Towards Secure Information Sharing Models for Community Cyber Security

Ravi Sandhu, Ram Krishnan and Gregory B. White
Institute for Cyber Security
University of Texas at San Antonio

Secure Information Sharing (SIS)

- Share *but* protect

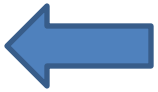
Saltzer-Schroeder[1] identified the desirability and difficulty of maintaining:
"some control over the user of the information even after it has been released"

[1]J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of IEEE, 63(9):1278–1308, 1975.*

SIS Major Challenges

- Policy Challenge
  - Modeling, specifying and enforcing SIS policies
  - Need intuitive yet formal models, guaranteed security properties, etc.
- Containment Challenge
  - Ensure that protected information is accessible to users as permitted by the policy
  - Security mechanisms such as authentication, cryptography, trusted hardware, etc.

SIS Major Challenges

- Policy Challenge 
  - Modeling, specifying and enforcing SIS policies
  - Need intuitive yet formal models, guaranteed security properties, etc.
- Containment Challenge
  - Ensure that protected information is accessible to users as permitted by the policy
  - Security mechanisms such as authentication, cryptography, trusted hardware, etc.

Community Cyber Security

- Community refers to a geographical area
  - E.g. county or a city with demarcated boundary
- The Center for Infrastructure Assurance and Security at UTSA conducts nation-wide cyber security preparedness exercises and training
  - communication
  - incident response
  - disaster recovery
  - business continuity
  - security awareness, etc.

The Current Status…

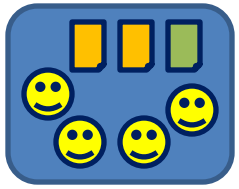
- Exchange of business cards
  - No process exists for information sharing
- Technology is not the bottleneck
  - Resistance due to political/competitive reasons
  - Also want to avoid embarrassment
    - E.g. by sharing attack data
- Participants have no clue as to *what* to share and *how* to effectively specify what to share

Requirements

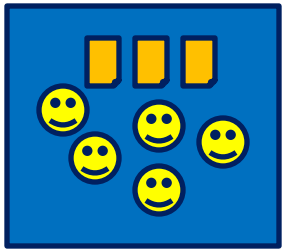
- Need abstract models
  - With rigorous mathematical foundations
  - Should ease administration
- Classic models are limited
  - Discretionary Access Control
    - Too low-level to configure
  - Lattice-Based Access Control (E.g. Bell LaPadula)
    - Rigid
    - One directional info flow is not the primary concern
  - Lot of work on Dynamic Coalitions
    - Many times heavy-weight
    - Mainly focus on technological/infrastructural integration

Life-Cycle of a Cyber Incident
Secure Sharing in a Community

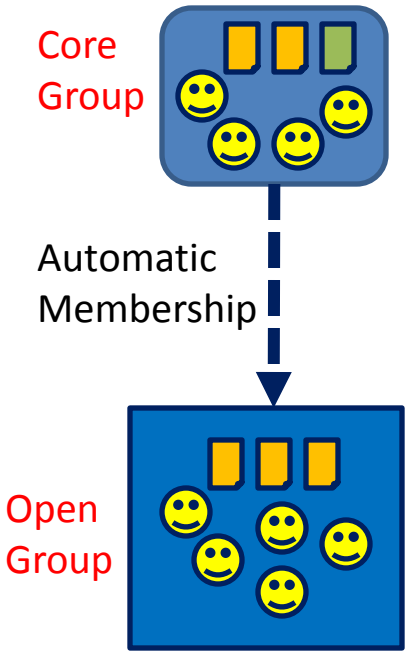
Core Group



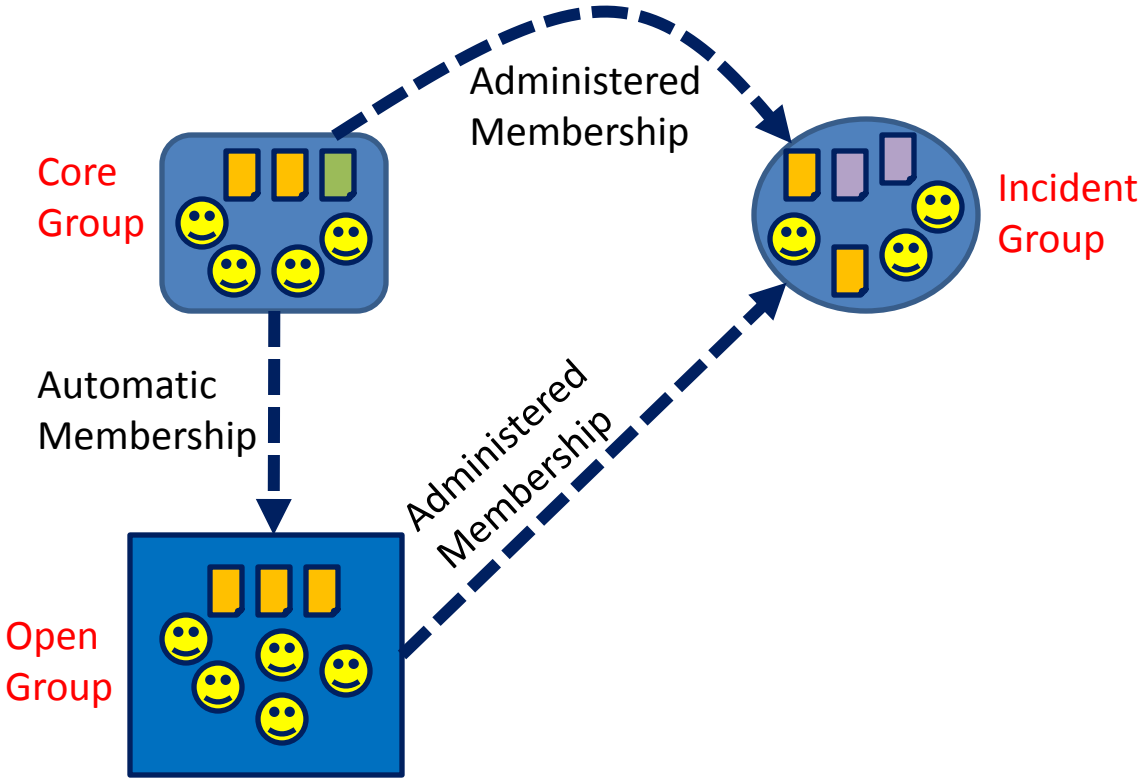
Open Group



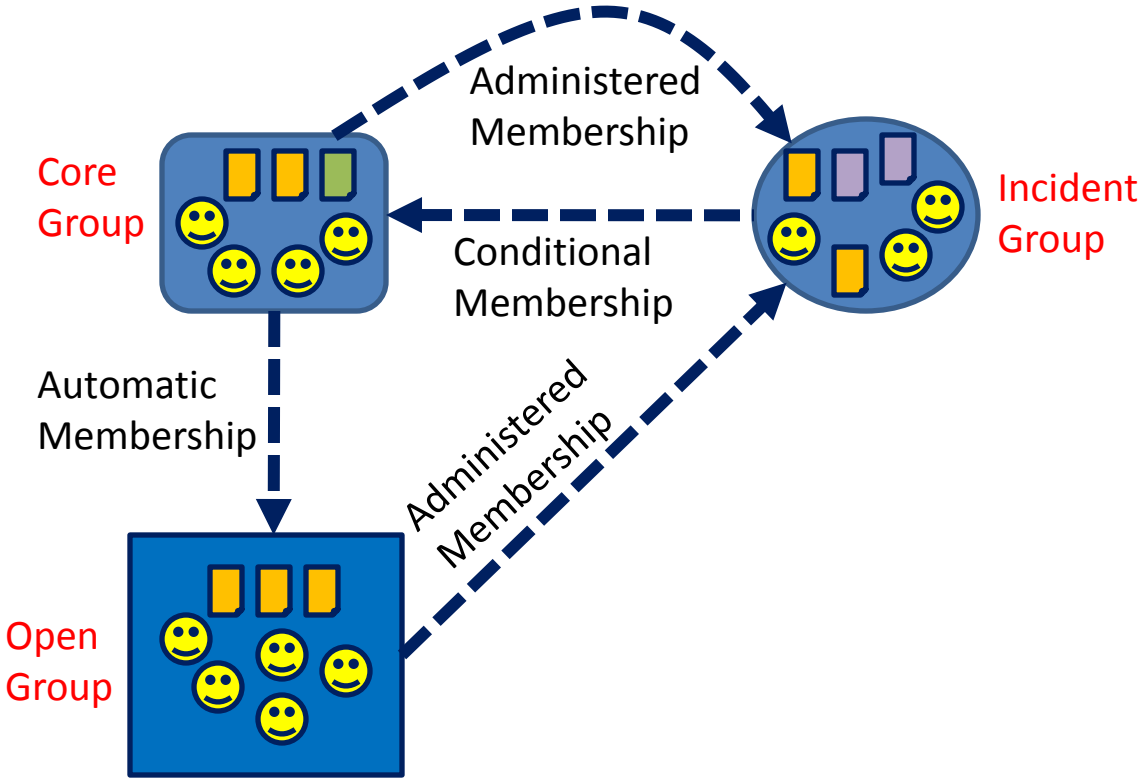
Life-Cycle of a Cyber Incident
Secure Sharing in a Community



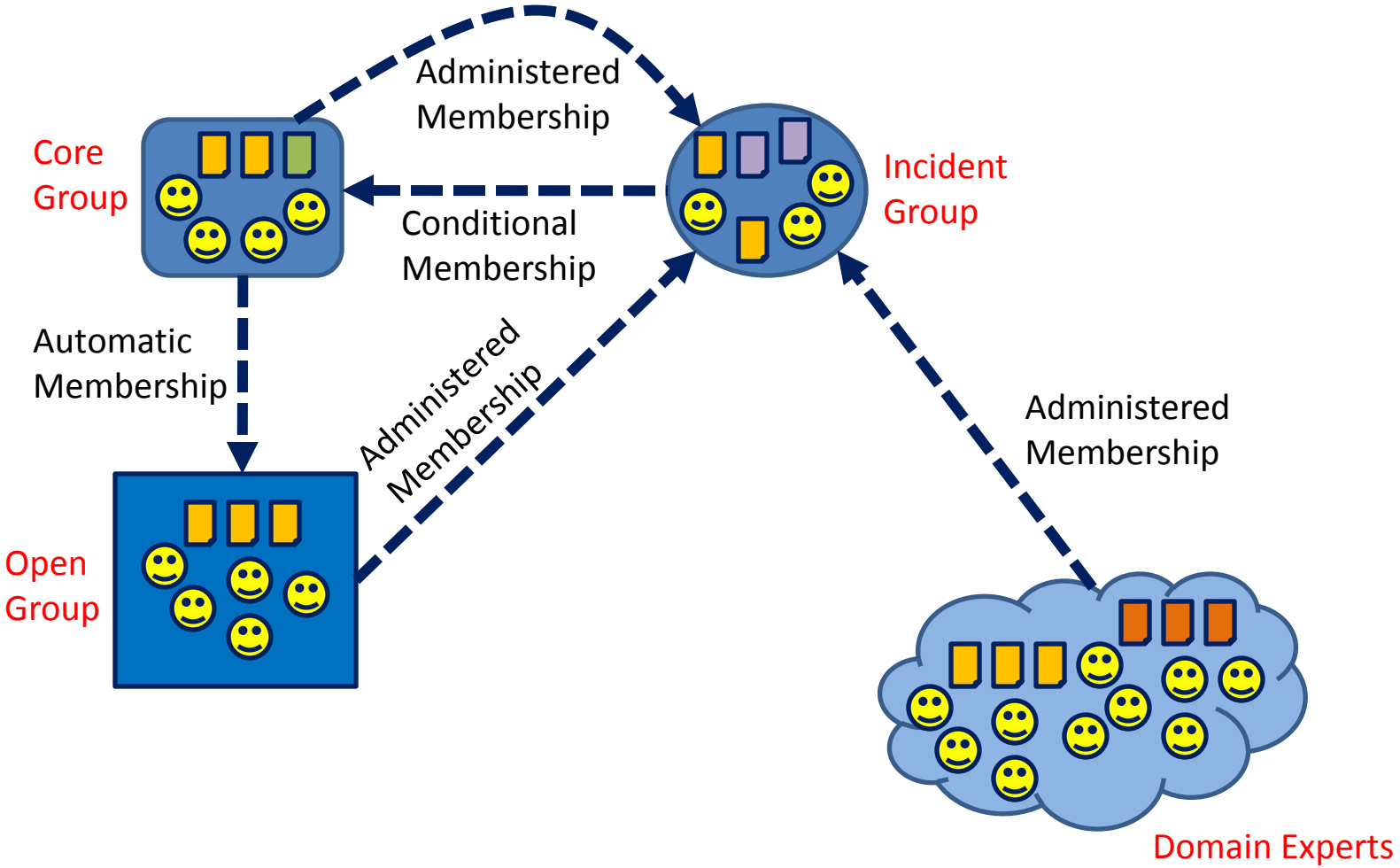
Life-Cycle of a Cyber Incident Secure Sharing in a Community



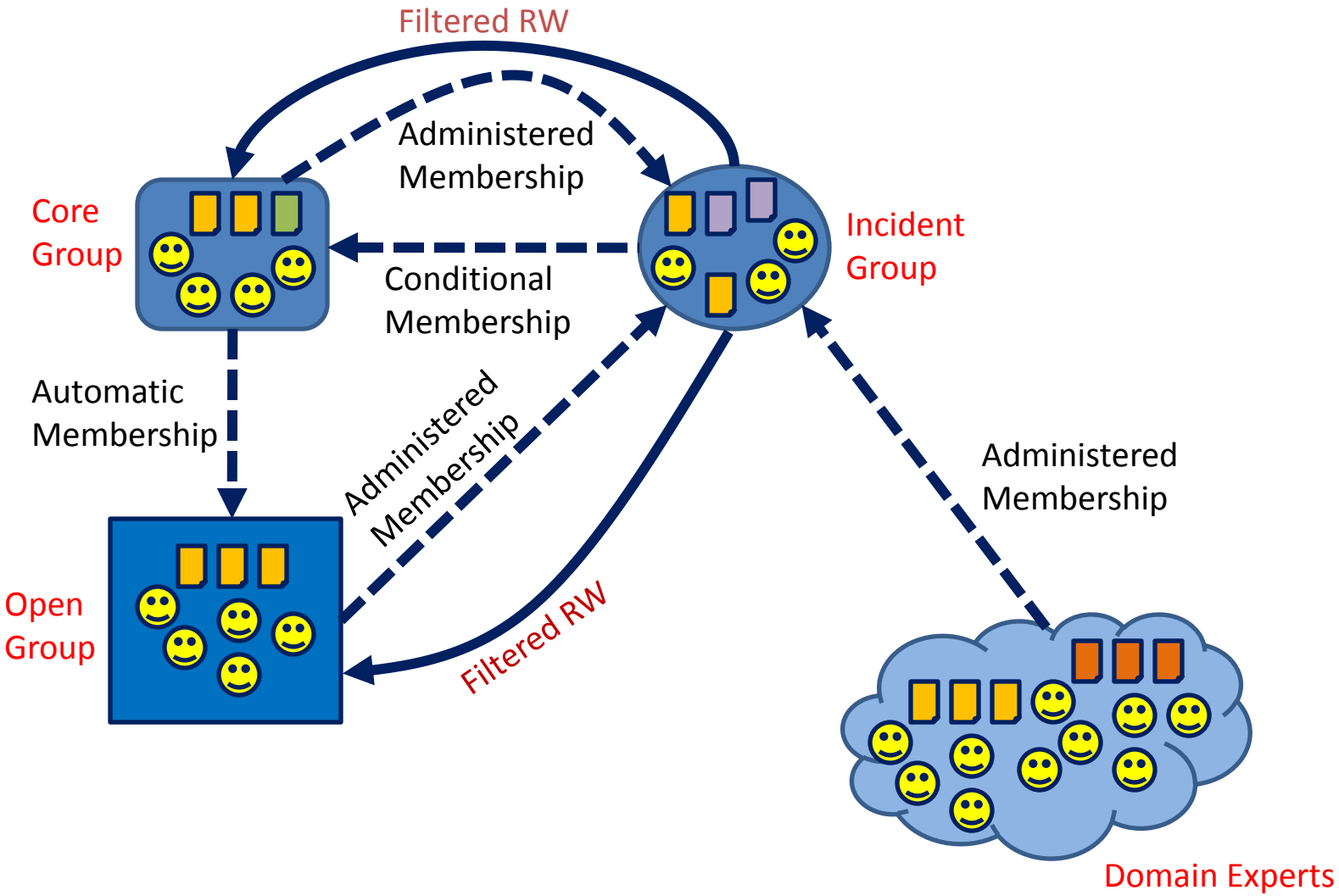
Life-Cycle of a Cyber Incident Secure Sharing in a Community



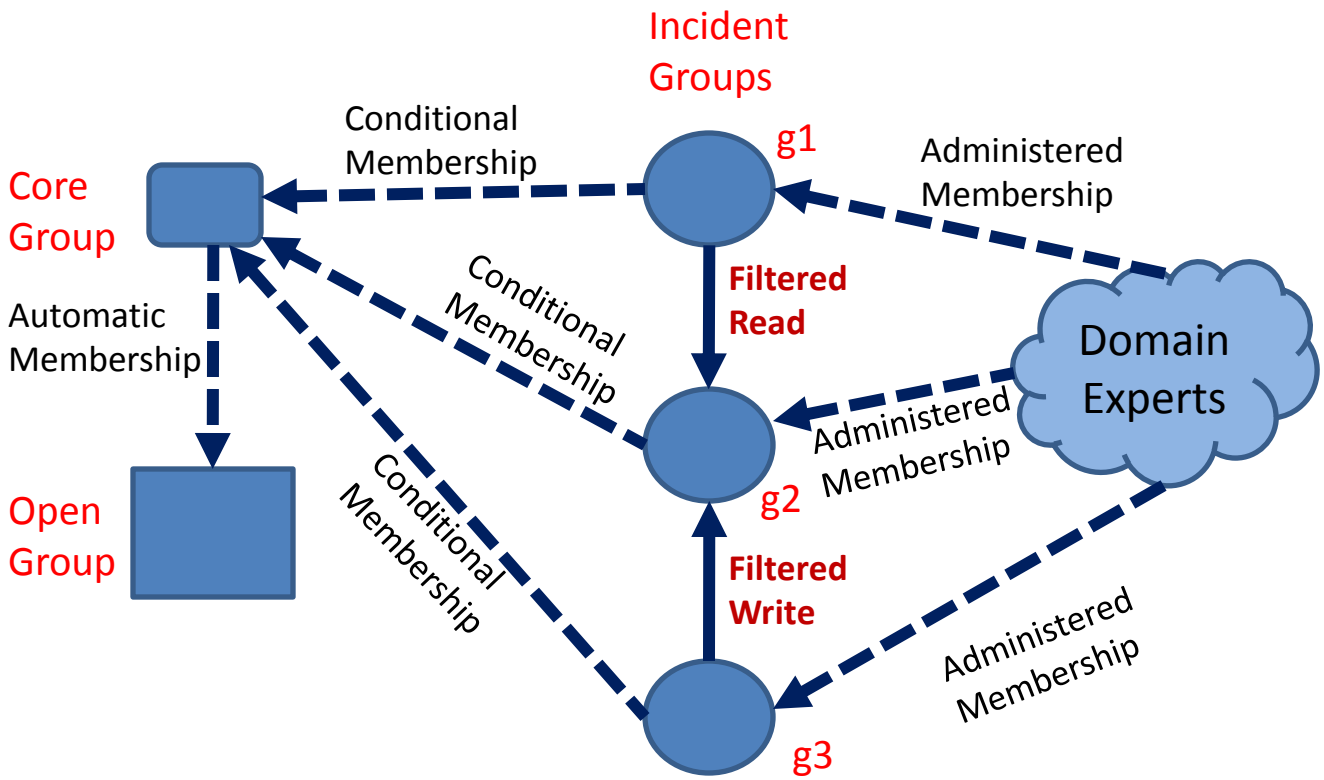
Life-Cycle of a Cyber Incident Secure Sharing in a Community



Life-Cycle of a Cyber Incident Secure Sharing in a Community

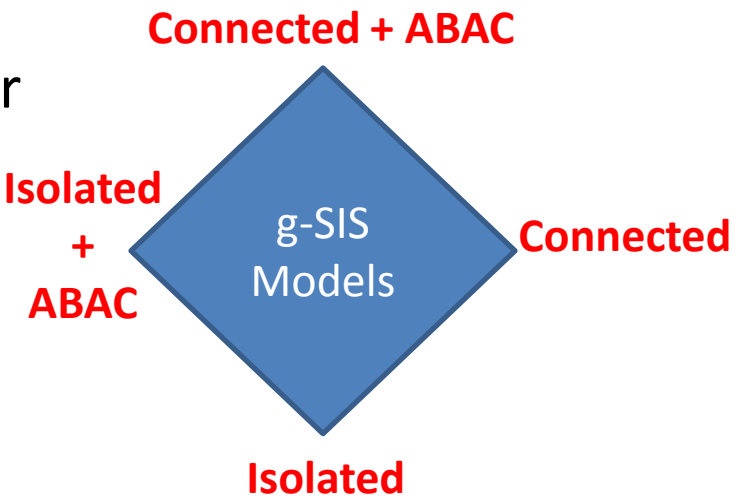


Life-Cycle of Cyber Incident Secure Sharing in Community (contd)



A Family of Group-Centric SIS Models

- Isolated
  - Users and objects are isolated
  - Membership in one group has no impact on authorizations in another group
- Connected
  - Membership in one group impacts authorization in another
  - E.g. Subordination, conditional membership, mutual exclusion, etc.
- Attribute-Based Access Control
  - For fine-grained authorization



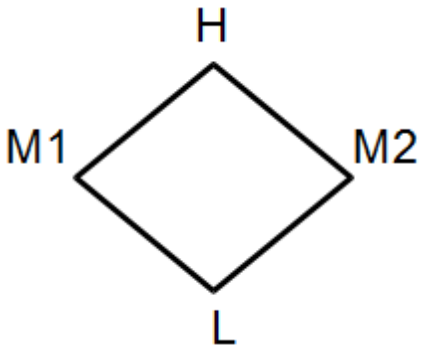
Conclusion

- SIS is still an open problem
- Technology is relatively under control
- Policy specification is key to SIS
  - Clear, usable and friendly policies can overcome political and competitive barriers to SIS
- One size does not fit all
  - Domain and application specific modeling and analysis is needed

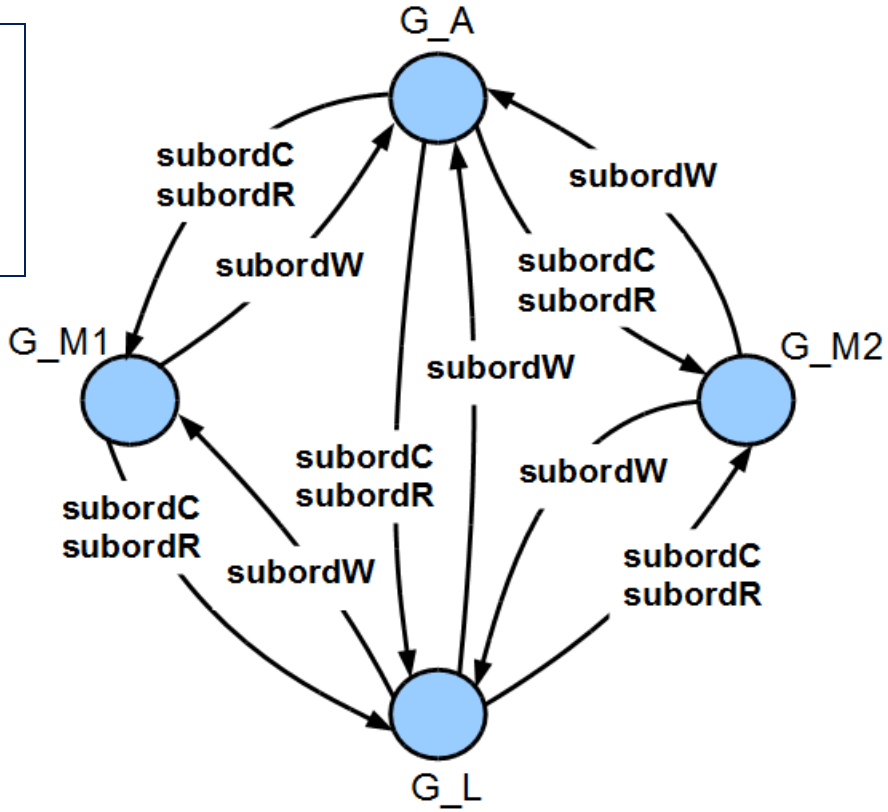
Backup

g-SIS and LBAC

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



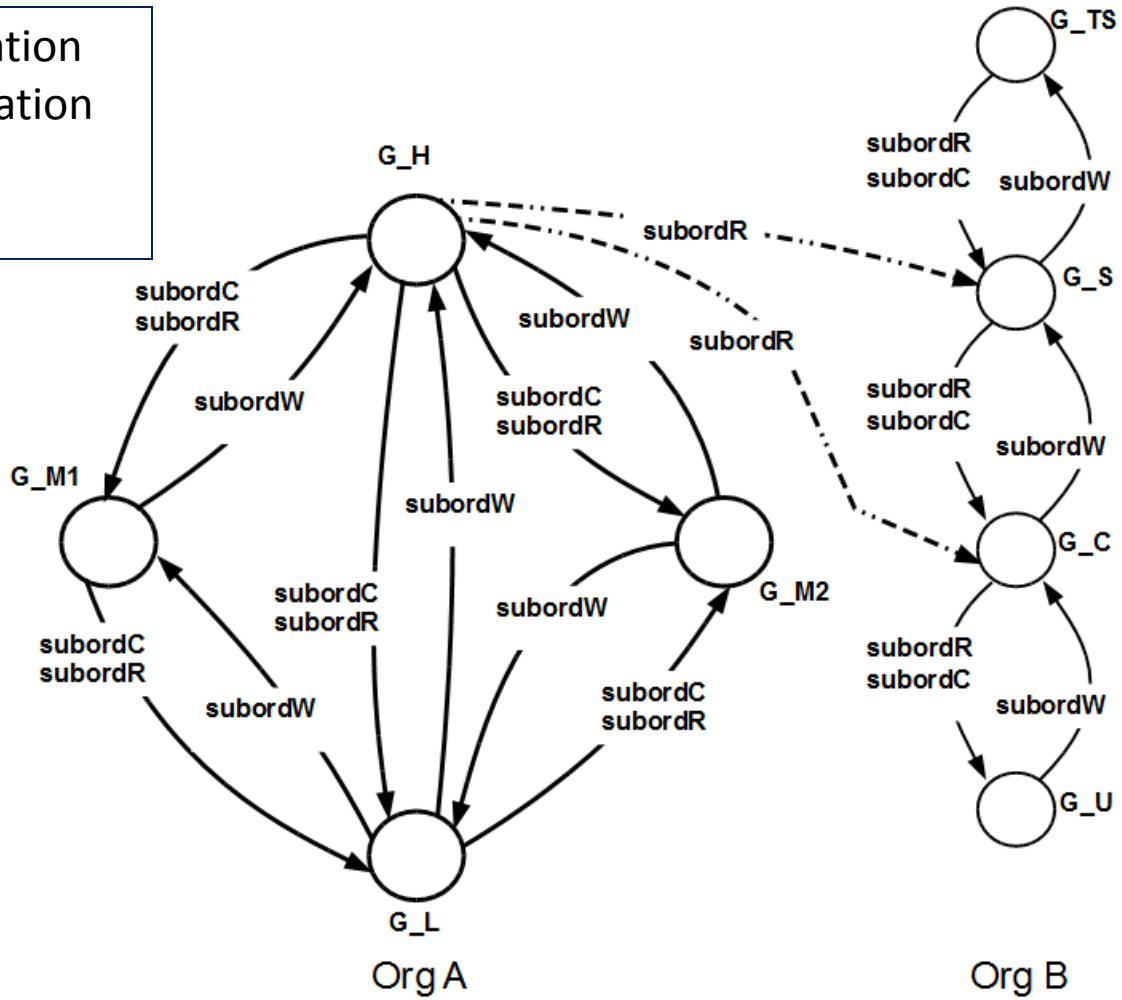
A sample lattice for one directional information flow



Equivalent g-SIS configuration of Org A lattice

Agile Collaboration

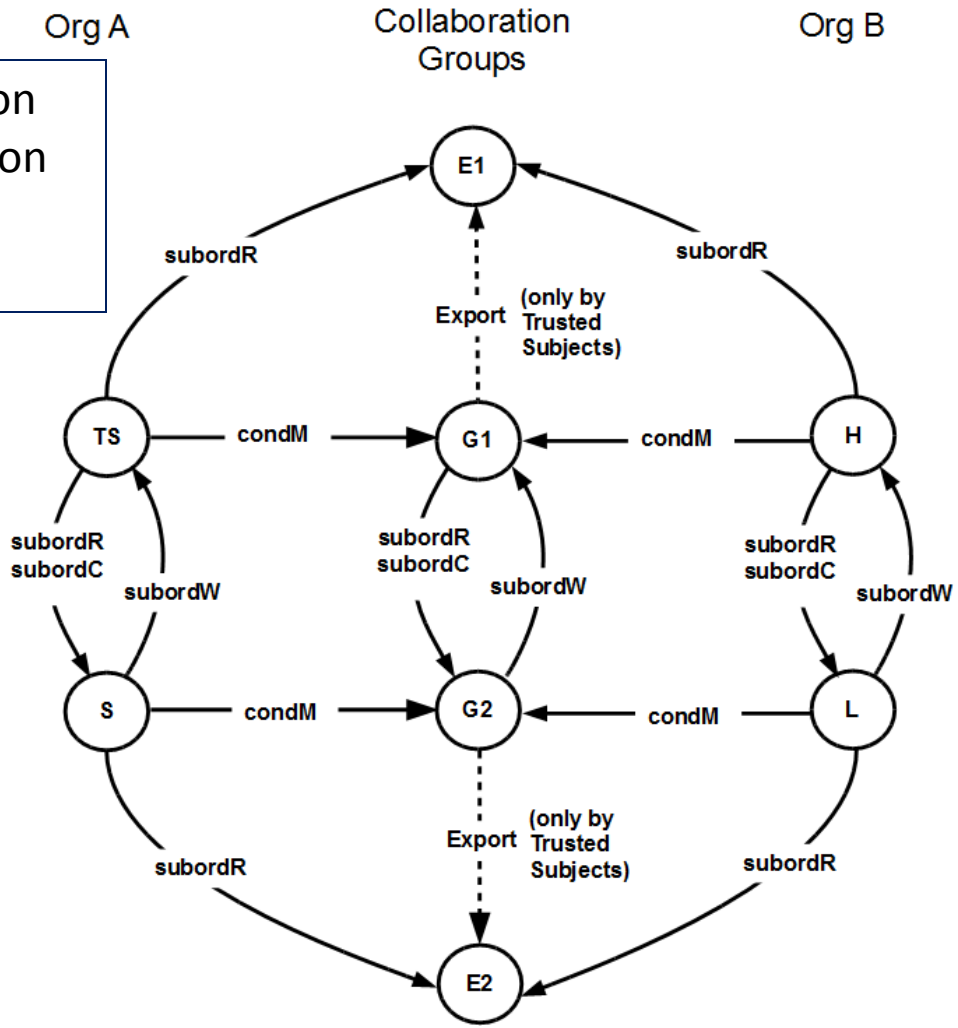
1. Read Subordination
2. Write Subordination
3. Subject Create Subordination



Agile collaboration in LBAC enabled by g-SIS

Agile Collaboration (continued)

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



Collaboration groups established between two different lattices

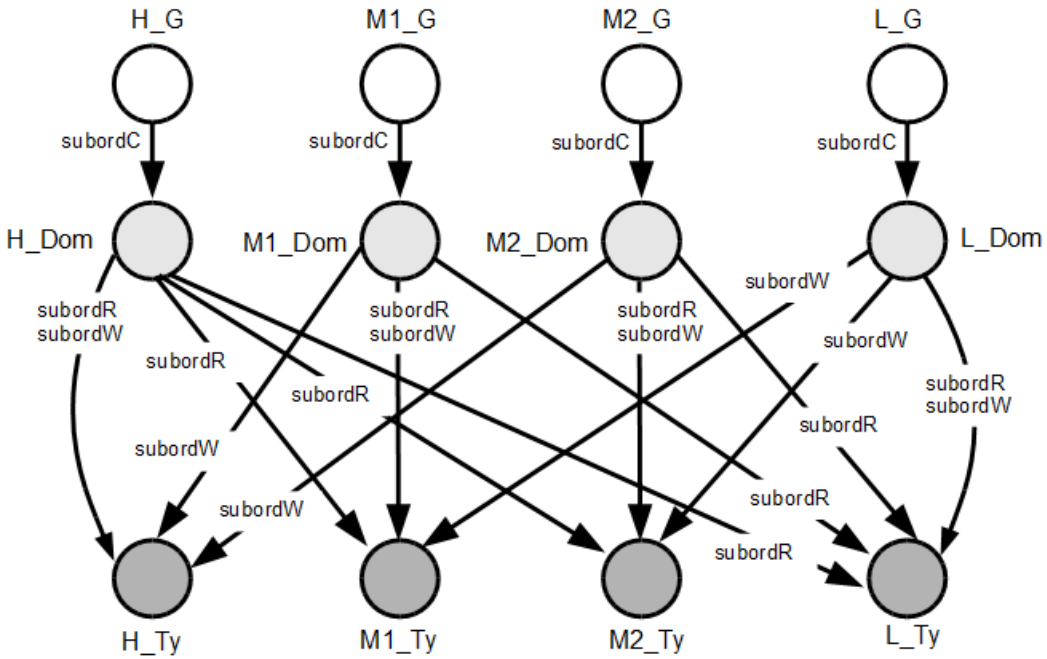
Domain and Type Enforcement and g-SIS

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*

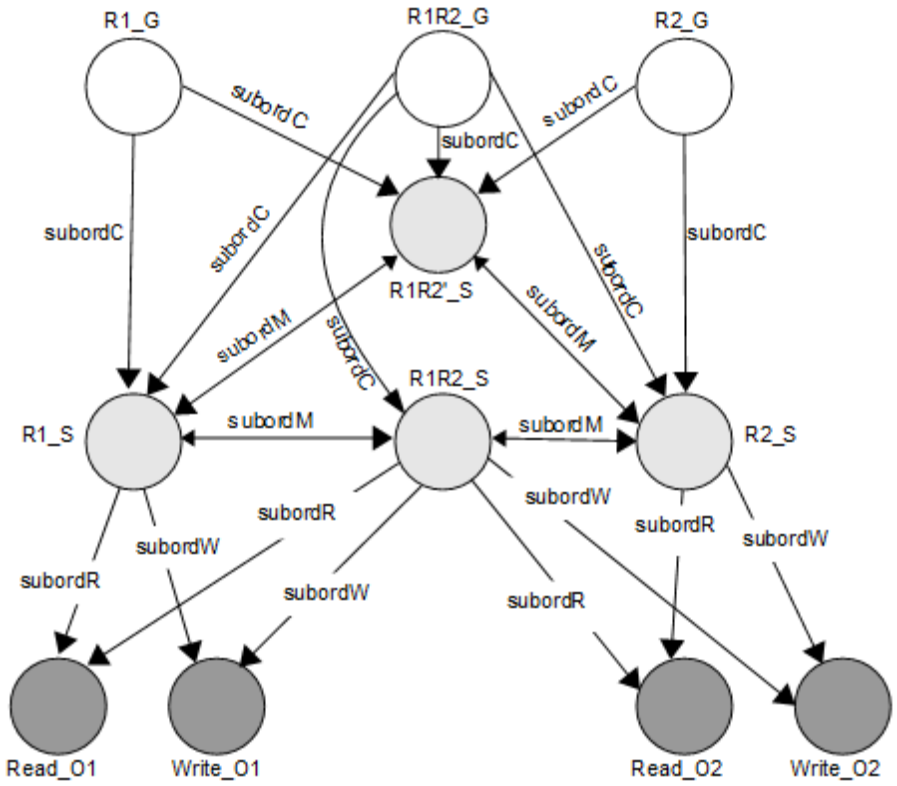
A sample DTE matrix



Equivalent g-SIS configuration

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

$RBAC_0$ and g-SIS



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

$RBAC_0$ with RW permissions in g-SIS