Group-Centric Models for Secure and Agile Information Sharing

Ravi Sandhu
Executive Director and Endowed Professor
September 2010

ravi.sandhu@utsa.edu, www.profsandhu.com, www.ics.utsa.edu

Joint work with ICS colleagues
Ram Krishnan, Jianwei Niu and Will Winsborough

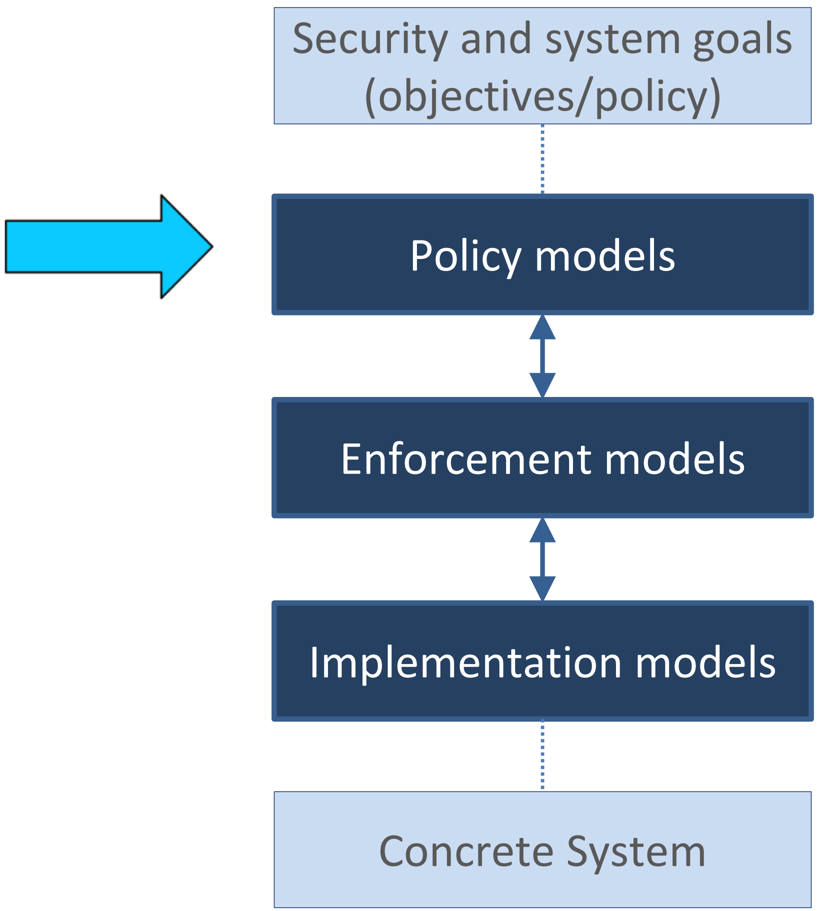
Goal: 4th Element

- 3 succesful access control models in 40+ years
  - Discretionary Access Control (DAC)
  - Mandatory Access Control (MAC)
    also called Lattice-Based Access Control (LBAC)
  - Role-base Access Control (RBAC)
- Numerous others defined and studied, implemented but no success
- Will Group Centric Models be the 4th element?
  - Strong mathematical foundations
  - Strong intuitive foundations
  - Significant real-world deployment

Goal: Share but protect

- Containment challenge
  - Client containment
    - High assurance infeasible (e.g., cannot close the analog hole)
    - Low to medium assurance achievable
  - Server containment
    - Will typically have higher assurance than client containment
- Policy challenge
  - How to construct meaningful, usable SIS policy
  - How to develop an intertwined information and security model

PEI Models



Necessarily informal

Specified using users, subjects, objects, admins, labels, roles, groups, etc. in an ideal setting.

Security analysis (objectives, properties, etc.).

Approximated policy realized using system architecture with trusted servers, protocols, etc.

Enforcement level security analysis (e.g. stale information due to network latency, protocol proofs, etc.).

Technologies such as Cloud Computing, Trusted Computing, etc.

Implementation level security analysis (e.g. vulnerability analysis, penetration testing, etc.)

Software and Hardware

Fundamental Goal: Share BUT Protect

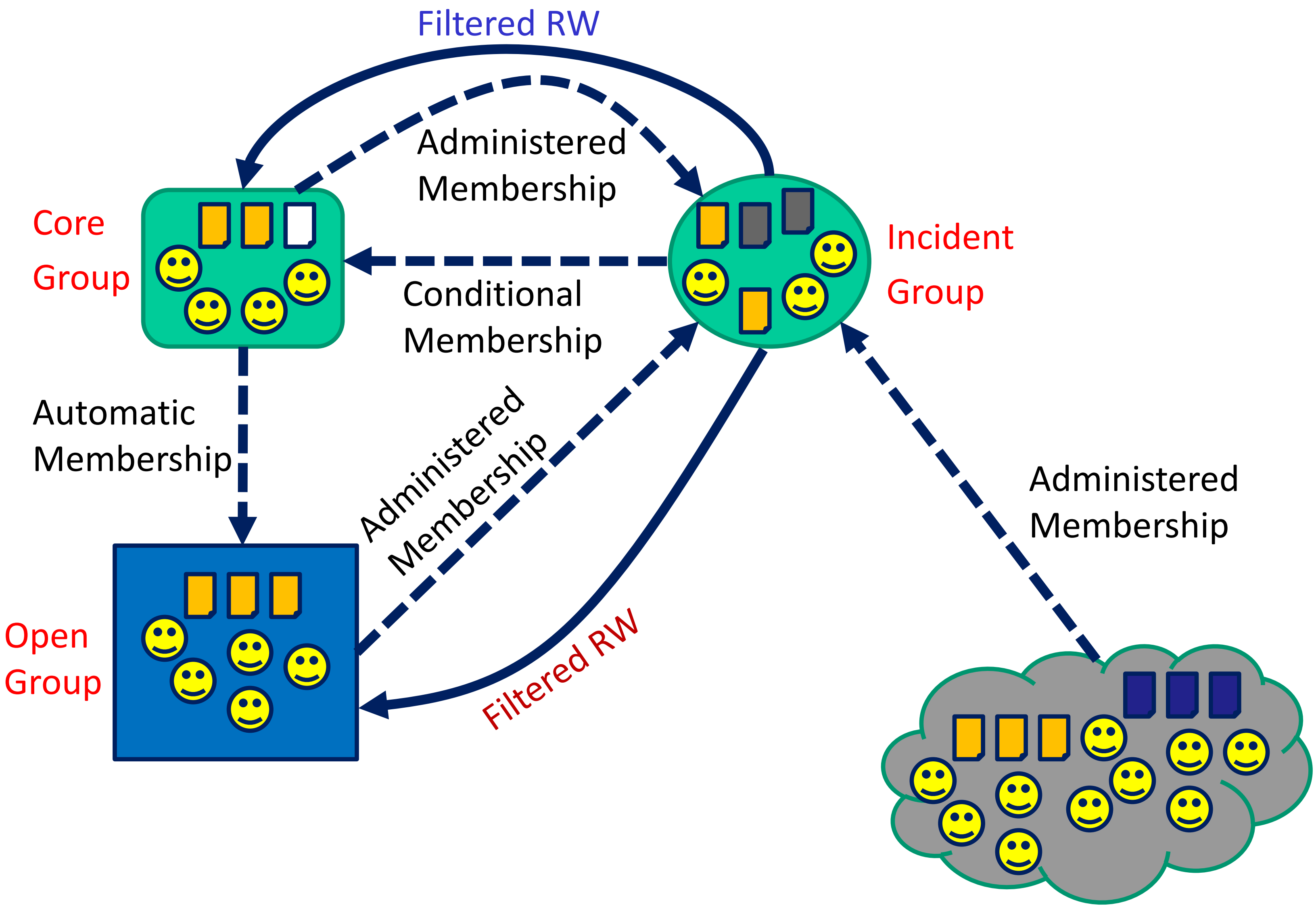
I. Dissemination-Centric Sharing
- Digital Rights Management
- Enterprise Rights Management
- XrML
- Workflow-centric sharing

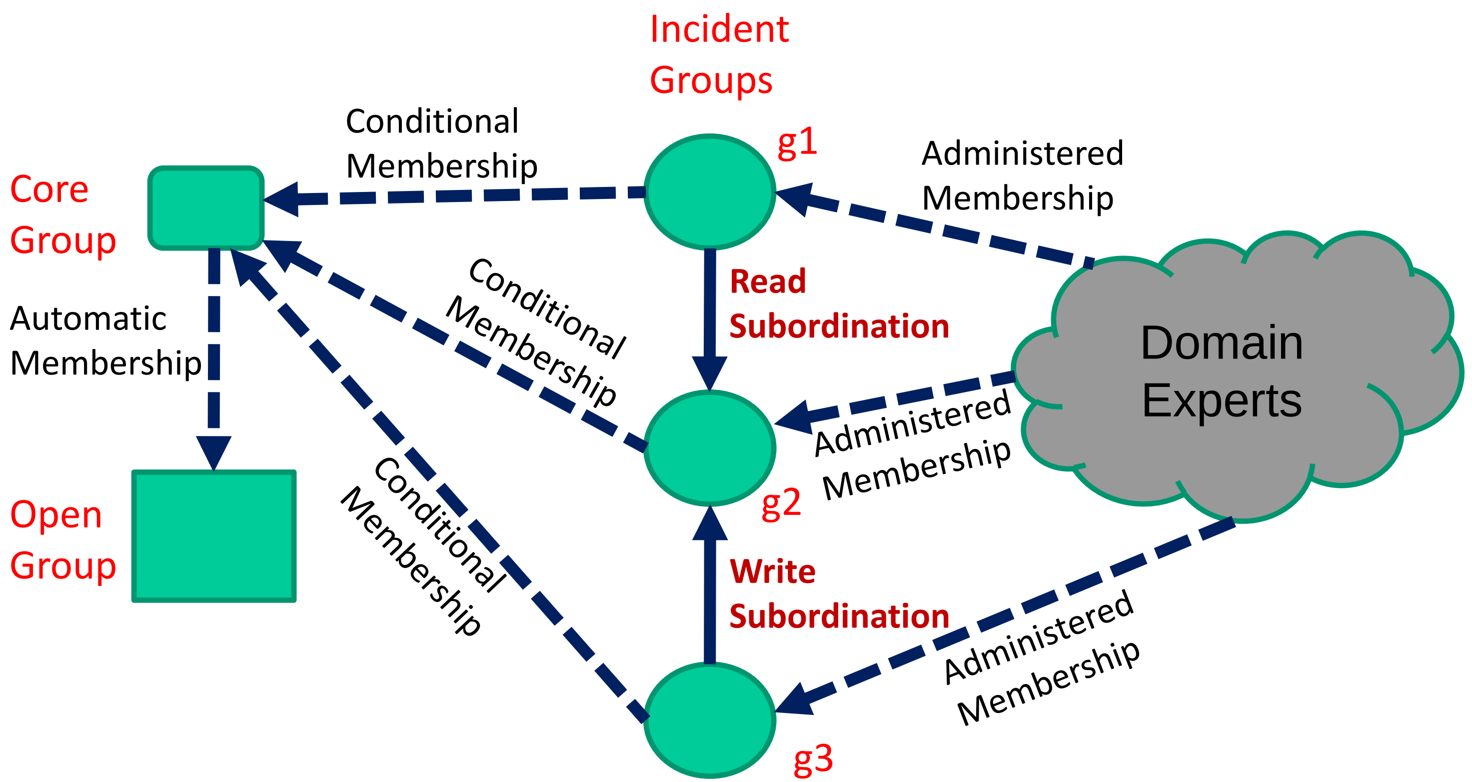
II. Query-Centric Sharing
- Queries wrt a protected dataset
- Privacy/confidentiality protection
- More generally de-aggregation/inference protection

III. Group-Centric Sharing
- Sharing for a purpose
- Mission-centric sharing
- Purpose-centric sharing

Community Cyber Security

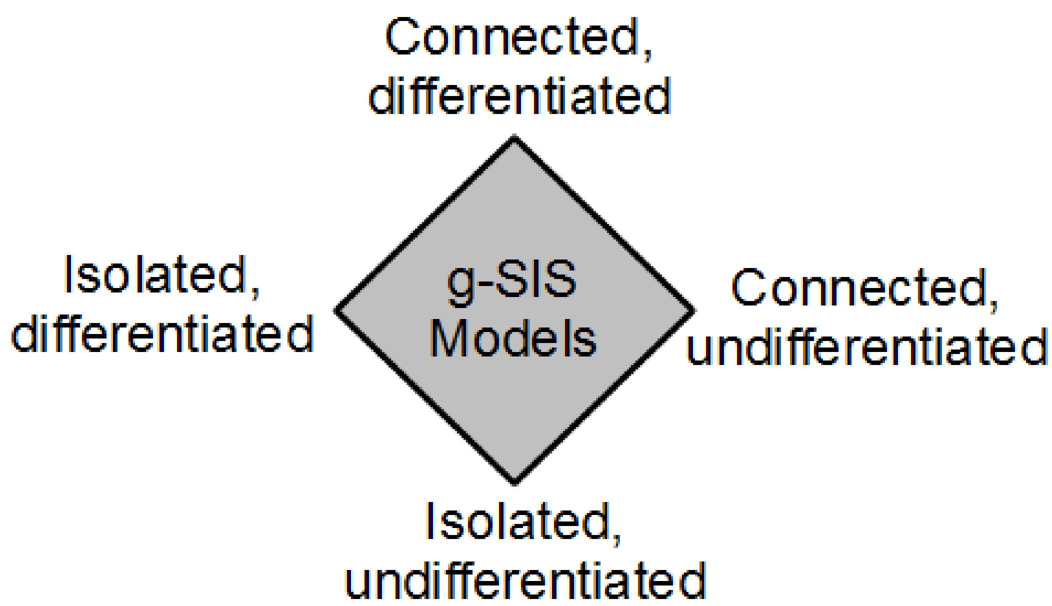
- A community is a county or larger city size unit
  - Clearly demarcated geographical boundary
  - More or less aligned with governance boundary
- The ICS Center for Infrastructure Assurance and Security has a decade long experience conducting cyber security exercises and training for communities all across USA
  - Community cyber security incident life cycle


Filtered RW
Administered Membership
Core Group
Incident Group
Conditional Membership
Automatic Membership
Administered Membership
Administered Membership
Open Group
Filtered RW



Incident
Groups
Conditional
Membership
g1
Administered
Membership
Core
Group
Read
Subordination
Conditional
Membership
Automatic
Membership
Domain
Experts
Administered
Membership
g2
Conditional
Membership
Open
Group
Write
Subordination
Administered
Membership
g3

A Family of g-SIS Models

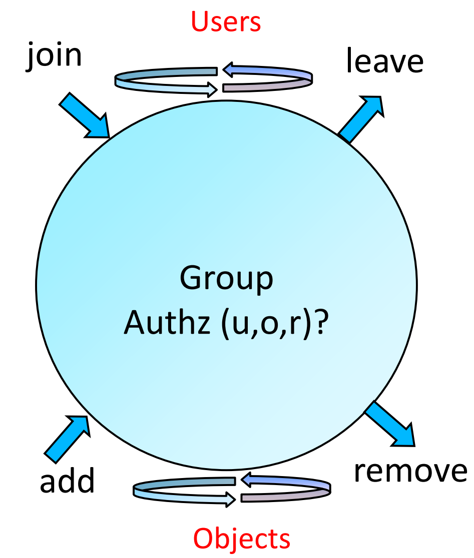


Model Development

- Formal stateless behavioral model with
  - Provable security properties
- Formal stateful enforceable model with
  - Proof of correspondence between stateless and stateful models

g-SIS Model Components

- Operational aspects
  - Group operation semantics
    - Add, Join, Leave, Remove, etc
    - Multicast group is one example
  - Object model
    - Read-only
    - Read-Write (no versioning vs versioning)
  - User-subject model
    - Read-only Vs read-write
  - Policy specification
- Administrative aspects
  - Authorization to create group, user join/leave, object add/remove, etc.



- Authorization Persistence
  - *Authorization cannot change unless some group event occurs*

$$\kappa_0 = \forall u : \mathrm{U}.\forall o : \mathrm{O}.\forall v : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{Authz}(u,o,v,g,\mathbf{r}) \rightarrow (\mathrm{Authz}(u,o,v,g,\mathbf{r})\,\mathcal{W}\,(\mathrm{Join}(u,g) \vee \mathrm{Leave}(u,g) \vee \mathrm{Add}(o,v,g) \vee \mathrm{Remove}(o,v,g))))$$

$$\kappa_1 = \forall u : \mathrm{U}.\forall o : \mathrm{O}.\forall v : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{Authz}(u,o,v,g,\mathbf{w}) \rightarrow (\mathrm{Authz}(u,o,v,g,\mathbf{w})\,\mathcal{W}\,\mathrm{Leave}(u,g)))$$

$$\kappa_2 = \forall u : \mathrm{U}.\forall o : \mathrm{O}.\forall v_1 : \mathrm{V}.\forall g : \mathrm{G}.\exists s : \mathrm{S}.\exists v_2 : \mathrm{V}.$$
$$\Box(\neg\mathrm{Authz}(u,o,v_1,g,\mathbf{r}) \rightarrow (\neg\mathrm{Authz}(u,o,v_1,g,\mathbf{r})\,\mathcal{W}\,(\mathrm{Join}(u,g) \vee \mathrm{Leave}(u,g) \vee \mathrm{Add}(o,v_1,g) \vee \mathrm{Remove}(o,v_1,g) \vee \mathrm{CreateO}(o,v_1,g) \vee \mathrm{update}(s,o,v_2,v_1,g))))$$

$$\kappa_3 = \forall u : \mathrm{U}.\forall o : \mathrm{O}.\forall v_1 : \mathrm{V}.\forall g : \mathrm{G}.\exists s : \mathrm{S}.\exists v_2 : \mathrm{V}.$$
$$\Box(\neg\mathrm{Authz}(u,o,v_1,g,\mathbf{w}) \rightarrow (\neg\mathrm{Authz}(u,o,v_1,g,\mathbf{w})\,\mathcal{W}\,(\mathrm{Join}(u,g) \vee \mathrm{CreateO}(o,v_1,g) \vee \mathrm{update}(s,o,v_2,v_1,g))))$$

The π-system Specification

Table 1: The π-system.

$$\chi_0 = \forall u : \mathrm{U}.\forall o : \mathrm{O}.\forall v : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{Authz}(u,o,v,g,\mathbf{r}) \leftrightarrow \exists v_1 : \mathrm{V}.\exists s : \mathrm{S}.(\lambda_0(u,o,v,g) \vee \ldots \vee \lambda_3(u,s,o,v_1,v,g) \vee \lambda'_0(u,o,v,g) \vee \ldots \vee \lambda'_4(u,s,o,v_1,v,g)))$$

$$\chi_1 = \forall u : \mathrm{U}.\forall o : \mathrm{O}.\forall v : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{Authz}(u,o,v,g,\mathbf{w}) \leftrightarrow \mathrm{Authz}(u,o,v,g,\mathbf{r}) \wedge (\neg \mathrm{Leave}(u,g)\, \mathcal{S}\, \mathrm{Join}(u,g)) \wedge (\exists v_1 : \mathrm{V}.\exists s : \mathrm{S}.\blacklozenge \mathrm{update}(s,o,v_1,v,g) \vee \blacklozenge(\mathrm{LC}(o,v,g) \vee \mathrm{SC}(o,v,g))))$$

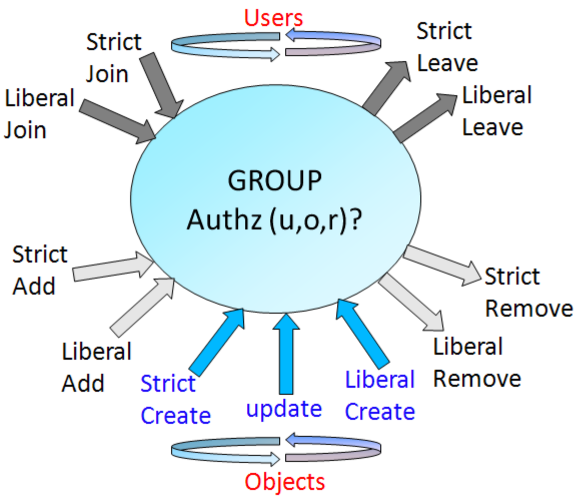
$$\chi_2 = \forall u : \mathrm{U}.\forall s : \mathrm{S}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{createS}(u,s,g) \rightarrow \blacklozenge \mathrm{Join}(u,g))$$

$$\chi_3 = \forall s : \mathrm{S}.\forall o : \mathrm{O}.\forall v : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{AuthzS}(s,o,v,g,\mathbf{r}) \leftrightarrow \exists u : \mathrm{U}.(\mathrm{Authz}(u,o,v,g,\mathbf{r}) \wedge (\neg \mathrm{kill}(u,s,g)\, \mathcal{S}\, \mathrm{createS}(u,s,g))))$$

$$\chi_4 = \forall s : \mathrm{S}.\forall o : \mathrm{O}.\forall v : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{AuthzS}(s,o,v,g,\mathbf{w}) \leftrightarrow \exists u : \mathrm{U}.(\mathrm{Authz}(u,o,v,g,\mathbf{w}) \wedge (\neg \mathrm{kill}(u,s,g)\, \mathcal{S}\, \mathrm{createS}(u,s,g))))$$

$$\chi_5 = \forall s : \mathrm{S}.\forall o : \mathrm{O}.\forall v_1, v_2 : \mathrm{V}.\forall g : \mathrm{G}.$$
$$\Box(\mathrm{read}(s,o,v_1,g) \rightarrow \mathrm{AuthzS}(s,o,v_1,g,\mathbf{r})) \wedge$$
$$\Box(\mathrm{update}(s,o,v_1,v_2,g) \rightarrow \mathrm{AuthzS}(s,o,v_1,g,\mathbf{w}))$$

$$\chi_6 = \forall u_1, u_2 : \mathrm{U}.\forall s_1, s_2 : \mathrm{S}.\forall o : \mathrm{O}.\forall v_1, v_2, v_3 : \mathrm{V}.\forall g_1, g_2 : \mathrm{G}.$$
$$\tau_0(u_1,s_1,s_2,o,v_1,v_2,v_3,g_1) \wedge \ldots \wedge \tau_3(u_1,s_1,o,v_1,g_1)$$

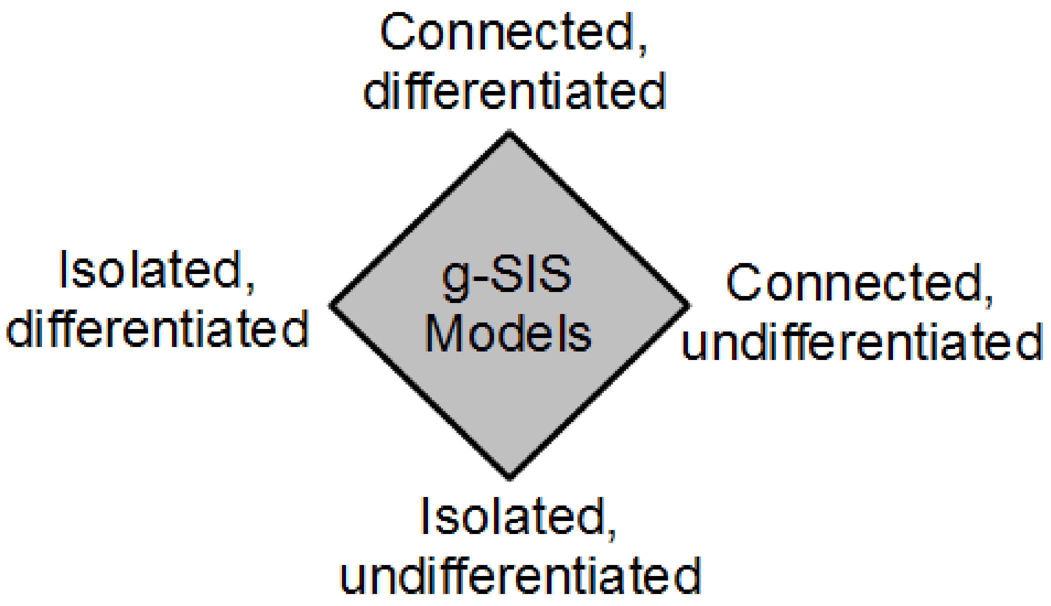


$$\pi = \chi_0 \wedge \chi_1 \wedge \chi_2 \wedge \chi_3 \wedge \chi_4 \wedge \chi_5 \wedge \chi_6$$

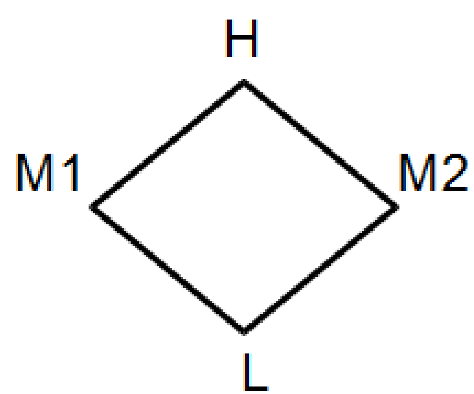
Model Development

- Formal stateless behavioral model with
  - Provable security properties
- Formal stateful enforceable model with
  - Proof of correspondence between stateless and stateful models

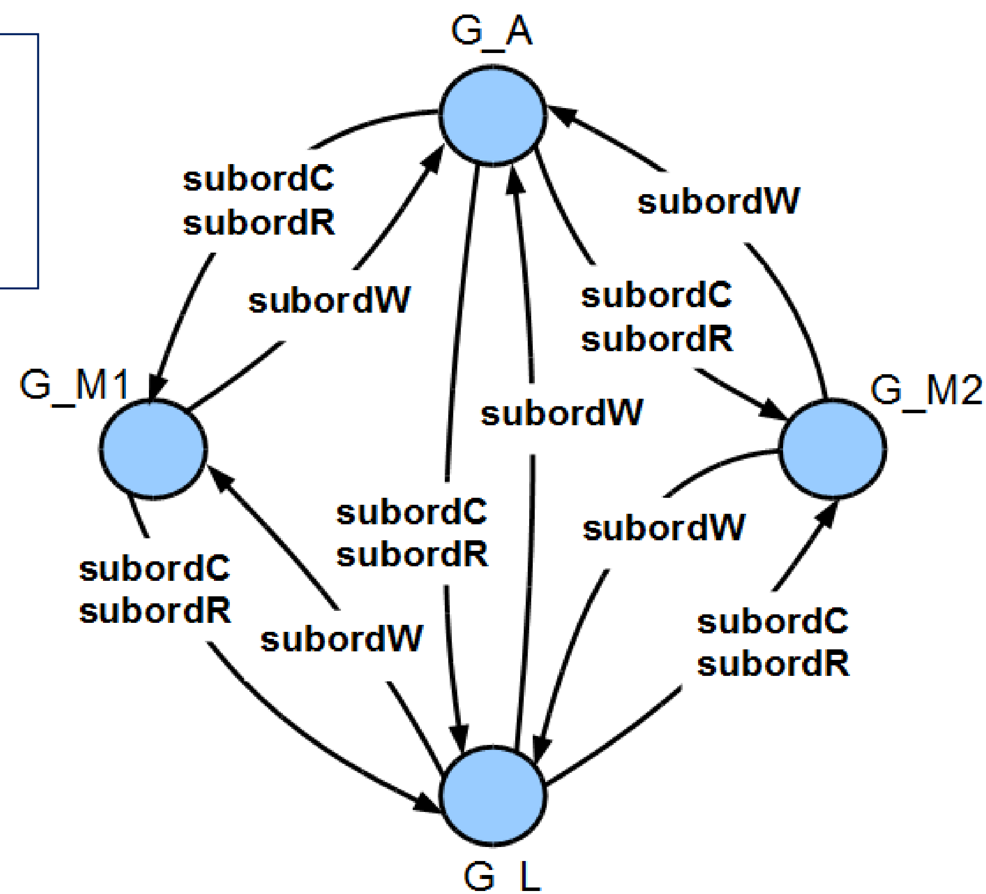
A Family of g-SIS Models



1. Read Subordination
2. Write Subordination
3. Subject Create Subordination



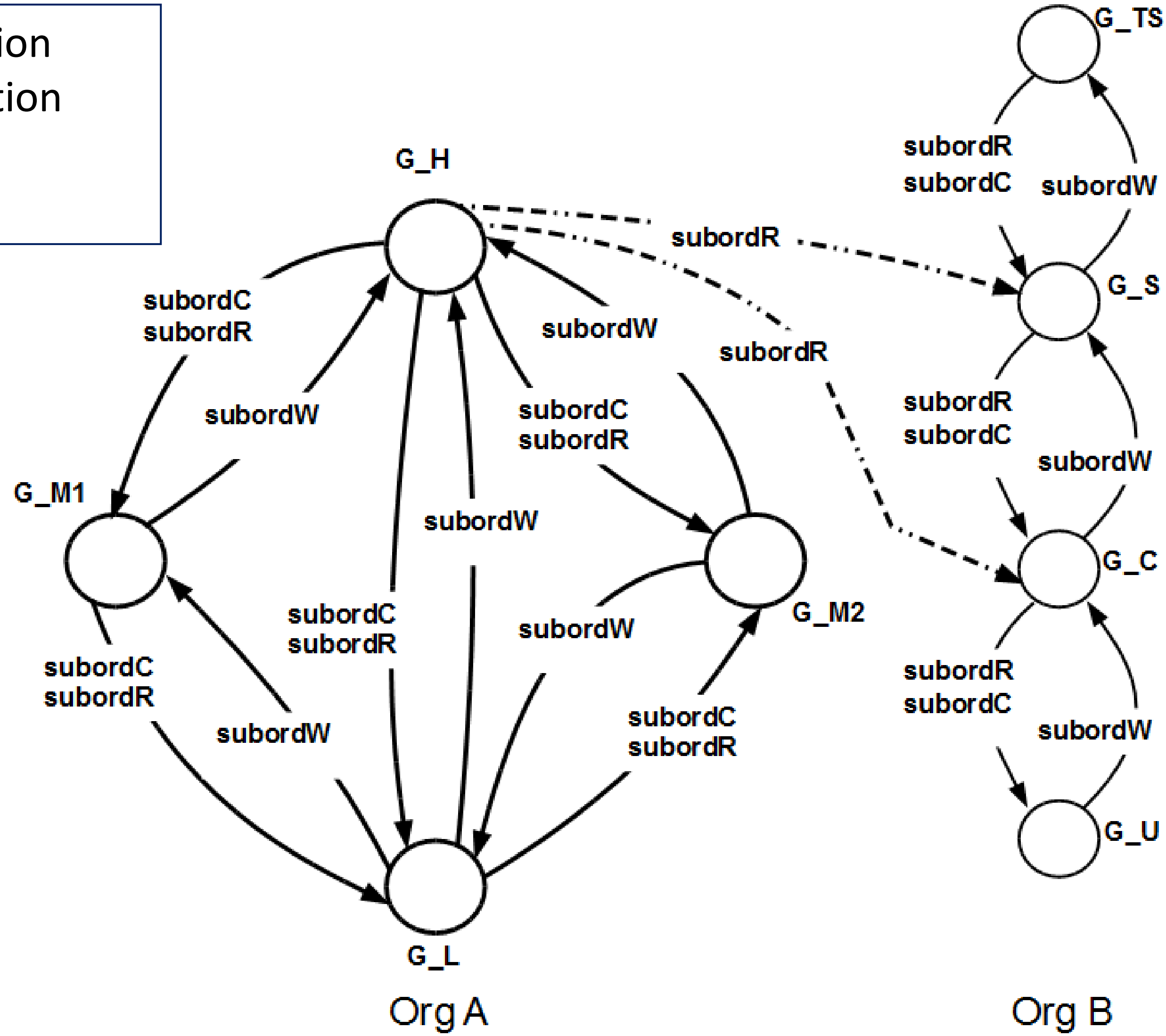
A sample lattice for one directional information flow



Equivalent g-SIS configuration of Org A lattice

Agile Collaboration

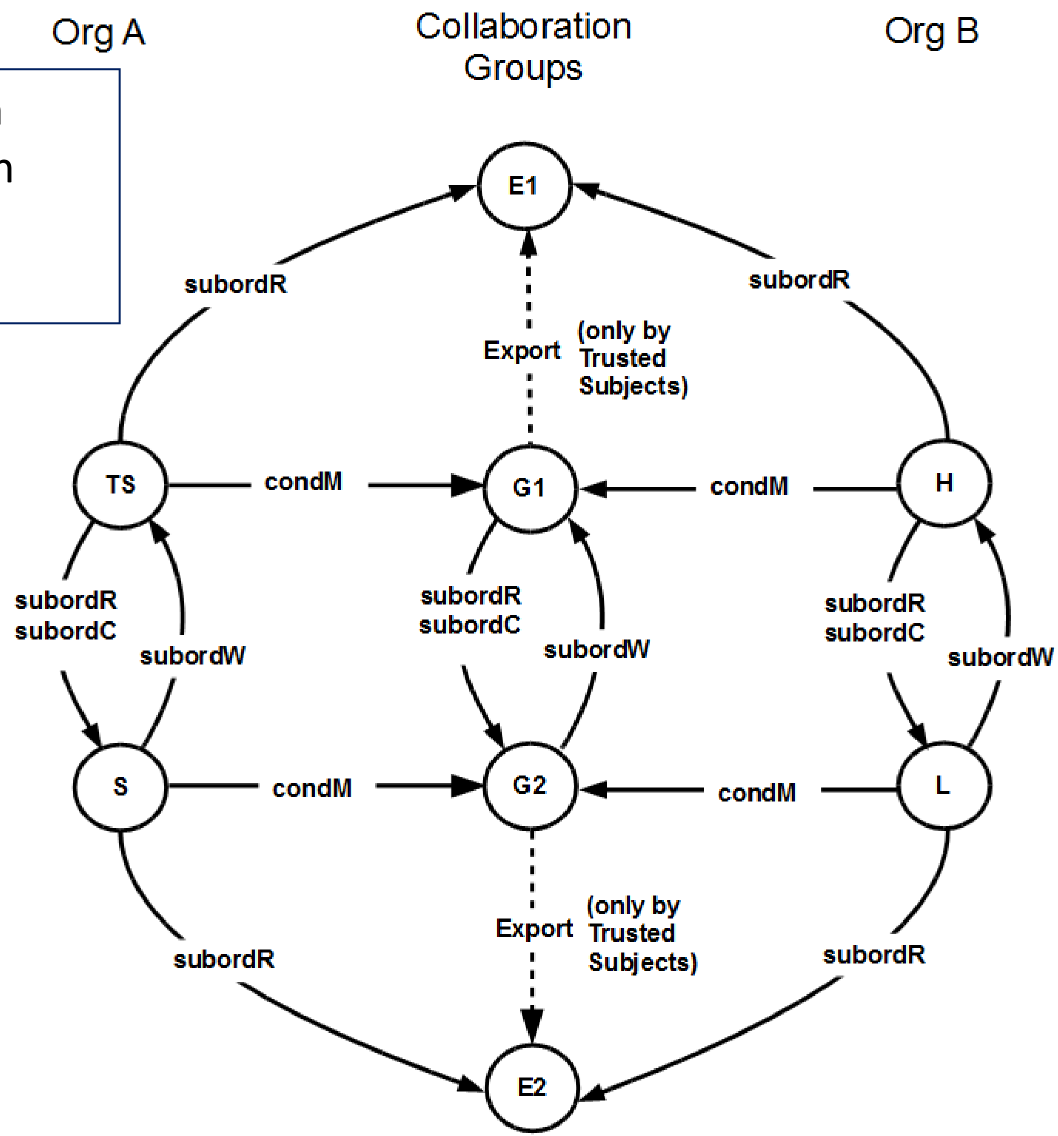
1. Read Subordination
2. Write Subordination
3. Subject Create Subordination



Agile collaboration in LBAC enabled by g-SIS

Agile Collaboration

1. Read Subordination
2. Write Subordination
3. Subject Create Subordination



Collaboration groups established between two different lattices

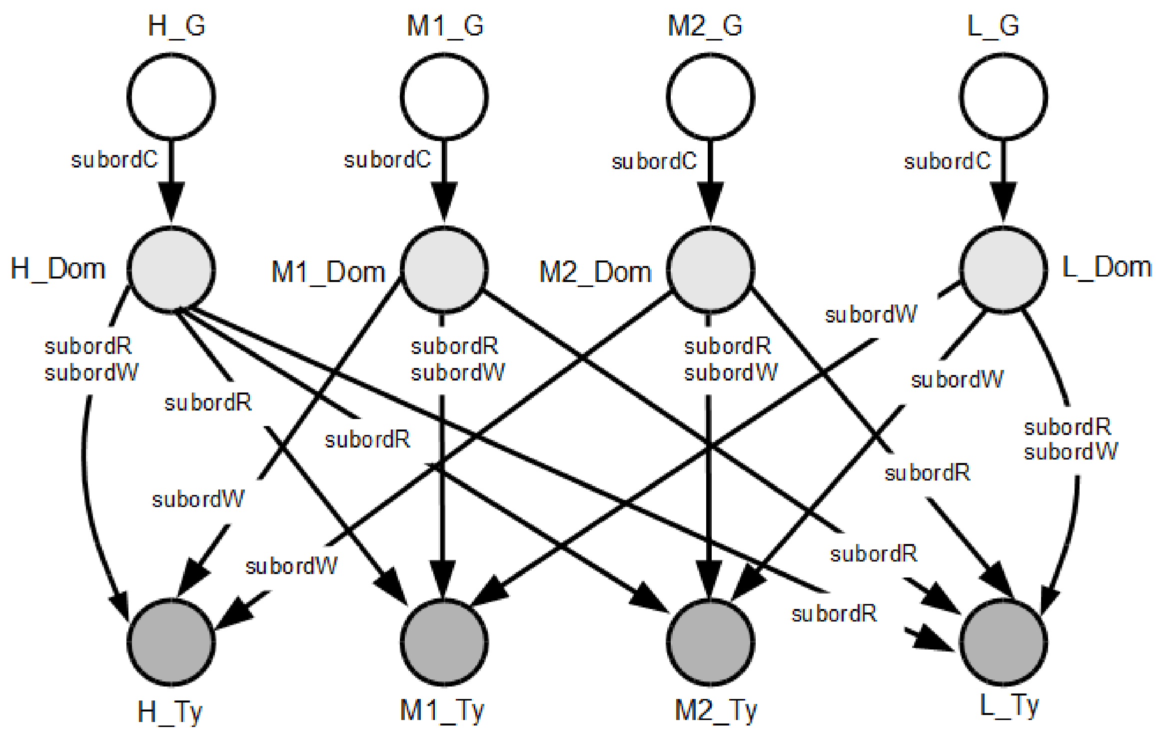
Domain and Type Enforcement and g-SIS

Objects →

Subjects ↓

Domain \ Type	H_Ty	M1_Ty	M2_Ty	L_Ty
H_Dom	*rw*	*r*	*r*	*r*
M1_Dom	*w*	*rw*	-	*r*
M2_Dom	*w*	-	*rw*	r
L_Dom	-	*w*	*w*	*rw*

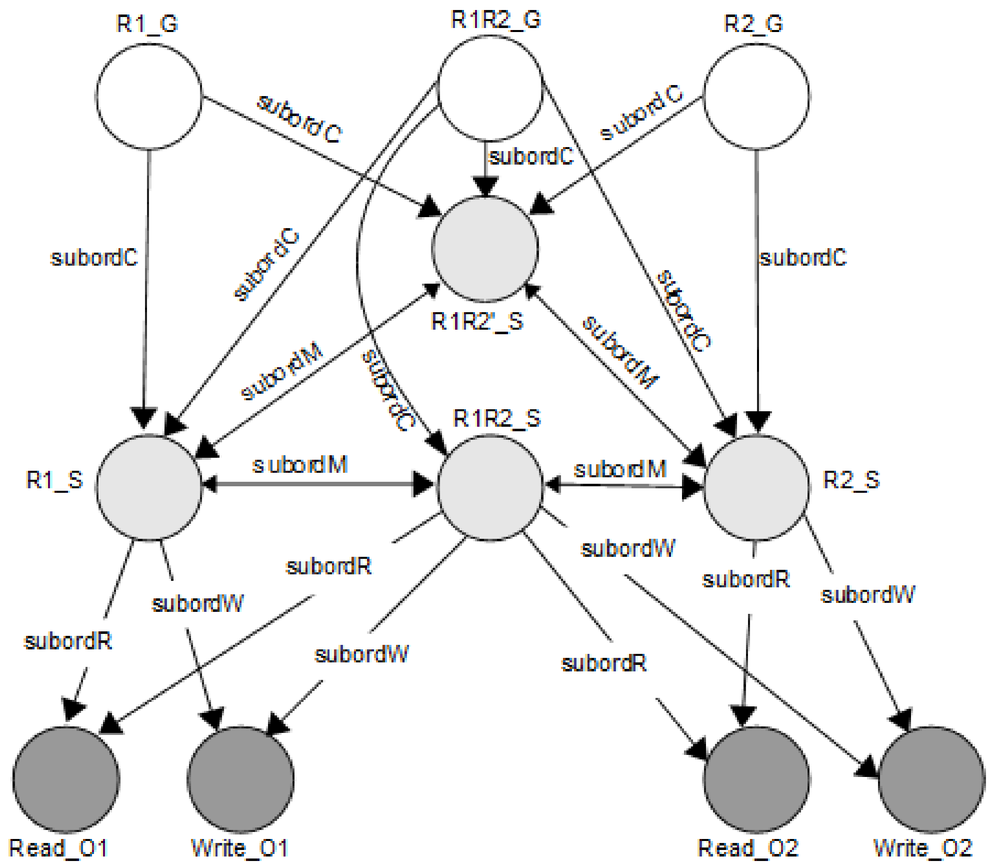
A sample DTE matrix



Equivalent g-SIS configuration

1. Read Subordination
2. Write Subordination
3. Subject Create Subordination

RBAC0 and g-SIS



RBAC$_0$ with RW permissions in g-SIS

1. Read Subordination
2. Write Subordination
3. Subject Create Subordination
4. Subject Move Subordination

Goal: 4th Element

- 3 succesful access control models in 40+ years
  - Discretionary Access Control (DAC)
  - Mandatory Access Control (MAC)
    also called Lattice-Based Access Control (LBAC)
  - Role-base Access Control (RBAC)
- Numerous others defined and studied, implemented but no success
- Will Group Centric Models be the 4th element?
  - Strong mathematical foundations
  - Strong intuitive foundations
  - Significant real-world deployment