A Logic Specification for Usage Control

Xinwen Zhang, Jaehong Park
Francesco Parisi-Presicce, Ravi Sandhu
George Mason University
SACMAT 2004

Outline

• Introduction of UCON
• Temporal Logic of Action (TLA)
• Logic Model for UCON with TLA
• Specification of Authorization Core Models
• Obligation and Conditions
• Conclusions and Future Work

UCON

• UCON provides a general model beyond DRM and Trust management:
  – Digital Rights Management (DRM)
    • Mainly focus on intellectual property rights protection with architecture and mechanism level studies
  – Trust Management
    • Authorization for strangers’ access based on credentials
    • Lack of an abstract model with attribute-based.

OM-AM Layered Approach

What ?

Model

Architecture

Mechanism

OM-AM Framework

Usage Control System

Assurance

How ?

Objective

Policy Neutral

UCON_ext model

CRM/SRM, CDoI architectures

DRM technologies, Trusted computing, etc.

• Model examples: Access Matrix, Lattice-based model, Role-base access control model

UCON

• A unified framework for next generation access control
• A comprehensive model to represent the underlying mechanism of existing access control models and policies.
• Try to extend the limits of traditional access control models:
  – Authorization only – No obligation or condition based control
  – Identity based only – No attributes based support
  – Decision is made before access – No ongoing control
  – No consumable rights - No mutable attributes
  – Rights are pre-defined and granted to subjects

UCON Model

• Basic components:
  – Subjects and attributes
  – Objects and attributes
  – Rights
• Logically, UCON is a mapping from a set of \{subject/object attributes, right\} to \{true, false\}
• Usage control decisions are based on authorization, obligations, and conditions.
• Referred as UCON_{ext} model
Continuity and Mutability of UCON

- A single usage process has three phases
  - before access, during usage, and access
- Continuity: control decision can be checked before or during access
- Mutability: attribute updates can be performed before, during or after access
  - Pre-update, on-update, and post-update

Core Authorization Models

- According to the authorization control attribute update points, we have seven core authorization models:
  - preA0: control decision is determined before access, and there is no attribute update.
  - preA1: control decision and attribute update before access.
  - preA3: control decision is determined before access, and attribute update after access.
  - onA0: control decision is checked and determined during usage, and there is no attribute update.
  - onA1: control decision is checked and determined during usage, and there is attribute update before access.
  - onA2: control decision is checked and determined during usage, and there is attribute update during usage.
  - onA3: control decision is checked and determined during usage, and there is attribute update after usage.
- A real UCON system may be a hybrid of them.

Temporal Logic of Action

- Basic Terms:
  - Variables: x, y
  - Values: 5, “abc”
  - Constants
  - A state is an assignment of values to variables
- Functions: nonboolean expression with variables and constants
  - Semantically, a function is a mapping from states to values.
- State Predicates: boolean expression with variables and constants
  - Semantically, a predicate is a mapping from states to boolean.
- Actions: boolean expression with variables, primed variables, and constants
  - Semantically, an action is a function assigning a boolean to a pair of states (x, i), where x is the old state with variables, and i is the new state with primed variables.

TLA

- Behavior: a sequence of states
  `<s0, s1, s2, ...,

Semantics of an action A:

- Temporal operator: (always)

- Temporal Formula:

- Semantics:

Other temporal operators:

- “Eventually”:

- “Next”:

- “Until”:

Past temporal operators:

- Has-always-been, Once, Previous, Since

Outline

- Introduction of UCON
- Temporal Logic of Action (TLA)
- Logic Model for UCON with TLA
- Specification of Authorization Core Models in UCON
- Obligation and Conditions
- Conclusions and Future Work
Outline

- Introduction of UCON
- Temporal Logic of Action (TLA)
- Logic Model for UCON with TLA
- Specification of Authorization Core Models in UCON
- Obligation and Conditions
- Conclusions and Future Work

Logical Model of UCON: Attributes

- A state of UCON is an assignment of values to attributes:
  - Subject attributes: role, security clearance, credit amount, etc.
  - Object attributes: type, directory, etc.
  - System attributes: time, location, etc.
- A special system attribute:
  - state(s, o, r) = {initial, requesting, denied, accessing, revoked, end}
  - To specify the status of a single access process (s, o, r)
  - Authorization actions defined to change this state.

Logic Model of UCON: Predicates

- Predicates: boolean expression built from subject attributes, object attributes, and system attributes:
  - Unary predicates:
    - Alice.credit > $1000, file1.classification = "secure"
  - Binary predicates:
    - Dominate(Alice.clearance, file1.classification)
    - in((Bob, read), file2.ACL)
  - Ternary predicate permit(s, o, r):
    - usage control decision
    - True if s is allowed to access o with r.

Logic Model of UCON: Actions

- Two types of actions:
  - Actions performed by a subject
  - Actions performed by the system
- state(s, o, r) transition with actions:
Outline

- Introduction of UCON
- Temporal Logic of Action (TLA)
- Logic Model for UCON with TLA
- Specification of Authorization Core Models
- Obligation and Conditions
- Conclusions and Future Work

Specification of Core Models

- \( \text{preA}_1 \):
  \[
  p_1 \land \ldots \land p_n \rightarrow \text{permits}(s, a, r) \\
  \text{trace}(s, a, r) \\
  \rightarrow \Box (\text{permits}(s, a, r))
  \]

- Example 2: BLP model
  \[
  \text{dominate}(s, \text{clearance}, a, \text{classification}) \\
  \rightarrow \text{permits}(s, a, \text{read}) \\
  \text{trace}(s, a, \text{read}) \\
  \\
  \rightarrow \Box (\text{permits}(s, a, \text{read}))
  \]

- Example 3: DAC with ACL
  \[
  \text{in}(s, \text{I.D}, r, \text{auth}) \\
  \rightarrow \text{permits}(s, a, r) \\
  \text{trace}(s, a, r) \\
  \rightarrow \Box (\text{permits}(s, a, r))
  \]

Specification of Core Models

- \( \text{preA}_2 \):
  \[
  p_1 \land \ldots \land p_n \rightarrow \text{permits}(s, a, r) \\
  \text{trace}(s, a, r) \\
  \rightarrow \Box (\text{permits}(s, a, r))
  \]

- Example 4: DRM pay-per-use application
  \[
  (\text{Alice.credit} \geq \text{ebook1.value}) \\
  \rightarrow \text{permits}(\text{Alice, ebook1, read}) \\
  \text{trace}(\text{Alice, ebook1, read}) \\
  \\
  \rightarrow \Box (\text{permits}(\text{Alice, ebook1, read}))
  \]

Specification of Core Models

- \( \text{onA}_1 \):
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- Example 5: DRM membership-based application
  \[
  \text{subject: Alice, with attributes of ID and read.expire} \\
  \text{object: book1, with attributes of title and reading.expire} \\
  \]

- \( \text{onA}_2 \):
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- Example 6:
  \[
  \Box (\neg \text{Bob is employee} \\
  \land \text{Bob.expire cert is ACL}) \\
  \land \text{state}(\text{Bob, a, r}) = \\
  \text{accessing} \\
  \rightarrow \Box (\text{revokeaccess}(\text{Bob, r})))
  \]

- \( \text{onA}_3 \):
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- Example 7:
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- \( \text{onA}_4 \):
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- Example 8:
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- \( \text{onA}_5 \):
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]

- Example 9:
  \[
  \Box (\neg p_1 \land \ldots \land p_n \rightarrow (\text{trace}(s, a, r) \\
  \land \Box (\text{permits}(s, a, r))))
  \]
**Specification: an Example**

- Example 7: Resource-constrained access control
  - Limited number (10) of ongoing accessing for a single object
  - Object attribute: accessingS = [s] is accessing s
  - When 11th subject requesting new access, one ongoing accessing subject will be revoked.

  a. revocation by earliest usage will be revoked
  - Subject attribute: startTime
    1. true → permit(s,a,r)
    2. permit(a, s, a, r) → ⋈(preupdate(aaccessingS), where preupdate : aaccessingS → aaccessingS = aaccessingS \ s)
    3. ⋈(preupdate(aaccessingS), where preupdate : aaccessingS → aaccessingS = aaccessingS \ s) \ aaccessingS ∧ ⋈(s ∈ aaccessingS)
    4. ⋈(s ∈ aaccessingS)

  b. revocation by longest idle usage
  - Subject attribute: idleTime
    1. true → permit(s,a,r)
    2. permit(a, s, a, r) → ⋈(preupdate(aaccessingS), where preupdate : aaccessingS → aaccessingS = aaccessingS \ s)
    3. ⋈(preupdate(aaccessingS), where preupdate : aaccessingS → aaccessingS = aaccessingS \ s) \ aaccessingS ∧ ⋈(s ∈ aaccessingS)
    4. ⋈(s ∈ aaccessingS)

  c. revocation by longest total usage
  - Subject attribute: usageTime
    1. true → permit(s,a,r)
    2. permit(a, s, a, r) → ⋈(preupdate(aaccessingS), where preupdate : aaccessingS → aaccessingS = aaccessingS \ s)
    3. ⋈(preupdate(aaccessingS), where preupdate : aaccessingS → aaccessingS = aaccessingS \ s) \ aaccessingS ∧ ⋈(s ∈ aaccessingS)
    4. ⋈(s ∈ aaccessingS)

**Outline**

- Introduction of UCON
- Temporal Logic of Action (TLA)
- Logic Model for UCON with TLA
- Specification of Authorization Core Models
- Obligation and Conditions
- Conclusions and Future Work

**Conditions**

- Conditions are environment restrictions before or during usage.
- In UCON, a condition is a predicate built from system attributes, such as time and location.

**Example:**

\[(\text{condition}) \land (\text{time}) \land (\text{location})\]

\[\begin{align*}
\text{a} :: \text{a} & \land \text{b} \land \text{c} \\
\text{b} :: \text{d} \land \text{e} \land \text{f} \\
\text{c} :: \text{g} \land \text{h} \land \text{i}
\end{align*}\]
Outline

- Introduction of UCON
- Temporal Logic of Action (TLA)
- Logic Model for UCON with TLA
- Specification of Authorization Core Models
- Obligation and Conditions
- Conclusions and Future Work

Conclusions

- A logical model for UCON with:
  - States with:
    - subject attributes and values
    - Object attributes and values
    - System attributes and values
  - Predicates:
    - Authorization predicates built from subject and object attributes
    - Condition predicates built from system attributes
  - Actions:
    - Attribute update actions
    - Usage control actions
    - Obligation actions
  - Temporal formulas of usage control policies
- First-order logic specification of the UCON models with new features of:
  - Mutability
  - Continuacity

Future Work

- UCON:
  - Enrich UCON model, such as constraints, delegations
  - Administrative UCON model
    - Attribute management
    - Administrative policies
  - Expressive power and safety analysis for UCON
  - Concurrency of UCON
- Development of architecture and mechanism for UCON system