Integrated Security Services for Dynamic Coalition Management

Virgil D. Gligor and John S. Baras

Electrical and Computer Engineering Department,
University of Maryland
College Park, Maryland 20742

DARPA DC PI Meeting
January 22, 2002
San Diego, California

The views and conclusions presented herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA or AFRL
Project Title: Integrated Security Services for Dynamic Coalition Management
Project Start Date: March 15, 2000
Project Duration: 36 months
Options: None
Contract No: F30602-00-2-0510
Agent POC: Brian T. Spink, AFRL/IFGB, Rome Research Site
Personnel:
  Virgil D. Gligor (PI) ECE UMCP, gligor@eng.umd.edu
  John S. Baras (Co-PI) ECE/CS/ISR UMCP, baras@isr.umd.edu
  Faculty Research Assistant: Serban Gavrila
  Graduate students: Vijay Bharadwaj, Himanshu Khurana
  Radostina Koleva, Emilian Dinu, Rakesh Bobba, Vicky Zhao
Integrated Security Services for Dynamic Coalition Management

Outline

• Coalition-Resource Management Tools: Architecture, Implementation

• Joint Administration of Access Policies

• Progress of Project Tasks

• Future Work
Research Area: Dynamic Coalitions

♦ Diverse membership
  – Overlapping but not identical member interests
  – Multiple, autonomous domains of security policy administration
  – Potentially different access control, authentication, and secure communication technologies

♦ Dynamically changing membership
  – Member join / departure at any time after coalition creation/setup
  – Membership duration can be lower than current time to set up a coalition (e.g., weeks, months)

♦ Dynamic management of member access to resources
  – Member join / departure must not require new coalition creation/setup
  – Access control negotiations can take place at any time
Integration Goal: Develop Coalition Resource Management (CRM) Tools
Example: CRM Components for Two Domains

* Policy specification/negotiation language and visual tools are included
Stage II: Architecture
Stage II: Flows
(Integration of RBAC and PKI)

Central
Attr. Authority

Access CA

Negotiation Server

RBAC

Domain 1

Negotiation Client

RBAC

User

Domain 2

Negotiation Client

RBAC

App Server

(1) Negotiation

(3) Attribute Certificate

(for foreign users)

(2) Role Membership

(4) Access Request with Attribute Cert.
Stage II: Flows
(Integration with Group Communication)

Central AA
- Access_CA
- Neg. Server
- RBAC
  - Group Key Mgmt.
  - Chat Server

Domain 1
- Negotiation Client
- RBAC
- Key Generation
- SPREAD
- User Bob

Domain 2
- Negotiation Client
- RBAC
- Key Generation
- SPREAD
- User Anne

(5) Attribute Certificate
(6) Group Key Gen.
(7) Secure Multicast Communication
Stage II: Enforcing Dependencies in Integration

- **Enforce dependency between role membership and attribute certificate**
  - Automatic *distribution* of attribute certificate for user $\Rightarrow$ role membership
  - Automatic *revocation* of attribute certificate for user $\Rightarrow$ role membership

- **Enforce dependency between role membership and group communication**
  - User *joins* secure-communication group using role-membership certificate
  - Automatic *group re-keying* on user $\Rightarrow$ role membership revocation (and on attribute certificate revocation)
Stage II: Implementation

- **Operating System**: Windows 2000 Server
- **Access Control Policy**: RBAC Tool (Role Control Center-RCC)
  - Developed by VDG Inc with partial NIST support
- **Negotiation Client/Server**: Active Server Pages
- **Certificate Authority**: Windows 2000 Server Stand-Alone CA
  - CA policy modules with interface to RCC in Visual Basic
- **Application Server**: Windows IIS Web Server
  - Access to website using certificates over SSL
  - Resources of common state: Web pages
- **Group Communication Server**: Java
  - Integrated with Spread Toolkit for multicast (*developed at Johns Hopkins Univ.*)
- Negotiation and commitment of common state requires *no modification* to OS, network protocols and to PKI enabled application servers
1. Multicast Key Management

J.S. Baras and H. Zhao “Design of Hierarchical Key Management Schemes for Group Communications in Mobile Wireless Networks” submitted for publication

2. Enforcing Dependencies among PKI certificates


3. Negotiations of Access Control Policies


4. Joint Administration of Access Control Policies for Coalition Resources

January 22, 2002
Joint Administration of Access Policies for Coalition Resources

A Coalition Example

- Pharmaceutical Company
- Genetics Firm (Discovers Gene Sequence)
- Private Hospital

Joint Administration of Access Policies

Research Data for Gene Sequence
1 Central attribute authority (AA)
   – Supports access to coalition resources in the presence of coalition dynamics
   – Supports separate administration of local and coalition resources

2 Coalition-closed administration of shared access policies
   – Outsourcing policy administration => trust in outside domain
   – Outside domain would become single point of trust failure

3 Consensus for administration of access policies
   – A single coalition domain should be unable to unilaterally define and modify access policies of a jointly owned resource
     • At least, no single domain should be able to do this in a repudiable manner
Satisfying Requirements 1-2

Joint Creation: Central Attribute Authority (AA)

Threshold Attribute Certificate

Coalition Web Server

CA1

User_D1

Domain 1

CA2

User_D2

Domain 2

CA3

User_D3

Domain 3

ID Cert.

ID Cert.

ID Cert.

Joint Access Request

ACL write

ACL

Coalition Object O
Satisfying Requirement 3 (Consensus)

- **Problem:** Who creates and protects Central AA’s private key?
  - If domain $D_i$, then *it* can choose to issue new attribute certificates unilaterally
  - If *all* domains, then *any* domain can issue new attribute certificates unilaterally

- **Solution (inadequate): indirect creation and protection**
  - Central AA’s private key is created and stored *inside* a “lock box” (e.g., IBM 4758)
  - *Trust liabilities:*
    - External penetration of “lock box” $\Rightarrow$ Central AA is a *single point of trust failure*
      - Formal verification of *crypto transaction sets* accessing “lock box” is still a challenge
    - Internal (insiders*) penetration of “lock box” $\Rightarrow$ *consensus violation*

*viz., Richard Clayton and Mike Boyd’s example of IBM 4758*
• **Solution:** use *shared public-key cryptosystems*
  – All domains create Central AA’s public-key while retaining shares of the private key
    • Signatures with shared key requires application of joint signature algorithm
  – Trust liabilities minimized: *all* domains have to be compromised to obtain private key

• **Theory used**
  – Logic extensions to capture trust relations of shared public-keys
Evaluation and Design of Key Management Schemes

Simple model example: 1 GSC, $n_1$ GSAs, each cluster $n_2$ members

$K_{ij}^1$ GSA to member
$K_{i}^2$ GSA to cluster
$K^3$ GSC to members

Assumptions
- Broadcast available at both layers
- Timeline: only one motion event at an event instant
- “member moves”, “GSA moves”, different frequencies
- When GSA leaves, new one selected and cluster reconstituted
- Ignore details of handoff & reallocation to clusters

PARAMETERS:
- Key tree parameters ($d$, $h$), frequencies of member motion ($p$), length of keys ($K$), comp. cost for generating a key ($C_r$), comp. cost per PKI encryption/decryption ($C_{PE}/C_{PP}$), comp. cost per symmetric key encryption/decryption ($C_{SE}/C_{SD}$)

Metrics compared: ($d$, $p_2/p_1$, $n_2/n_1$

Design sizing (init), Operational Storage (GSC, GSA, member)
Computations (GSC, GSA, member)
Communications (key updates)
Developing theory and software tool for evaluation / design

January 22, 2002
Comparisons

- **Metrics**: storage, computation, communication costs
- **First Scheme/Architecture**: Single Key Tree, Two Groups of Members
- **Second Scheme/Architecture**: GSC to GSAs Key Tree, Each Cluster GKMP
- **Third Scheme/Architecture**: GSC to GSAs Key Tree, Each Cluster Key Tree
- **Fourth Scheme/Architecture**: Single Key Tree, Single Group of Members
Results

- Robustness of comparisons for a variety of values
- \( n_2/n_1 = l \ (1, 4, 16), \quad p_2/p_1 = r \ (5, 10, 15, 20, 25) \)
- Metrics much more sensitive to \( l \), values for \( K, Cs, \)
- Metrics insensitive to \( r \)
- Average (operating communication costs)
  - First scheme: 15.5 \( K \)
  - Second scheme: \( (16\sqrt{l + 1}) K(17, 33, 65) K \)
  - Third Scheme: \( (8 + (3/4)\log_2 l) K \ (8, 9.5, 11) K \)
  - Fourth Scheme: 15.5 \( K \)
- Developed general multi-criteria trade-off design and evaluation methods and algorithms, including models of costs and dynamics of mobile networks with hierarchies
• Used general and realistic dependencies of costs (metrics) on network parameters
• Used general hierarchies
• Results in: John S. Baras and Hong Zhao “Design of Hierarchical Key Management Schemes for Group Communications in Mobile Wireless Networks” *submitted for publication*

• **Efficient Distributed Key Generation for Secure Group Communications**

  • Designed new Group Key Authentication for distributed key generation scheme
  • Distributed authentication schemes for Group Communications
  • Integration of distributed authentication schemes and distributed key generation schemes
  • Implementation in wireless MANETs
  • Results in: John S. Baras, Arvind Mani and Vijay Bharadwaj “Distributed Key Generation, Authentication and Management for Group Communications in MANETs” *in preparation for submission*
Cooperative Games and Dynamic Coalitions

- Have a number of players, some can be coalitions themselves
- How do they negotiate an “acceptable” DC security policies set?
- What are the properties of the final result: “the negotiated policy set”?
- Is there an efficient scheme that gets us there?
- Can we handle different notions of “reasonable” “fair”?
- Cooperative games allow us to set up different types of games between the players, and to examine different concepts of solutions and values
- Can prove mathematically properties of the solution and value: e.g. minimizes maximum dissatisfaction, is anonymous, is stable
- Can get iterative methods to get to solution (negotiation schema), can use all kinds of constraints, invariance to aV + b scaling (preferences)
- Working on extensions to partial information and learning, robustness to uncertainties
Cooperative Games

- **Cooperative Game** in characteristic function form
  \[ \Gamma = \{N, v\}, \; N = \{1, 2, \ldots, n\}, \; v : 2^N \rightarrow R \] defined on all subsets (coalitions of \( N \))

- A coalition, \( v(S) \) is “interpreted” as the maximum utility \( S \) can get without the cooperation of players in \( N \setminus S \)

- \( \Gamma \) **superadditive**: \( S, T \subseteq N, S \cap T = \emptyset, v(S \cup T) \geq v(S) + v(T) \)

- \( \Gamma \) **monotone**: \( S \supset T \) implies \( v(S) \geq v(T) \)

- \( \Gamma \) **convex**: for each \( i \in N, S \subset T \), implies \( d_i(S) \leq d_i(T) \)

  \[ d_i(S) = \begin{cases} 
  v(S \cup \{i\}) - v(S), & \text{if } i \notin S \\
  v(S) - v(S \setminus \{i\}), & \text{if } i \in S 
  \end{cases} \]

  increasing marginal contribution of \( i \)

- \( \Gamma \) **rational**: \( v(N) \geq \sum_i v(\{i\}) \)

- \( \Gamma, \Gamma' \) **S-equivalent**: \( v(S) = \alpha v'(S) + \sum_{i \in S} \beta_i \)
Cooperative Games and Payoffs

- **Feasible payoff** vectors \( I^{**}(N, v) = \{ x \mid x \in \mathbb{R}^n, x(N) \leq v(N) \} \)
- **Efficient payoff** vectors \( I^*(N, v) = \{ x \mid x \in \mathbb{R}^n, x(N) = v(N) \} \)
- **Individually rational payoff** vectors
  \( I(N, v) = \{ x \mid x \in \mathbb{R}^n, x(N) = v(N), x_i \geq v(\{i\}) \text{ all } i \in N \} \)
- For a *set of games*, a *solution* \( \sigma \) associates with each game \( \Gamma \) a subset of \( I^*(N, v) \). Can be characterized either by math relations or axioms. Helps capture different notions of “desirable” or “reasonable” properties of solutions
- **\( x \) dominates \( y \) through coalition \( S \)** (\( x \geq^S y \)) if \( x_i > y_i, i \in S, x(S) \geq v(S) \)
- **\( x \) dominates \( y \)** (\( x \geq y \)) if \( x \geq^S y \) for some coalition \( S \)
Cooperative Games: Solution Concepts

- **Core** (stable, reasonable payoffs): gives each coalition at least as much as could get by itself
  \[ \{ \mathbf{x} \in \mathbb{R}^n ; x(S) = \sum_{i \in S} x_i \geq v(S) \text{ all } S \subset N, x(N) = v(N) \} \]
  - Convex and average convex games have nonempty cores
  - For a set of games the core is the unique solution that is individually rational, superadditive, nonempty and satisfies the reduced game property

- **Stable sets**: \( V \subset I \), there is no \( \mathbf{x}, \mathbf{y} \in V \) s.t. \( \mathbf{x} \geq \mathbf{y} \), and if \( \mathbf{y} \notin V \), there is \( \mathbf{x} \in V \) s.t. \( \mathbf{x} \geq \mathbf{y} \)

- **Nucleolus**: excess \( e(S, x) = v(S) - x(S) \), measure of dissatisfaction of coalition \( S \) for payoff \( x \). Set \( \theta(x) = (e(S, x))_{S \subset N} \); solution obtained by \( \min \{ \psi(\theta(x)) \mid x \in I(N,v) \} \). Minimize maximal complaint. Least squares nucleolus (excesses are close to average)

- **The Nucleolus** is always in the core

January 22, 2002
Cooperative Games: Solution Concepts

- **Nucleolus** is the individually rational payoff that lexicographically minimizes the excess vector
  - Leads to iterative procedure for getting there
  - Use a small set of linear programs that iteratively minimize the highest excess, then the second highest excess, etc.
  - A solution concept is the **Nucleolus** if and only if it is anonymous (ind. of payer labeling), covariant (ind. of scale expressing preferences), satisfies the reduced game property

- **Shapley Value**: solution $\phi$ with components the *expected marginal contribution* made by $i$ when entering coalition $N$
  - $T$ is a carrier if $v(S) = v(S \cap T)$, $v(S) = \sum_{i \in S} \phi_i(v)$. The **Shapley Value** is the unique solution that has this property, is anonymous and is additive
  - For convex games **Shapley Value** is in the core

- **Kernel, Bargaining Set**: consider coalition structures, their stability, objections and counterobjections
• Two or more autonomous policy domains form a coalition to share data
• Domains must negotiate a common access control policy
• Games $\rightarrow$ automate negotiation process (by software agents residing on domain controllers)
• Negotiation process: each domain has a central controller to administer its local policies and negotiate with other domains
• Outcome of negotiation is determined by the distribution of power among participants
• Games; players are rational and intelligent; solution concept is a rule that associates a game with a set of outcomes
• Unanimous agreement is required to stop negotiation
• For simplicity assume absence of exogenous rewards or costs
Negotiation Example: Bandwidth Sharing

Two cities: X, Y
3 ISP: D1, D2, D3
Need to pool resources
• No-one from X to Y
• D1 is needed
• D1 & D2 can get 1Mbps
• D1 & D3 can get 0.5 Mbps
• All can get 3.5 Mbps
• B₁, B₂, B₃ share of
  • aggregate data capacity
  • assigned to D1, D2, D3
Negotiation Example: Intelligence Sharing

- $n_i$ No of sources owned by agency $i$
- $s_i$ No of sources $i$ shares with others

**Agency 1**
- 7 sources
- Most secure ($P_{\text{leak}} = 0.1$)

**Agency 2**
- 4 sources
- Least security ($P_{\text{leak}} = 0.5$)

**Agency 3**
- 10 sources
- Medium secure ($P_{\text{leak}} = 0.2$)
Negotiation Example: Intelligence Sharing

- Three domains (intelligence agencies)
  - Need to share output from some or all of their sources
  - Sharing increases risk of exposure to undesirable parties
  - Leak of intelligence compromises source

- Game theoretic solutions
  - Nash bargaining solution
    - Pareto optimal (no one can do better without harming someone else)
  - Nucleolus
    - Fair (evenly distribute benefits of cooperation)
  - Shapley value
    - Measure of “bargaining power” – depends on each domain’s contribution to the coalition
Nash Bargaining Solution

- Individual rationality
  - Bandwidth: \( B_1 \geq 0, B_2 \geq 0, B_3 \geq 0 \)
  - Intelligence: \( \frac{s_1}{s_1 + s_2 + s_3} \leq 0.4, \frac{s_2}{s_1 + s_2 + s_3} \leq 0.72, \frac{s_3}{s_1 + s_2 + s_3} \leq 0.45 \)
  - A2 needs to share more, A1 least interested

- Nash showed any game has a solution s.t.

  *individual rationality, symmetry, scale covariance, Pareto optimality, independence from unfavorable alternatives*

- Individually rational solution which maximizes the product of the gains made by each player due to coalition formation
  - Bandwidth: \( B_1 = B_2 = B_3 = \frac{3.5}{3} \) Mbps
  - Intelligence: \( s_1 = 4, \ s_2 = 4, \ s_3 = 5 \)
Core and Nucleolus

- **Core**: set of solutions that give to every possible coalition at least as much payoff as that coalition could get without the support of the other players (stability)
  - Bandwidth: maximum capacity can get without help from non-coalition members
  - Intelligence: sum of the values of Nash bargaining solution for coalition
  - Core for intelligence sharing is empty

- **Nucleolus**: solution that distributes payoffs so as to lexicographically minimize the dissatisfaction of all coalitions in the grand coalition
  Or it maximizes the benefit of least rewarded member of coalition (fairness)

- **Nucleolus is the unique solution** s.t.: anonymity, scale covariance, imputation saving reduced game property
  - Bandwidth: \( B_1 = B_2 = B_3 = \frac{3.5}{3} \text{Mbps} \)
  - Intelligence: \( s_1 = 2, \quad s_2 = 4, \quad s_3 = 0 \)
Shapley Value

- **Shapley value**: assigns payoffs to players depending on their average marginal contribution to grand coalition.
- It favors players who play a larger role in the success of grand coalition over smaller players.
- **Shapley value is the unique solution** s.t.: anonymity, carrier (dummy player), linearity
  - Bandwidth: $B_1 = \frac{17}{12} \text{ Mbps}$, $B_2 = \frac{14}{12} \text{ Mbps}$, $B_3 = \frac{11}{12} \text{ Mbps}$
  - Intelligence: $V_1 = 9.91$, $V_2 = 0.55$, $V_3 = 10.84$
- Results in: Vijay Bharadwaj and John Baras “Negotiating Access Control Policies Between Autonomous Domains” submitted for publication.
# Negotiation Example: Intelligence Sharing

<table>
<thead>
<tr>
<th>Domain</th>
<th>No. of sources</th>
<th>Prob. of leak</th>
<th>Nash solution</th>
<th>Nucleolus</th>
<th>Shapley value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>7</td>
<td>0.1</td>
<td>4</td>
<td>2</td>
<td>9.91</td>
</tr>
<tr>
<td>A2</td>
<td>4</td>
<td>0.5</td>
<td>4</td>
<td>4</td>
<td>0.55</td>
</tr>
<tr>
<td>A3</td>
<td>10</td>
<td>0.2</td>
<td>5</td>
<td>0</td>
<td>10.84</td>
</tr>
</tbody>
</table>
Ongoing and Future Work

- A state-transition model for negotiating access to coalition resources
  - commitment of negotiated common state requires satisfaction of negotiation constraints, policy administrability and compatibility requirements

- Enhance policy negotiation language and visual tool

- Develop a service for joint administration of coalition resources

- Introduce identity certificates and enforce dependencies between attribute and identity certificates

- Add integrity for multicast messages in group communication
Ongoing and Future Work

- Complete game analysis of solutions with extended finite state machine models for RBAC
- Efficient algorithms for fast solutions
- Accommodation of time varying RBAC schemes
- Problems in DC where negotiators know resources and reward functions of others only partially
### Progress on Project Tasks

#### Tasks

<table>
<thead>
<tr>
<th>Task Description</th>
<th>FY 00</th>
<th>FY 01</th>
<th>FY 02</th>
<th>FY 03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of Policy Integration</td>
<td>100%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impl. &amp; Test of Policy Server</td>
<td>84%</td>
<td>84%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design, Impl., Test of VPR Tool</td>
<td>53%</td>
<td>53%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration and Demonstration</td>
<td>38%</td>
<td>38%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis of Key Mgmt. Properties</td>
<td>89%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design of Key Mgmt. Server</td>
<td>75%</td>
<td>90%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impl., Test, of Key Mgmt. Server</td>
<td>31%</td>
<td>31%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration and Demonstration</td>
<td>23%</td>
<td>1%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis of Cert. Mgmt. Properties</td>
<td></td>
<td>TASK NOT FUNDED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design of Admin. Server</td>
<td>73%</td>
<td>85%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Impl., Test, of Admin. Server</td>
<td>33%</td>
<td>33%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration and Demonstration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Actual Planned**

- 1 2 3 4 5 6 7 8 9 13 15 17 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

**Progress**

- **Completed**
- **In Progress**
- **Not Started**
- **Adjusted**