Role Delegation for a Distributed, Unified RBAC/MAC*

M. Liebrand and H. Ellis
Computer and Info. Science Dept.
Rensselaer at Hartford
Hartford, CT 06120-2991
mark.liebrand@snet.net, heidic@rh.edu
860.548.5387, fax: 860.547.0868

C. Phillips, S. Demurjian, T.C. Ting and J. Ellis
Computer Science & Engineering Department
The University of Connecticut
Storrs, Connecticut 06269-3155
Charles.Phillips@usma.edu, {steve, ting}@engr.uconn.edu
http://www.engr.uconn.edu/~steve/DSEC/dsec.html

*Supported in part by a grant from the GE Foundation to UConn’s Engineering School.
Introduction/Motivation
Distributed Application Access

- Premise: **Artifacts** - set of
  - DB, Legacy, COTS, GOTS, Each w/ API

- Premise: **Users**
  - New and Existing
  - Utilize Artifact APIs

- **Distributed Application**
  - **Artifacts** + **Users**

- Can we Control **User**
  - Access to **Artifact** APIs (Methods) by …
    - Role (who)
    - Classification (MAC)
    - Time (when)
    - Data (what)
Overview of Remainder of Presentation

- **Background**
  - Security Model Definitions
  - Examples of Definitions

- **Role Delegation**
  - Security Model Extensions for Role Delegation
  - Examples of Role Delegation
  - See Paper for Additional Analyses/Discussion Regarding Enforcement Framework

- **Security Assurance and Delegation Enforcement**
  - Assurance at Design and Runtime
  - Security Enforcement Framework
  - Screen Shots of Various Security Tools

- **Conclusions and Ongoing/Future Work**
Security Model Definitions
Lifetimes and MAC Concepts

- **Definition 1**: A *lifetime*, $LT$, is a Discrete Time Interval $[LT.st, LT.et]$ with $LT.et > LT.st$
  - $LT.st$ (start time) or $LT.et$ (end time) is a tuple (day, month, year, hour, minute, second)
  - where $x \succ y$ means $x.LT.st \geq y.LT.st$ and $x.LT.et \geq y.LT.et$

- **Definition 2**: Relevant MAC Concepts are:
  - A *sensitivity level*, $SLEVEL$, $SLEVEL = \{U,C,S,T\}$
    - unclassified ($U$) - no impact; confidential ($C$) causes some damage; secret ($S$), causes serious damage; top secret ($T$) causes exceptionally grave damage
  - $SLEVEL$s form a hierarchy: $U < C < S < T$
  - *Clearance* ($CLR$) is $SLEVEL$ given to users
  - *Classification* ($CLS$) is the $SLEVEL$ given to entities (roles, objects, methods, etc.)
Security Model Definitions
Methods, Services, and Resources

- **Definition 3**: A distributed application, DAPPL, is composed of a set of software/system resources (e.g., a legacy, COTS, DB, etc.), each composed of a set of services, which in turn are each composed of a set of methods.

- **Definition 4**: Every method is registered as:

\[
M_{ijk} = [M_{ijk}^{Name}, M_{ijk}^{LT}, M_{ijk}^{CLS}, M_{ijk}^{Params}]
\]

- **Definition 5**: Every service is registered as:

\[
S_{ij} = [S_{ij}^{Name}, S_{ij}^{LT}, S_{ij}^{CLS}]
\]

- **Definition 6**: Every resource is registered as:

\[
R_i = [R_i^{Name}, R_i^{LT}, R_i^{CLS}]
\]
Global Command and Control System (GCCS) Resource/Service/Methods

GCCS Resource with Two Services - Classification Precedes Method

**Joint Service with Methods:**

<table>
<thead>
<tr>
<th>Service Method</th>
<th>a.k.a</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S) Weather (Token);</td>
<td>METOC</td>
</tr>
<tr>
<td>(S) VideoTeleconference (Token, fromOrg, toOrg);</td>
<td>TLCF</td>
</tr>
<tr>
<td>(S) JointOperationsPlanning (Token, CrisisNum);</td>
<td>JOPES</td>
</tr>
<tr>
<td>(S) CrisisPicture (Token, CrisisNum, Grid1, Grid2);</td>
<td>COP</td>
</tr>
<tr>
<td>(S) TransportationFlow (Token);</td>
<td>JFAST</td>
</tr>
<tr>
<td>(S) LogisticsPlanningTool (Token, CrisisNum);</td>
<td>LOGSAFE</td>
</tr>
<tr>
<td>(S) DefenseMessageSystem (Token);</td>
<td>DMS</td>
</tr>
<tr>
<td>(T) NATOMessageSystem (Token);</td>
<td>CRONOS</td>
</tr>
</tbody>
</table>

**Component Service with Methods:**

<table>
<thead>
<tr>
<th>Service Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(S) ArmyBattleCommandSys (Token, CrisisNum);</td>
<td>ABCS</td>
</tr>
<tr>
<td>(S) AirForceBattleManagementSys (Token, CrisisNum);</td>
<td>TBMCS</td>
</tr>
<tr>
<td>(S) MarineCombatOpnsSys (Token, CrisisNum);</td>
<td>TCO</td>
</tr>
<tr>
<td>(S) NavyCommandSystem (Token, CrisisNum);</td>
<td>JMCIS</td>
</tr>
</tbody>
</table>
Security Model Definitions
User Roles, UR List, Users, User List

- **Definition 7:** A *user role*, $UR$, representing a set of responsibilities for an application, is defined as:
  \[ UR = [ UR^{Name}, UR^{LT}, UR^{CLS} ] \]
  - $[CDR\_Crisis1, UR^{LT}, T]$

- **Definition 8:** A *user-role list*, $URL$, is the set of $r$ unique roles that have been defined for DAPPL.

- **Definition 9:** A *user*, $U$, who will be accessing the DAPPL via a client application, is defined as:
  \[ U = [ U^{UserId}, U^{LT}, U^{CLR} ] \]
  - $[Colonel\_DoGood, U^{LT}, S]$

- **Definition 10:** A *user list*, $UL$, is the set of $u$ users that have been defined for DAPPL.
Security Model Definitions
Signature and Time Constraints

- **Definition 11:** A signature constraint, \( SC \), is a boolean expression defined on method’s signature to limit the allowable values on the parameters.

  \[
  \text{CrisisPicture (Token, CrisisNum, Grid1, Grid2);}
  \text{Grid1 < NA20 and Grid2 < NC40}
  \]

- **Definition 12:** A time constraint, \( TC \), is a lifetime that represents when a method can be assigned to a user role (or invoked by a user) or when a user is allowed to play a role.

  \[
  \text{ArmyBattleCommandSys (Token, CrisisNum);}
  \text{TC = [10dec00, 16feb01] – shorthand notation}
  \]
Security Model Definitions
MAC Constraint/User Role Authorizations

- **Definition 13:** A *mandatory access control constraint*, MACC, is the domination of the SLEVEL of one entity over another entity:
  - CLS of Role Dominate ($\geq$) CLS of Resource, Service, or Method
  - CLR of User Dominate ($\geq$) CLS of Role

- **Definition 14:** A *user-role authorization*, URA, signifies a UR authorized to invoke a method under optional TC and/or SC, and is defined as:

  \[
  URA = [UR, M, TC, SC]
  \]

- **Definition 15a:** *UR authorization matrix*, URAM:

  \[
  URAM (UR_i, M_j) = \begin{cases} 
  1 & \text{if } UR_i \text{ is authorized to invoke } M_j \\
  0 & \text{otherwise}
  \end{cases}
  \]
Example Users, User Roles, and URAs

**Users:**

\[ U = [U^{UserId}, U^{LT}, U^{CLR}] \]

(T) General DoBest: [DoBest, [ct, ∞], T]
(T) Colonel DoGood: [DoGood, [01dec00, 01jun01], T]
(S) Major DoRight: [DoRight, [01dec00, 01jan01], S]
(T) Major CanDoRight: [CanDoRight, [01jan01, 01feb01], T]

**User-Roles:**

\[ UR = [UR^{Name}, UR^{LT}, UR^{CLS}] \]

[CDR_CR1, [01dec00, 01dec01], T]
[JPlannerCR1, [01dec00, 01jun01], S]
[JPlannerCR2, [01jul01, 01sep01], C]
[ArmyLogCR1, [10dec00, 01mar01], S]
[ArmyLogCR2, [01jul01, 01aug01], C]

**User Role Authorizations:**

\[ URA = [UR, M, TC, SC] \]

[JPlannerCR1, CrisisPicture, [ct, ∞], true]
[JPlannerCR1, ArmyBattleCommandSys, [10dec00, 16feb01], true]

[ArmyLogCR1, CrisisPicture, [10dec00, 16feb01],
Grid1 ≤ NA20 AND Grid2 ≤ NC40],
[ArmyLogCR1, LogPlanningTool, [10dec00, 16feb01], CrisisNum=CR1]
Security Model Definitions

Remaining Definitions

- **Definition 15b**: A valid user-role authorization list, \( VURAL \) has \( URAM(UR,M) = 1 \).

- **Definition 16**: A user authorization, \( UA \), is a user authorized to play a role: \( UA = [U, UR, TC] \).

- **Definition 17a**: User authorization matrix, \( UAM \): \[
UAM(UR_i, U_j) = \begin{cases} 
1 & \text{if } U_j \text{ is authorized to } UR_i \\
0 & \text{otherwise}
\end{cases}
\]

- **Definition 17b**: A valid user authorization list, \( VUA \) has \( UAM(UR,U) = 1 \).

- **Definition 18**: A client, \( C \), is authorized user \( U \), uniquely identified via a client token \( C = [U, UR, IP-Address, Client-Creation-Time] \).
Role Delegation Extensions

- **Role Delegation**: Where Authorized Individual (not the Security Officer) may delegate all or part of his/her Authority (roles) to Another Individual
  - Identify Roles that are Delegatable
  - Distinguish: Original and Delegated Users
  - Delegation Authority and Delegated Role

- **What Can be Delegated?**
  - Authority – can do task, but no responsibility
  - Responsibility – implies accountability
  - Duty – obligation to execute task
  - **Our Focus: Delegate Authority Only**
Role Delegation
Delegatable UR, Original/Delegated User

- **Definition 19**: A *delegatable UR, DUR, is a UR that is eligible for delegation.*

- **Definition 20**: The *delegatable UR vector, DURV, is defined for all r as*:
  \[
  DURV(UR_i) = \begin{cases} 
  1 & \text{if } UR_i \text{ is a DUR} \\
  0 & \text{if } UR_i \text{ is not a DUR} 
  \end{cases}
  \]

- **Definition 21**: An *original user, OU \in UL, is authorized to the UR such that there exists a VUA for the OU/UR, \text{i.e.}, \text{UAM}(UR,OU) = 1.*

- **Definition 22**: A *delegated user, DU \in UL, is a user eligible to be delegated a UR by an OU or a DU (there is not a VUA \text{i.e.}, \text{UAM}(UR,DU) \not= 1). Note a DU of a UR cannot be an OU for same UR.*
Role Delegation Model Extensions

Definition 23: User delegation/authorization matrix, UDAM:

$$UDAM ( UR_i, U_j ) = \begin{cases} 
2 & U_j \text{ is a DU of } UR_i \\
1 & U_j \text{ is an OU of } UR_i \\
0 & U_j \text{ is not authorized to } UR_i 
\end{cases}$$

Definition 24: Delegation authority, DA, is given to the OU to allow delegation of a DUR.

Definition 25: Pass-on delegation authority, PODA, allows an OU (DU) to pass on DA for a DUR to another user (OU or DU).

Definition 26: Delegation authority matrix, DAM:

$$DAM ( UR_i, U_j ) = \begin{cases} 
2 & U_j \text{ has DA and PODA for } UR_i \\
1 & U_j \text{ has only DA for } UR_i \\
0 & U_j \text{ has neither DA nor PODA for } UR_i 
\end{cases}$$
Example - Role Delegation

- General DoBest Delegates his Role to Colonel DoGood with DA, where DoBest, CDR_CR1, and DoGood defined as:

  OU: [DoBest, [ct, ∞], T]
  UR: [CDR_CR1, [01dec00, 01dec01], T]
  UA: [DoBest, CDR_CR1, [01dec00, 01dec01]]
  DA: Yes
  PODA: Yes

- After Delegation:

  DU: [DoGood, [01dec00, 01jun01], T]
  UA: [DoGood, CDR_CR1, [01dec00, 01jun01]]
Example - Role Delegation

- **Now**, Colonel DoGood wishes to re-delegate CDR_CR1 to Major CanDoRight, which can be defined as:

  DU: [DoGood, [01dec00, 01jun01], T]
  UR: [CDR_CR1, [01dec00, 01dec01], T]
  UA: [DoGood, CDR_CR1, [01dec00, 01jun01]]
  DA: Yes
  PODA: No

- **After Delegation**:

  DU: [CanDoRight, [01jan01, 01feb01], T]
  UA: [CanDoRight, CDR_CR1, [01dec00, 01jun01]]
### Example Matrices Used to Define/Track/Monitor Delegation

#### User Authorization Matrix (UAM): 1 = authorized, 0 = other

<table>
<thead>
<tr>
<th>User\User-Role</th>
<th>ArmyLogCR1</th>
<th>ArmyLogCR2</th>
<th>JPlannerCR1</th>
<th>JPlannerCR2</th>
<th>CDR_CR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoBest</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DoGood</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>DoRight</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CanDoRight</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Delegation Authority Matrix (DAM): 2 = has DA and PODA, 1 = has DA, 0 = neither

<table>
<thead>
<tr>
<th>User\User-Role</th>
<th>ArmyLogCR1</th>
<th>ArmyLogCR2</th>
<th>JPlannerCR1</th>
<th>JPlannerCR2</th>
<th>CDR_CR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoBest</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>DoGood</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>DoRight</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CanDoRight</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### User Delegation/Authorization Matrix (UDAM): 2 = U is a DU, 1 = U is a OU, and 0 = not authorized

<table>
<thead>
<tr>
<th>User\User-Role</th>
<th>ArmyLogCR1</th>
<th>ArmyLogCR2</th>
<th>JPlannerCR1</th>
<th>JPlannerCR2</th>
<th>CDR_CR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoBest</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>DoGood</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>DoRight</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CanDoRight</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### User-Role Authorization Matrix (URAM): 1 = UR authorized to invoke Method, 0 = otherwise

<table>
<thead>
<tr>
<th>Method\User-Role</th>
<th>ArmyLogCR1</th>
<th>ArmyLogCR2</th>
<th>JPlannerCR1</th>
<th>JPlannerCR2</th>
<th>CDR_CR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArmyBattleCommandSys</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CrisisPicture</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MarineCombatOpnsSys</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LogPlanningTool</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Related Work: Role Delegation

- Role Administration [Awis97]
- Delegation with RBAC [Bark00, Na00]
- Delegation Principals [Zhang01]
- Similarities and Differences
  - In Our Approach, OU Maintains Control of Delegation
    - DU Cannot Give Delegation Authority
  - Our Approach is Dynamic, in that, Delegations have LTs Changeable During Runtime
  - Our Delegation Incorporates MACC
  - We extend Zhang’s Definitions to Include
    - Delegation Authority, Revocation Authority, Delegated Role, and Delegatable Role
Design Time Security Assurance for Delegation

- Design Time Checks – Policy Realization
  - MACC Domination
    - Clearance (CLR) Dominates Classification (CLS)
  - Role Delegation
    - DU Not Already a Role Member
  - User to User Delegation Authority
    - Must Check User Delegation Authority Matrix
    - DU Meets MACC Requirements
  - Lifetime Consistency
    - DU’s LT Must be Within OU’s LT
  - Modified Boolean Delegation
    - OU can Delegate and Pass On Delegation Authority
    - DU cannot Pass On Delegation Authority
Run Time Security Assurance for Delegation

- Executed While Running Distributed Application
  - MACC Domination
  - Role Delegation
  - User to User Delegation Authority
  - Lifetime Consistency
  - Modified Boolean Delegation (additional checks)
  - Delegation Revocation Authorization Rule
    - OU/DU Can Revoke Any Initiated Delegation
  - Cascading Revocation Rule
    - Whenever OU is Revoked, OU’s Delegations are revoked, Including Passed On Delegations
Security Enforcement Framework

- **Unified Security Resource (USR) Services to:**
  - Manage URs and Privileges
  - Authorize URs to Us
  - Identify Users and Track Security Behavior
  - USR Performs Run Time Assurance

- **Security Tools Performs Design Time Assurance**
  - *Security Policy Client* to Grant/Revoke Privileges (TCs, methods, SCs)/set CLS/CLR
  - *Security Authorization Client* to Assign CLR and Authorize URs to End Users
  - *Security Delegation Client (SDC)* for Defining and Managing Role Delegation
  - *Security Analysis Tool (SAT)* to Track all Client Activity (Logons/Method Invocations)
Security Enforcement Framework
Software Architecture

- Database Client
- COTS Client
- Lookup Service
- Wrapped Resource for COTS Application
- Wrapped Resource for Database Application
- General Resource
- Wrapped Resource for Legacy Application
- Lookup Service
- Java Client
- Software Agent
- Legacy Client
- Security Policy Client (SPC)
- Security Authorization Client (SAC)
- Security Delegation Client (SDC)
- Global Clock Resource (GCR)

Unified Security Resource (USR)
- Security Policy Services
- Security Authorization Services
- Security Registration Services
- Security Analysis and Tracking (SAT)

Oracle DB
Security Policy Client
Creating a UR

Security Policy Client
File Help

Role Resource Checking Tool

Create/Erase Role Update Role Grant Resource Grant Service Grant Method Grant IP Query

Role: CDR-CR1
Description: CDR-CR1
Classification: T

Delegation Authority

Create Erase Clear

Success
role 'CDR-CR1' created

OK
Security Authorization Client
Creating a User

User Profile
- Country: None
- Sex: None
- Age: None
- Service: None

User ID: General DoBest
Password: ******
Description: General DoBest
Clearance: T

Delegation Authority
- Start: 7 (Month), Date: 22, Year: 2002
- Time: 11 (Hour), Min: 33, Sec: 10 (PM)
- End: 7 (Month), Date: 22, Year: 2003
- Time: 11 (Hour), Min: 33, Sec: 11 (PM)

Create
Clear

Success
user 'General DoBest' is created
OK
Security Authorization Client
Granting UR to User
Security Delegation Client
Granting Delegation

Granting Delegation

Delegate to:
Colonel DoGood
Colonel DoGood

Delegatable Role list
Major DoRight
Major CanDoRight

USER ID: General DoBest
PASSWORD: ******
Get Your Role(s): CDR_CR1

Start
Month: 7
Date: 22
Year: 2002

Time
Hour: 11
Min: 41
Sec: 3
PM

End
Month: 7
Date: 22
Year: 2003

Time
Hour: 11
Min: 41
Sec: 3
PM

Pass-on-delegation Authority

Grant
Security Delegation Client
Granting Delegation

Security Delegation Client

Delegate to:
Colonel DoGood
Colonel DoGood

Delegatable Role list
Major DoRight
Major CanDoRight

Start: Month 7, Date 22, Year 2002
Time: Hour 11, Min 41, Sec 3

End: Month 7, Date 22, Year 2003
Time: Hour 11, Min 41, Sec 3

Pass-on-delegation Authority

Grant

Mac constraint violated

OK
Security Delegation Client
Revoking Delegation
Conclusions and Ongoing/Future Work

Conclusions

- Presented Underlying Security Model
- Detailed Role Delegation Extensions to Security Model
- Introduced Assurance Checks for Delegation
- Presented Screen Shots of Prototype

Ongoing/Future Work

- User Role Deconfliction
- Internal Role Deconfliction
- Increased Method Level MAC Constraints
- Establish Group Roles

See: http://www.engr.uconn.edu/~steve/DSEC/dsec.html