Usage Control (UCON) 
or 
ABAC on Steroids

Prof. Ravi Sandhu 
Executive Director and Endowed Chair

February 26, 2016

ravi.sandhu@utsa.edu 
www.profsandhu.com
Motivation

- Traditional access control models are not adequate for today’s distributed, network-connected digital environment.
  - Authorization only – No obligation or condition based control
  - Decision is made before access – No ongoing control
  - No consumable rights - No mutable attributes
  - Rights are pre-defined and granted to subjects
Motivation

- No access control model available to capture Digital Rights Management (DRM)
  - Control after dissemination
  - IPR protection

- Need for a unified model that can encompass traditional access control models, DRM and other enhanced access control models from recent literature
Usage Control (UCON)

- **Scope**
  - Encompass traditional access controls, trust management, digital rights management and more
  - For sensitive information protection, IPR protection, and privacy protection

- **Model**
  - General purpose, policy neutral models
  - Policy is assumed to be given to the system
  - Transaction based control
  - Existence of right is determined when access is attempted by a subject (no predefined access matrix)
  - Attribute-based access control
Usage Control (UCON)

Security Objectives

- Privacy Protection
- Intellectual Property Rights Protection
- Sensitive Information Protection
- Intellectual Property Rights Protection
- Sensitive Information Protection
- Trust Management

Security Architectures

- Server-side Reference Monitor (SRM)
- Client-side Reference Monitor (CRM)
- Traditional Access Control
- Usage Control

World-Leading Research with Real-World Impact!
Building UCON\textsubscript{ABC} Models

Continuity of Decisions

- Pre
- Usage
- Post

Mutability of Attributes

- Pre
- Ongoing
- Post

Continuity

Decision can be made during usage for continuous enforcement

Mutability

Attributes can be updated as side-effects of subjects’ actions

Subjects (S) → Rights (R) → Objects (O)

Subject Attributes (ATT(S)) → Obligations (B) → Conditions (C) → Object Attributes (ATT(O))
Building UCON_ABC Models

Continuity of Decisions
Decision can be made during usage for continuous enforcement

Mutability of Attributes
Attributes can be updated as side-effects of subjects’ actions

Usage Decisions

Before
pre
ongoing
N/A
After

Continuity

Mutability
Examples

- Long-distance phone (pre-authorization with post-update)
- Pre-paid phone card (ongoing-authorization with ongoing-update)
- Pay-per-view (pre-authorization with pre-updates)
- Click Ad every 30 minutes (ongoing-obligation with ongoing-updates)
- Business Hours (pre-/ongoing-condition)
<table>
<thead>
<tr>
<th></th>
<th>0(Immutable)</th>
<th>1(pre)</th>
<th>2(ongoing)</th>
<th>3(post)</th>
</tr>
</thead>
<tbody>
<tr>
<td>preA</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>onA</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>preB</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>onB</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>preC</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>onC</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

N : Not applicable
A Family of UCON\textsubscript{ABC} Core Models

(a) \[\text{UCON}_A \xrightarrow{\text{on}} \text{UCON}_B \xrightarrow{\text{on}} \text{UCON}_C\]

(b) \[\text{preA}_0 \xrightarrow{\text{on}} \text{preA}_1 \xrightarrow{\text{on}} \text{preA}_2 \xrightarrow{\text{on}} \text{preA}_3\]

(c) \[\text{preB}_0 \xrightarrow{\text{on}} \text{preB}_1 \xrightarrow{\text{on}} \text{preB}_2 \xrightarrow{\text{on}} \text{preB}_3\]

(d) \[\text{preC}_0 \xrightarrow{\text{on}} \text{onC}_0\]
- Online content distribution service
  - Pay-per-view (pre-update)
  - Metered payment (post-update)
• Pay-per-minutes (pre-paid Phone Card)
UCON\textsubscript{preA}: pre-Authorizations Model

- \textbf{UCON}_{\text{preA}0}:
  - S, O, R, ATT(S), ATT(O) and \textit{preA} (subjects, objects, rights, subject attributes, object attributes, and pre-authorizations respectively);
  - \textit{allowed}(s,o,r) \Rightarrow \textit{preA}(ATT(s),ATT(o),r)

- \textbf{UCON}_{\text{preA}1}:
  - \textit{preUpdate}(ATT(s)),\textit{preUpdate}(ATT(o))

- \textbf{UCON}_{\text{preA}3}:
  - \textit{postUpdate}(ATT(s)),\textit{postUpdate}(ATT(o))
$L$ is a lattice of security labels with dominance relation $\geq$

- clearance: $S \rightarrow L$
- classification: $O \rightarrow L$

$\text{ATT}(S) = \{\text{clearance}\}$

$\text{ATT}(O) = \{\text{classification}\}$

$\text{allowed}(s, o, \text{read}) \Rightarrow \text{clearance}(s) \geq \text{classification}(o)$

$\text{allowed}(s, o, \text{write}) \Rightarrow \text{clearance}(s) \leq \text{classification}(o)$
DAC in UCON: with ACL ($CON_{preA0}$)

- $N$ is a set of identity names
- $id : S \rightarrow N$, one to one mapping
- $ACL : O \rightarrow 2^{N \times R}$, $n$ is authorized to do $r$ to $o$
- $ATT(S) = \{id\}$
- $ATT(O) = \{ACL\}$
- $allowed(s,o,r) \Rightarrow (id(s), r) \in ACL(o)$
• $P = \{(o,r)\}$
• $ROLE$ is a partially ordered set of roles with dominance relation $\geq$
• $actRole: S \rightarrow 2^{ROLE}$
• $Prole: P \rightarrow 2^{ROLE}$
• $ATT(S) = \{actRole\}$
• $ATT(O) = \{Prole\}$
• $allowed(s,o,r) \Rightarrow \exists role \in actRole(s), \exists role' \in Prole(o,r), role \geq role'$
• $M$ is a set of money amounts
• $\text{credit: } S \rightarrow M$
• $\text{value: } O \times R \rightarrow M$
• $\text{ATT}(s): \{\text{credit}\}$
• $\text{ATT}(o,r): \{\text{value}\}$
• $\text{allowed}(s,o,r) \Rightarrow \text{credit}(s) \geq \text{value}(o,r)$
• $\text{preUpdate(credit(s))}: \text{credit}(s) = \text{credit}(s) - \text{value}(o,r)$
• Membership-based metered payment
  - $M$ is a set of money amount
  - $ID$ is a set of membership identification numbers
  - $TIME$ is a current usage minute
  - $member: S \rightarrow ID$
  - $expense: S \rightarrow M$
  - $usageT: S \rightarrow TIME$
  - $value: O \times R \rightarrow M$ (a cost per minute of $r$ on $o$)
  - $ATT(s): \{member, expense, usageT\}$
  - $ATT(o,r): \{valuePerMinute\}$
  - $allowed(s,o,r) \Rightarrow member(s) \neq \emptyset$
  - $postUpdate(expense(s)): expense(s) = expense(s) + (value(o,r) \times usageT(s))$
UN\textsubscript{onA}: ongoing-Authorizations Model

- **\textbf{UCON}_{onA0}**
  - \(S, O, R, ATT(S), ATT(O)\) and \(onA\);
  - \(allowed(s,o,r) \Rightarrow true\);
  - \(Stopped(s,o,r) \Leftarrow \neg onA(ATT(s),ATT(o),r)\)

- **\textbf{UCON}_{onA1}, UCON_{onA2}, UCON_{onA3}**
  - \(preUpdate(ATT(s)),preUpdate(ATT(o))\)
  - \(onUpdate(ATT(s)),onUpdate(ATT(o))\)
  - \(postUpdate(ATT(s)),postUpdate(ATT(o))\)

- **Examples**
  - Certificate Revocation Lists
  - revocation based on starting time, longest idle time, and total usage time
- **Free Internet Service Provider**
  - Watch Ad window (no update)
  - Click ad within every 30 minutes (ongoing update)
UCON\textsubscript{preB0} \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \textregistered \text registered

\begin{itemize}
  \item $S$, $O$, $R$, $ATT(S)$, and $ATT(O)$;
  \item $OBS$, $OBO$ and $OB$ (obligation subjects, obligation objects, and obligation actions, respectively);
  \item $preB$ and $preOBL$ (pre-obligations predicates and pre-obligation elements, respectively);
  \item $preOBL \subseteq OBS \times OBO \times OB$;
  \item $preFulfilled$: $OBS \times OBO \times OB \rightarrow \{true, false\}$;
  \item $getPreOBL$: $S \times O \times R \rightarrow 2^{preOBL}$, a function to select pre-obligations for a requested usage;
  \item $preB(s,o,r) = \Lambda (obs\_i,obo\_i,ob\_i) \in getPreOBL(s,o,r) \ preFulfilled(obs_i,obo_i,ob_i)$;
  \item $preB(s,o,r) = true$ by definition if $getPreOBL(s,o,r)=\emptyset$;
  \item $allowed(s,o,r) \implies preB(s,o,r)$.
\end{itemize}

\begin{itemize}
  \item Example: License agreement for a whitepaper download
\end{itemize}
• $S, O, R, ATT(S), ATT(O), OBS, OBO$ and $OB$;
• $T$, a set of time or event elements;
• $onB$ and $onOBL$ (on-obligations predicates and ongoing-obligation elements, respectively);
• $onOBL \subseteq OBS \times OBO \times OB \times T$;
• $onFulfilled$: $OBS \times OBO \times OB \times T \rightarrow \{true, false\}$;
• $getOnOBL$: $S \times O \times R \rightarrow 2^{onOBL}$, a function to select ongoing-obligations for a requested usage;
• $onB(s,o,r) = \bigwedge_{(obs_i,obo_i,ob_i,t_i) \in getOnOBL(s,o,r)} onFulfilled(obs_i,obo_i,ob_i,t_i)$;
• $onB(s,o,r) = true$ by definition if $getOnOBL(s,o,r) = \emptyset$;
• $allowed(s,o,r) \Rightarrow true$;
• $Stopped(s,o,r) \iff \neg onB(s,o,r)$.

• Example: Free ISP with mandatory ad window
• Location check at the time of access request
• Accessible only during business hours

**Usage Decision**

- Location check at the time of access request
- Accessible only during business hours

**Update of Attributes:** No-Update is possible
UCON\textsubscript{preC0}: pre-Condition model

- \( S, O, R, ATT(S), \text{ and } ATT(O) \);
- \( \text{preCON} \) (a set of pre-condition elements);
- \( \text{preConChecked}: \text{preCON} \rightarrow \{\text{true},\text{false}\} \);
- \( \text{getPreCON}: S \times O \times R \rightarrow 2^\text{preCON} \);
- \( \text{preC}(s,o,r) = \bigwedge_{\text{preCon}_i \in \text{getPreCON}(s,o,r)} \text{preConChecked}(\text{preCon}_i) \);
- \( \text{allowed}(s,o,r) \Rightarrow \text{preC}(s,o,r) \).

- Example: location checks at the time of access requests
**UCON\textsubscript{onC0}: ongoing-Condition model**

- \( S, O, R, ATT(S), \text{ and } ATT(O) \);
- \( \text{onCON} \) (a set of on-condition elements);
- \( \text{onConChecked: } \text{onCON} \rightarrow \{\text{true}, \text{false}\} \);
- \( \text{getOnCON: } S \times O \times R \rightarrow 2^{\text{onCON}} \);
- \( \text{onC}(s,o,r) = \Lambda_{\text{onCon}_i \in \text{getOnCON}(s,o,r)} \text{onConChecked}(\text{onCon}_i) \);
- \( \text{allowed}(s,o,r) \Rightarrow \text{true} \);
- \( \text{Stopped}(s,o,r) \leftarrow \neg \text{onC}(s,o,r) \)

- Example: accessible during office hour
- **Free ISP**
  - Membership is required *(pre-authorization)*
  - Have to click Ad periodically while connected *(on-obligation, on-update)*
  - Free member: no evening connection *(on-condition)*, no more than 50 connections *(pre-update)* or 100 hours usage per month *(post-updates)*
Beyond the $\text{UCON}_{\text{ABC}}$ Core Models

- **Objects (O)**
- **Consumer Subjects (CS)**
- **Provider Subjects (PS)**
- **Identifiee Subjects (IS)**

Usage Controls:
- Parallel
- Serial
Logic Model of UCON

- Actions: boolean expressions built from attributes in two states.
  - Alice.credit' = Alice.credit - $50.0
- Two types of actions:
  - Control actions: change the state of single usage process
    - Actions performed by the subject
    - Actions performed by the system
  - Obligation actions:
    - Actions that have to be performed before or during an access.
    - May or may not be performed by the requesting subject and on the target object.
Summary of UCON

- Coined the concept of Usage Control for modern computing systems.
- Developed a family of UCON_{ABC} core models for Usage Control (UCON) to unify traditional access control models, DRM, and other modern enhanced models.
- UCON_{ABC} model integrates authorizations, obligations, conditions, as well as continuity and mutability properties.
Discuss Pretschner 2006 paper