Integrated Security Services for Dynamic Coalition Management

Himanshu Khurana and Vijay Bharadwaj

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

DARPA DC PI Meeting
January 30, 2003
San Antonio, Texas

This presentation contains no copyrighted, proprietary or sensitive information and the United States Department of Defense may freely disseminate the information contained herein.
Integrated Security Services for Dynamic Coalition Management

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
Research Associate: Himanshu Khurana
Faculty Research Assistant: Serban Gavrila
Graduate students: Vijay Bharadwaj, Rakesh Bobba, Emilian Dinu, Radostina Koleva, Anuja Sonalker
Integrated Security Services for Dynamic Coalition Management

Recent Work

• Coalition-Resource Management Tools: Architecture, Implementation
  – Supporting Coalition Dynamics (Demo Yesterday)

• Software Agents for Decentralized Negotiation of Coalition Resources: Design, Analysis
Integrated Security Services

• Certificate Services
  – Distribution/revocation of Identity and Attribute certificates

• Resource Management Services
  – Common view of coalition operations (intuitive graphical interface)
  – Access policy administration of shared applications

• Joint Administration Services
  – Joint administration requirements: Joint coalition authority, coalition-closed administration, consensus for access policy administration
  – Threshold CA distributes/revokes jointly signed certificates

• Secure Group Communication Services
  – Reliable multicast communication
  – Group key management

* Our services are provided over the internet or over an existing coalition network
Integration Goal: Develop Coalition Resource Management (CRM) Tools
Example: CRM Components for Two Domains

[Diagram showing CRM components for two domains, including CA, User, Negotiator, RBAC Tool, Sec. Group Comm., Attribute Certs, and Shared Keys.]
Stage IV: Components

- **Operating System**: Windows 2000 Server
- **Access Control Policy**: RBAC Tool (Role Control Center-RCC) - Java
  - Initially developed by VDG Inc with partial NIST support
- **Certificate Authority**:
  - Windows 2000 Server Stand-Alone CA
  - Shared Access CA (Yalta – NC State, MCNC) - Java
- **Group Communication**
  - Key Management Tool – Java
  - Server: Windows IIS Web Server – ASP
  - Multicast Communication Toolkit: SPREAD (JHU)
- **Application Server**: Windows IIS Web Server
  - Access to website using certificates (X.509 v3) over SSL
Efficient Support for Coalition Dynamics

- **Domain Leave Event:** Departing domain
  - Withdraws private resources (NOT jointly administered resources)
  - Excluded from private resource sharing
    - Large scale revocation of attribute certificates
    - Modification of CA trust relations
  - Excluded from joint administration
    - Re-key shared CA public-key or modify sharing threshold (if any)
    - Distribution of attribute certificates

- **Domain Join Event:** Joining domain
  - Shares private resources
    - Large scale distribution of attribute certificates
    - Modification of CA trust relations
  - Included in joint administration
    - Re-key shared CA public-key
    - Distribution of attribute certificates
Stage IV Flows: Domain Departure

1. Withdrowls private resources
2. Excluded from private resources
3. CA trust relations modified
4. Excluded from joint administration

Domain 1: CA trust relations modified
User --> CA
Public-Key Acc CA
Attr Certs

Domain 2: Excluded from private resources
User --> CA
Public-Key Acc CA
Attr Certs

Domain 3: Re-key Shared CA Cert
User
CA
Id Cert.

Joint AA
Shared Access CA
RBAC Tool (RCC)
Automated Negotiation of Common Access State

- Initially, each domain’s administrator specifies coalition requirements and any additional objectives
- Domains take turns proposing coalition states
  - Negotiation ends when one state is accepted by all domains
  - Domain administrators can add constraints to their agents at any stage
Software Agents for Resource Negotiation

• Negotiating agent must find most favorable state that satisfies all constraints

• Sources of constraints
  – Hard (Boolean) constraints due to domain policy
  – Soft constraints (preferences) due to domain objectives
  – Additional (hard and soft) constraints due to requirements of coalition mission
  – Soft constraints determine what states are considered more “favorable” than others

• Can discover some preferences just by looking at current state – objects restricted to higher roles are more valuable
Why Soft Constraints?

- No unique state satisfying all domain constraints
  - In most problems, there is either no common access state that satisfies the coalition mission and the hard constraints of all the domains, or there are many
  - An agent that randomly picks a state satisfying all the hard constraints will often end up with a state that fulfills the coalition mission but would be considered unreasonable by a human administrator (e.g. share as many resources as possible)

- Domains last much longer than individual dynamic coalitions
  - A domain will form many coalitions over its lifetime
  - Therefore human negotiators often try and obtain a common access state that not only fulfills the coalition mission but also confers some benefit for domain operations or coalition formation in the future (e.g. access to other domains’ privileged data)
Structure of Negotiating Agents

- Constraint compiler converts domain policies (written in some policy language) to negotiation constraints
- Constraint evaluator checks proposal against negotiation constraints
- Optimizer generates counter-proposal
- Administrator can guide agent by adding constraints at any time
Negotiating Agent: Constraint Compiler

- Need to represent hard and soft constraints in a common framework – use partial orders
  - In general, domain preferences form partial orders
  - In RBAC systems, roles and permissions also have hierarchies that are partial orders
• **Use semirings to represent partial orders**
  – $\langle A, +, \times, 0, 1 \rangle$ where $A$ is a set, $+$ and $\times$ closed and associative, $+$ commutative and distributes over $\times$
  – In our case we also have $+$ and $\times$ idempotent
  – $+$ defines a partial order over $A$
  – $+$ is used for comparison, $\times$ for composition

• **Semiring-based Constraint Logic Programming (SCLP)**
  – Constraint Logic Programming, with constraints over the semiring
  – $H:-B$ means $I(H\theta) \geq I(\exists B\theta)$, where comparison is over the semiring
  – blevel – maximum level of consistency of a constraint set
  – Expressiveness – can be implemented as a Prolog extension; e.g. can express all RCL2000 constraints
Semiring Based Constraints: An Example

• Can define semiring of roles:
  – A is the set of all roles
  – + and × are the glb and lub of two roles, respectively
  – The 0 and 1 elements are the highest and lowest roles respectively

• Example: A web server offers remote access to a number of services, by assigning remote users to local roles on the basis of supplied credentials. Each service requires a number of operations on various objects. Given that a user supplied a set of credentials C, is he allowed to run service S?
Semiring Based Constraints: An Example

• Solution:
  – Create *authorization SCSP*
    • Role assignment algorithm can be represented as an SCLP – for each set of credentials C0, we can assign a role R0.
    • The value of set C in this SCSP is the role assigned to the remote user. Call it R1.
  – Create *rights SCSP*
    • Access control policy for objects of interest to service S is also an SCSP – for each set of rights P, we can assign a value R which is the minimum role which has those rights.
    • The value of the rights required for S in this SCSP is the minimum role required to run S. Call this R2.
  – If R1 dominates R2, access is allowed

• Remarks:
  – This example uses SCSP for clarity. Direct translation of policy clauses to SCLP is easier and more efficient.
  – This is a simplistic example. The important point is that policies can be expressed as SCSPs (or equivalently SCLPs).
  – For more complex coalitions, the semiring becomes very large. Choosing an economical semiring is an open problem.
Negotiating Agent: Optimizer

- SCLP solving and finding blevel are NP complete

- Can use Linear Programming relaxations as an inference mechanism to reduce search space
  - Allows much more efficient solving by breaking problem into many smaller LPs and using these to carry out logic cuts
  - Can be used incrementally – can add constraints over time
Negotiating Agent: Design

- Constraint compiler compiles domain constraints to an SCLP
- Use hybrid solving techniques to solve the resulting SCLP efficiently
- The solution gives you a counter-proposal
- When should we stop?
  - Trade-off between long bargaining and information leakage
  - Need to detect impossible situations
Protocol Design

• **Tools from Game Theory**
  – **Optimal exchanges in exchange economies**
    • Problem is NP complete, but can be tractable if partial acceptance of bids is allowed
  – **Algorithms for minimal preference elicitation in exchange economies**
    • Used to design a parsimonious protocol
    • Can also allow us to quantify information leaks through negotiation process
Ongoing and Future Work

- Develop tools for negotiating access to coalition resources
  - include language for expressing wide variety of negotiation constraints
- Develop solutions for supporting involuntary domain departure
- Establish expressiveness and complexity of constraint representation
- Efficient algorithms to transform policies and objectives to constraint framework
- Analysis of negotiation protocols and their efficiency
- Technology demonstration at DISCEXIII
1. Multicast Key Management

2. Enforcing Dependencies among PKI certificates
H. Khurana and V. D. Gligor, “Review and revocation of Access Privileges Distributes with PKI Certificates,”

H. Khurana and V.D. Gligor, "Enforcement of Certificate Dependencies in ad-hoc Networks,“
Proc. IEEE International Conference on Telecommunications, Bucharest, Romania, June 2001,

3. Negotiations of Access Control Policies
(To appear in Lecture Notes in Computer Science, Springer-Verlag, 2002)


4. Joint Administration of Access Control Policies for Coalition Resources
H. Khurana, V. Gligor and J. Linn, “Reasoning about Joint Administration of Access Policies for Coalition Resources,”