temporal Logic of Actions, which demonstrates that the BLP adaptability is in good consistency with the UCON model. The significance of the paper is twofold. On the one hand, it exhibits that adaptable quality of the traditional BLP model may be ensured theoretically by the philosophy of modern access control. On the other hand, it enriches the real sense of modern access control models by strengthening the power of traditional access control models.

Keywords—Access Control, Usage Control, UCON, BLP, Temporal Logic of Actions, TLA

I. INTRODUCTION

Without access control, there would be no protection for computer systems. In today’s open, distributed, networked communication environment, access control is even more important than ever. Traditional access control models primarily handle static authorization. That is to say, once an access right is granted, it is usually not easy to be handled again any more. In recent years, J. Park and R. Sandhu proposed a dynamic access control framework, called Usage Control, or UCON [1,2], to deal with the limitations of traditional access control models. The UCON model attracts world-wide attention and is regarded as the next generation access control model [3-7].

Although traditional access control models have their disadvantage, they are irreplaceable in handling issues they target at. Among a variety of traditional access control models, the BLP model [8] is a classical one that provides Multi-Level Security (MLS) support. Large amounts of work has been done to improve the BLP model or to investigate its application in real systems. In order to improve the practicability of the BLP model, in our previous work [9-11], we investigated its capability of flexibly adapting to an application situation according to a user’s behavior when the application is on going. In a conventional way, we proved that the BLP model can be enforced to shows dynamic adaptability with an appropriate approach, which we call the ABLP approach [9]. Unfortunately, the BLP model itself is not expressive enough to ensure such adaptability.

As a typical modern access control model, UCON is said to be able to cover traditional access control models. But it is a relatively higher level model, which is not suitable for describing specific access rules. Very limited work can be found that shows UCON’s support for any specific traditional model in detail.

In order to utilize the strength of both modern and traditional access control models and avoid their weakness, this paper endeavors to explore mutual support of modern and traditional access control models by using Temporal Logic of Actions (TLA) [12] as theoretical vehicles. It creates a case study that tries to build theoretical foundation for the adaptability of traditional BLP with modern UCON.

The rest of this paper is organized as follows. Section 2 briefly introduces background information. Section 3 constructs UCON properties for BLP adaptability and builds a formalized model. In Section 4, we use the TLA to develop logic specifications for BLP adaptability under the UCON framework. In Section 5, we analyze related work. Section 6 gives a short discussion and concludes the paper.

II. THE PRELIMINARIES

This section introduces brief background knowledge about the ABLP approach, the UCON_{ABC} model and the TLA logic that is necessary to understand the presentation of the later sections.

A. The ABLP Approach

The BLP model control accesses of subjects to objects based on sensitivity levels. Common methods enforce the BLP model with static sensitivity levels. The ABLP approach provides a way to enforce the BLP model, which allows appropriate tuning of subjects’ current sensitivity levels but not violate the BLP axioms. It adds more cases that allow access and supports attributes update, thus is dynamic and more flexible. It introduces two layers of access decisions. In the outer layer, it makes normal decisions. If it
fails, inner layer decisions will be made. So there are two allowed access cases. Along with a decision and access, some attributes of subjects may be updated. In [9-11], it is explained in detail. Some symbolic notations we use later will be defined explicitly where they appear, so they are not listed here.

B. The UCON$_{ABC}$ Model

Compared with traditional access control models, the UCON model has three distinguished properties. First, it includes three decision factors: Authorization, oBligation and Condition. So, the core part of the UCON model is called UCON$_{ABC}$. Authorizations are predicates based on subjects and objects attributes. oBligations are operations that have to be done. Conditions are rules based on system attributes. Second, it supports continuity of decision. That is to say, it not only can make access decision before access, but does additional checks while accessing when necessary. Third, it supports mutability of attributes. Updates of subjects and objects attributes may be enforced before, during and after access. The UCON$_{ABC}$ has six basic decision ways: pre-authorization, on-authorization, pre-obligation, on-obligation, pre-condition and on-condition. The former four ways may include attributes update.

C. The TLA Logic

The Temporal Logic of Actions (TLA) is the logic for specifying and reasoning about systems and their properties. It is extremely powerful, both in principle and in practice. The elements of the TLA logic language include system, behavior, states, actions and predicates. The TLA expresses relations among them through temporal operations and mathematic operations. For a behavior of a system, an action is performed or a predicate is true in a state, then the system goes into the next state. General TLA and its extended form are described explicitly in [12, 3, 4].

III. Construction of UCON Properties for BLP Adaptability

We discuss UCON properties of the BLP Model through the ABLP approach. In [11], the access control policies enforced by ABLP contain three access rights, which are read, append and write. In this section, we construct BLP’s UCON properties and build a formalized model to describe the working of the ABLP in the UCON framework. We go in the order of the three access rights.

A. Components and Notations

Components used in our presentation include subjects, subject attributes, objects, object attributes, rights, access matrix and function sets. Notations used for them are as follow:

- $S, ATT(S), O, ATT(O), R$
- $M, F$
- $ATT(S): \{f_i(S), f_i(S, L_{RH}, L_{WL})\}$
- $ATT(O): \{f_i(O)\}$
- $R: r, a, w$

Subjects($S$): Entities that actively exercise certain access rights, such as human beings or processes.

Subject Attributes ($ATT(S)$): Properties of subjects that are used for access decision. In the ABLP approach, there are four subject attributes: subject’s sensitivity level ($f_i(S)$), subject’s current sensitivity level ($f_i(S)$), maximum sensitivity level of objects that have been read by the subject ($L_{RH}$) and minimum sensitivity level of objects that have been written by the subject ($L_{WL}$).

Objects($O$): entities that subjects hold access rights on, such as files or processes.

Objects Attributes ($ATT(O)$): Properties of objects that are also used for usage decision. In ABLP, an object has an attribute of sensitivity level ($f_i(O)$).

Rights ($R$): privileges that a subject exercises on an object. In our model, three kinds of rights are used: read ($r$), append ($a$) and write ($w$).

Access matrix ($M = [M_{ij}]_{m \times n}$): Defines a matrix composed of access rights sets that subjects hold on objects. Its element $M_{ij}$ means the set of access rights that subject $S_i$ holds on $O_j$.

Function sets ($F = \{f_i, f_o, f_w\}$): The triple ($f_i, f_o, f_w$) is composed of three functions. $f_i$ is a function that calculates a subject’s sensitivity level and maps the subject to the sensitivity level. $f_o$ is a function that calculates a subject’s current sensitivity level and maps the subject to the current sensitivity level. $f_w$ is a function to get an object’s sensitivity level and maps the object to the sensitivity level.

Sensitivity levels are described by sensitivity labels. These two terms are often used interchangeably.

B. Formalized Model for Application of UCON$_{preA}$ on ABLP

In the ABLP approach, there are two allowed access cases for each access right. We develop a model for all the cases. All the allowed access cases are defined in [11]. We only describe allowed cases, because if an access request does not satisfy allowed access requirements, it will be denied. Since ABLP involves pre-authorization and pre-updates, we utilize pre-authorization model with pre-update attributes to build the model. UCON$_{preA}$ is the UCON model for pre-authorization, according to which we will model ABLP.

The general access process in ABLP is: when an access request comes, the system uses subject attributes, object attributes and rights to make access decisions. If formulas composed of attributes and rights are false, the access request is denied, else if formulas are true, updates are enforced and then the access is allowed by the system.

1) Modeling the read Access Right Authorization
The first allowed case is constructed as:

allowed($s_i, o_j, r$) $\Rightarrow$ ($r$ in $M_{ij}$) $\land$ ($f_i(s_i) \geq f_o(o_j)$) \hspace{1cm} (1)

preupdate($L_{RH}$): $L_{RH} = \gamma(L_{RH}, f_i(o_j))$

The second allowed case is constructed as:

allowed($s_i, o_j, r$) $\Rightarrow$ ($r$ in $M_{ij}$) $\land$ ($f_i(s_i) \geq f_o(o_j)$)

$\land$ ($f_o(s_i) \geq f_o(o_j) \land (L_{WL} \geq f_o(o_j)$) \hspace{1cm} (2)

preupdate($f_o(s_i)$): $f_o(s_i) = \gamma(f_o(s_i), f_o(o_j))$

preupdate($L_{RH}$): $L_{RH} = \gamma(L_{RH}, f_o(o_j))$
Formula (1) is a pre-authorization decision. It indicates when \( f_i(s_j) \) dominates \( f_j(o_j) \), if \( s_j \) is allowed to read \( o_j \), then the following requirements have to be satisfied:

1) The right \( r \) is in the rights set of \( s_j \) on \( o_j \);
2) \( f_i(s_j) \) dominates \( f_j(o_j) \).

Before the access, there is an update of \( L_{\text{bl}} \). The function \( \gamma(m,n) \) means get the least upper bound of \( m \) and \( n \).

Formula (2) means pre-authorization decision. It indicates when \( f_i(s_j) \) dominates \( f_j(o_j) \) while \( f_i(s_j) \) does not dominate \( f_j(o_j) \), if \( s_j \) is allowed to read \( o_j \), then the following requirements are needed:

1) The right \( r \) is in the rights set of \( s_j \) on \( o_j \);
2) \( L_{\text{bl}} \) dominates \( f_j(o_j) \).

Before an access, there are updates of \( f_i(s_j) \) and \( L_{\text{bl}} \).

We use symbol \( \Rightarrow \) to indicate that pre-checking authorization’s true value is only necessary rather than sufficient to grant access, because authorization is part of the requirements needed for grant. There may be some other checks before accessing. In the following parts, we use symbol \( \Rightarrow \) for the same reason.

2) Modeling the append Access Right Authorization
The first allowed case is constructed as:

\[
\text{allowed}(s_j,o_j,w) \Rightarrow (w \in M_p) \land (f_i(s_j) = f_j(o_j)) \quad (3)
\]

\[
\text{preupdate}(L_{\text{bl}}) : L_{\text{bl}} = \gamma(L_{\text{bl}}, f_i(s_j))
\]

The second allowed case is constructed as:

\[
\text{allowed}(s_j,o_j,w) \Rightarrow (w \in M_p) \land (f_i(s_j) \geq f_j(o_j)) \land (f_i(s_j) \neq f_j(o_j)) \land (f_i(s_j) = f_j(o_j)) \land (f_i(s_j) \geq L_{\text{bl}}) \quad (4)
\]

\[
\text{preupdate}(f_i(s_j)) : f_i(s_j) = \lambda(f_i(s_j), f_j(o_j))
\]

\[
\text{preupdate}(L_{\text{bl}}) : L_{\text{bl}} = \gamma(L_{\text{bl}}, f_i(s_j))
\]

Formula (3) and Formula (4) are both pre-authorization decisions. Formula (3) indicates if \( s_j \) is allowed to append on \( o_j \), then

1) The right \( a \) is in the rights set of \( s_j \) on \( o_j \);
2) \( f_i(o_j) \) dominates \( f_j(s_j) \).

Formula (4) indicates if \( s_j \) is allowed to append on \( o_j \), then

1) The right \( a \) is in the rights set of \( s_j \) on \( o_j \);
2) \( f_i(o_j) \) dominates \( L_{\text{bl}} \).

Before access is exercised, attributes updates are performed. The function \( \lambda(m,n) \) means get the greatest lower bound of \( m \) and \( n \).

3) Modeling the write Access Right Authorization
The first allowed case is constructed as:

\[
\text{allowed}(s_j,o_j,w) \Rightarrow (w \in M_p) \land (f_i(s_j) = f_j(o_j)) \quad (5)
\]

\[
\text{preupdate}(L_{\text{bl}}) : L_{\text{bl}} = \gamma(L_{\text{bl}}, f_i(o_j))
\]

The second allowed case is constructed as:

\[
\text{allowed}(s_j,o_j,w) \Rightarrow (w \in M_p) \land (f_i(s_j) \geq f_j(o_j)) \land (f_i(s_j) \neq f_j(o_j)) \land (f_i(s_j) \geq L_{\text{bl}}) \quad (6)
\]

\[
\text{preupdate}(f_i(s_j)) : f_i(s_j) = f_j(o_j)
\]

\[
\text{preupdate}(L_{\text{bl}}) : L_{\text{bl}} = \gamma(L_{\text{bl}}, f_i(o_j))
\]

\[
\text{preupdate}(L_{\text{bl}}) : L_{\text{bl}} = \lambda(L_{\text{bl}}, f_i(o_j))
\]

Formula (5) indicates when \( f_i(s_j) \) dominates \( f_j(o_j) \), if \( s_j \) is allowed to write \( o_j \), then

1) The right \( w \) is in the rights set of \( s_j \) on \( o_j \);
2) \( f_i(s_j) \) equals \( f_j(o_j) \).

Formula (6) indicates when \( f_i(s_j) \) dominates \( f_j(o_j) \), but \( f_i(s_j) \) is not equal to \( f_j(o_j) \), if \( s_j \) is allowed to write \( o_j \), then

1) The right \( w \) is in the rights set of \( s_j \) on \( o_j \);
2) \( L_{\text{bl}} \) dominates \( f_j(o_j) \) and \( f_j(o_j) \) dominates \( L_{\text{bl}} \).

Preupdates are performed by the system before access.

4) Modeling Rationale
What we constructed above in the UCON concepts is the adaptability of the BLP model, not the whole BLP model itself. The core of the BLP model is its *-property, which defines access rules for the read, append and write operations. That is why we modeled authorization for each of them respectively.

Formula (1), (3) and (5) are used to describe the original BLP authorizations. Formula (2), (4) and (6) are constructed to express the adaptable authorizations.

The BLP *-property uses a subject’s current sensitivity level to make access decisions. The BLP model does not stipulate how to determine the current sensitivity level for a subject. Most systems implemented the BLP model by using static current sensitivity levels. Our ABLP approach makes a subject’s current sensitivity level adaptable to real situations.

In order to obtain the adaptability, the ABLP introduces the \( L_{\text{bl}} \) and the \( L_{\text{bl}} \) attributes. For simplicity, they are defined as subject attributes. Actually, they are session attributes. They are used to record the read and write history of a session.

Changing \( L_{\text{bl}} \), \( L_{\text{bl}} \) and current sensitivity level appropriately without violating the BLP security policies constitutes the key part of the BLP adaptability. The previous subsections show that this can be modeled with the pre-authorization and pre-update concepts of the UCON model. In other words, without the support of the UCON methods, it is really hard to model the adaptability of the BLP model.
IV. Logic Specification of BLP Adaptability in UCON Framework

In section 3, we build a model for BLP adaptability in the language of UCON properties. Though it can generally express the principles of ABLP, the model is not precise enough. For example, the time sequence in which policies are enforced is not demonstrated. Since logic language is very powerful, we employ it to give a new specification for the previous model. In this section, we use the TLA logic and its extended form to construct BLP adaptability more precisely, so as to demonstrate the consistency of BLP adaptability in the UCON framework.

A. Primary Definitions

Corresponding with the elements in TLA, first we give some basic definitions needed in our following model.

Definition 1 (State): The BLP model is a state machine model. Access operations make the system transfer from one state to another. We divide states into six categories according to the stage an access process is in. Similar to [3-4], they are six states: initial state, requesting state, accessing state, denied state, revoked state and end state. Initial means a subject has not sent an access request. Requesting means an access request is generated and the subject is waiting for the system’s decision. Accessing is the state that a subject is accessing an object with a certain right. Denied means a subject is not allowed to access an object. Revoked is the state that the system revokes a subject’s right and its access is stopped. End is the state after a subject finishes its access. All states include substates except for initial state. That means some states are in the same one of the five states, but there are some differences, which do not influence the states. We define state function state(s, o, r) to map triple (s, o, r) to the six states.

Definition 2 (Action): Borrowing from [4], we define two kinds of actions. One is actions that change an access process’s state, including tryaccess, permitaccess, denyaccess, revokeaccess and endaccess. The other is actions that update subjects and objects attributes, including preupdate, onupdate and postupdate. If an action is finished, its value is true, else false. Actions and states may be depicted as Fig. 1.

Definition 3 (Predicates): Predicates are boolean expressions built from subjects, objects and system attributes. If the values of attribute variables fulfill the expression in a state, then we say the value of a predicate is true, else false.

Definition 4 (Behavior): A behavior is a sequence of states. There are three final states in the six states. They are end, denied and revoked. In general, there are three kinds of behavior terminated in three different final states. For example, a behavior terminated with end can be: (initial, requesting, requesting, accessing, accessing, end). Two requesting states are in this behavior, because these two requesting states are not all the same. In one state requesting, pre-authorization predicates are not checked. While in the other state requesting, predicates are fulfilled and the subject is waiting for the system to perform preupdates.

B. Logic Specification of the BLP Adaptability Model

Based on these elements, we use TLA to describe the former model in a new way. As before, we organize it in the order of access rights. For brevity, we construct all the specifications at first. Then we present explanations to them.

C. Logic Specification of read Access Right Authorization

The TLA logic specification for the first allowed case is as:

\[
\text{permitaccess}(s_i, o_j, r) \land (r \in M_f \land f_r(s_i, o_2) \land \text{preupdate}(L_{RH}))
\]

The TLA logic specification for the second allowed case is as:

\[
\text{permitaccess}(s_i, o_j, r) \land (r \in M_f \land f_r(s_i, o_2) \land f_r(s_i, o_2) \land L_{WL} \land \text{preupdate}(f_r(s_i)) \land \text{preupdate}(L_{RH}))
\]

\[
\text{preupdate}(f_r(s_i)) : f_r(s_i) = \gamma(f_r(s_i, f_r(o_j)))
\]

\[
\text{preupdate}(L_{RH}) : L_{RH} = \gamma(L_{RH}, f_r(o_j))
\]

D. Logic Specification of append Access Right Authorization

The TLA logic specification for the first allowed case is as:

\[
\text{permitaccess}(s_i, o_j, a) \land (a \in M_f \land f_a(s_i, o_2) \land \text{preupdate}(L_{WL}))
\]

The TLA logic specification for the second allowed case is as:

\[
\text{permitaccess}(s_i, o_j, a) \land (a \in M_f \land f_a(s_i, o_2) \land L_{RH} \land \text{preupdate}(f_a(s_i)) \land \text{preupdate}(L_{WL}))
\]

\[
\text{preupdate}(f_a(s_i)) : f_a(s_i) = \lambda(f_a(s_i, f_a(o_j)))
\]

\[
\text{preupdate}(L_{WL}) : L_{WL} = \lambda(L_{WL}, f_a(o_j))
\]

E. Logic Specification of write Access Right Authorization

The TLA logic specification for the first allowed case is as:
permitaccess(s, o, w) → ◇(tryaccess(s, o, w)) ∧
◇((w ∈ M ∧ f_s(s) = f_o(o) ∧ preupdate(L_{RHi})) ∧ preupdate(L_{WHi}))

The TLA logic specification for the second allowed case is as:
permitaccess(s, o, w) → ◇(tryaccess(s, o, w)) ∧
◇((w ∈ M ∧ f_s(s) ≥ f_o(o) ∧ f_s(s) = f_o(o)) ∧ L_{WHi} ≥ f_o(o) ∧ f_o(o) ≥ L_{RHi} ∧ preupdate(f_s(s)) ∧ preupdate(L_{RHi}) ∧ preupdate(L_{WHi}))

preupdate(f_s(s)) : f_s(s) = f_o(o)
preupdate(L_{RHi}) : L_{RHi} = γ(L_{RHi}, f_o(o))
preupdate(L_{WHi}) : L_{WHi} = λ(L_{WHi}, f_o(o))

F. Explanations to Logic Specifications

In the above specifications, action permitaccess(s, o, x, w) grants right x on o to s. Action tryaccess(s, o, x) means subject s generates a request to access o with the right x. Checking whether expressions built from attributes variables are true is pre-authorization decision. ◇ is a temporal operation, means once. In a states sequence of a system, a formula E → F implies if E is true in a state, then F must be true in a previous state. The state sequence that those formulas above are in is:
(initial, requesting_s, …, requesting_s, …, requesting_s, …, requesting_s, …, requesting_s, …, requesting_s, …, requesting_s, …).

Next, taking as a representative, we give detailed interpretation of the formula in the second allowed case with access right of write. If in the state “requesting_s” action permitaccess(s, o, w) is performed, then the following three actions have to be fulfilled:

1) In a previous state “initial”, action tryaccess(s, o, w) is performed by subject s.
2) In a previous state “requesting_s”, action pre-authorization decision is performed, predicate w ∈ M ∧ f_s(s) ≥ f_o(o) ∧ L_{WHi} ≥ f_o(o)
 ∧ f_s(s) = f_o(o) ∧ L_{WHi} ≥ f_o(o) ∧ f_s(s) = f_o(o)
 ∧ L_{WHi} ≥ f_o(o) ∧ preupdate(f_s(s)) ∧ preupdate(L_{RHi}) ∧ preupdate(L_{WHi})

G. Logic Specification Results

From the perspective of Temporal Logic of Actions, the above specifications are straightforward. Comparing with the model built in Section 3, we can easily see that these specifications actually map the BLP adaptability model from a relatively coarse style to an elaborate one. Naturally, they express the BLP adaptability correctly and in a precise way. Consequently, we may reach the following results:

Under the UCON framework and guided by the UCON concepts, the BLP adaptability can be modeled formally and precisely. Furthermore, the created model for BLP adaptability matches the UCON model very well.

V. RELATED WORK

The work of this paper studies modern access control models without ignoring the traditional ones. Since J. Park and R. Sandhu proposed the concept of Usage Control as the key spirit of next generation access control [2], a large amount of research has been being conducted with focus on its development and applications [3-7]. The authors themselves presented the core model of Usage Control as UCON_{ABC} [1]. Through comparison with traditional access control, digital rights management and so on, it is claimed that the UCON is capable of encompassing all of them. However, work that may show UCON’s ability to provide complete support for any traditional access control model is very limited. The BLP model is an important traditional access control model. As an application of the UCON_{ABC} model, [1] presented a brief expression of the BLP model with the UCON_{ABC} language. But that expression can even not convey the principle of the essential *-property of the BLP model. Most systems implementations of the BLP model showed that it is not flexible enough to reflect users’ changing operation intent in real world. Our previous work tried to endow the BLP model with efficient flexibility without sacrificing security quality [9-11]. Nonetheless, the insufficient expressiveness of the BLP model made it hard to model this flexibility. This paper overcomes this drawback depending on the UCON model. X. Zhang et al. developed a logic specification of the UCON model with TLA and proved that it has powerful expressing capability [3-4]. Our work in using the TLA logic was inspired by those of X. Zhang et al [3-4], but our motivation is different from theirs. Our goal is to seek theoretical support from the UCON model for the adaptability of the BLP model, while X. Zhang et al were to demonstrate the logical soundness of the UCON model. To our knowledge, no other people have presented similar work to ours.

VI. DISCUSSION AND CONCLUSION

The basic incentive of this paper is to model the BLP adaptability by using the UCON model as theoretical support. Because we found that insufficient theoretical capability of such traditional model as BLP hinders its own practical usefulness.

In our previous work, we proved that the BLP model can be made adaptable in real-world application situations in response to users’ behavior so long as certain attributes of
relevant entities can be updated properly. However, the BLP model itself is not powerful enough to ensure this adaptability. That is to say, we lacked strict modeling support for this adaptability.

In this paper, we try to find out whether the UCON model can be used to lay the theoretical foundation for the dynamic adaptability of the BLP model. First, we construct properties of the UCON concepts to manifest the BLP adaptability, which shows that the BLP adaptability can be ensured to function correctly by the continuity of decision measure of the UCON model. Further, we develop formal specifications for the BLP adaptability under the UCON framework with the Temporal Logic of Actions. The TLA logic specifications demonstrate that the BLP adaptability is of sound consistency under the UCON framework. To this point, we have demonstrated how UCON can provide theoretical support to ensure the BLP adaptability.

On the other hand, our work is motivated to explore the way to apply the modern UCON model to handle practical concrete access control issues. Due to relatively higher level characteristic of expressiveness, the UCON model has nothing to do with certain access control policies, hence is incapable of dealing with any specific access arbitration. Although the traditional BLP model has its inherent limitations, it can find its practical position in specific application scenario, that is, where control of confidentiality is essential. Modern access control model such as UCON can never replace traditional one such as BLP. A prospective thought should be to make modern models progress on the traditional ones, enabling these two styles of models to make mutual complement with each other’s advantages. In some sense, this paper develops an effective method for utilizing the UCON model to solve the multi-level security problem in a flexible way.

In summary, the work of this paper has two contributions. Taking BLP and UCON as an example of traditional and modern access control models respectively, first, it demonstrates critical theoretical support of modern access control models to traditional ones. Second, it enriches the real sense of modern access control models in solving practical problems by integrating traditional ones.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments that helped improve the presentation of this paper. The work of the paper was supported in part by National Natural Science Foundation of China (61070192, 91018008, 60873213, 60703103), National 863 High-Tech Research Development Program of China (2007AA01Z414), National Science Foundation of Beijing (4082018) and Open Project of Shanghai Key Laboratory of Intelligent Information Processing (IIPL-09-006).

REFERENCES