

# Foundations for Group-Centric Secure Information Sharing Models

Ram Krishnan (George Mason University)  
Ravi Sandhu, Jianwei Niu, William Winsborough  
(University of Texas at San Antonio)

ACM Symposium on Access Control Models and Technologies (SACMAT 2009)  
June 3-5, Stresa, Italy

# Presentation Outline

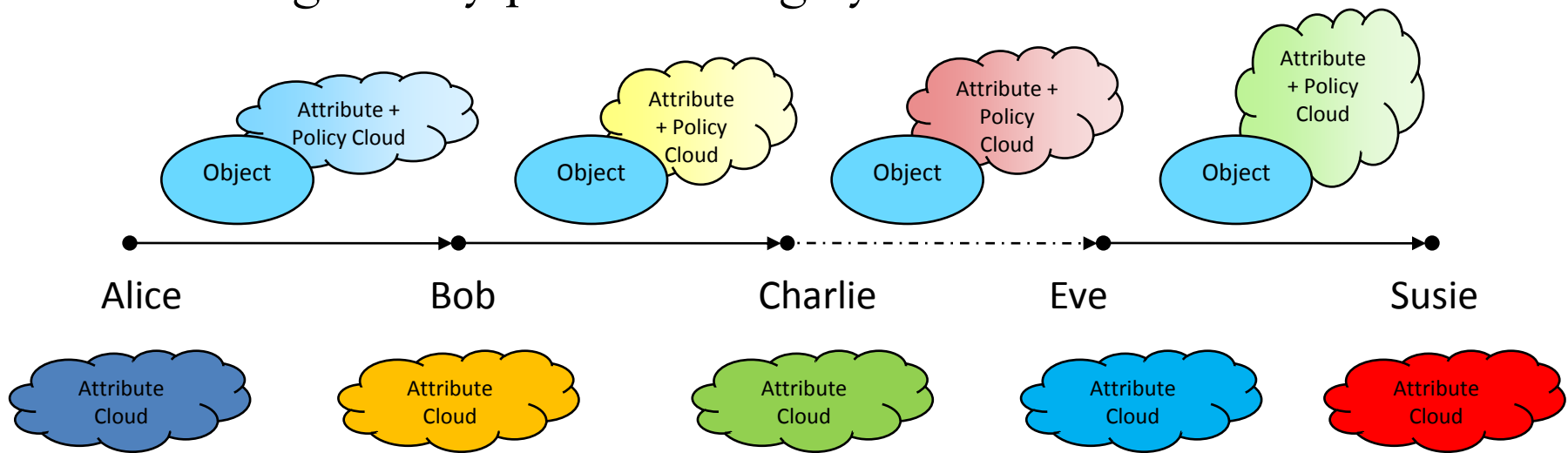
- Motivation for g-SIS
  - Group-Centric Vs Dissemination-Centric SIS
- Core g-SIS properties
- Group operation semantics
- $\pi$ -system g-SIS specification
- Verification of  $\pi$ -system
- Conclusion

# Secure Information Sharing (SIS)

- Share *but* protect
  - A fundamental problem in cyber security
- Traditional models do capture important SIS aspects
  - But not satisfactory
  - Discretionary Access Control (owner control)
    - Too fine-grained, lacks copy control
  - Bell-LaPadula (information flow)
    - Too rigid and coarse-grained
  - Role-Based Access Control (effective administration)
    - Too general and does not directly address information sharing
  - UCON/ABAC also too general
- Primary issues
  - Copy control
  - Manageability

# Dissemination-Centric Sharing

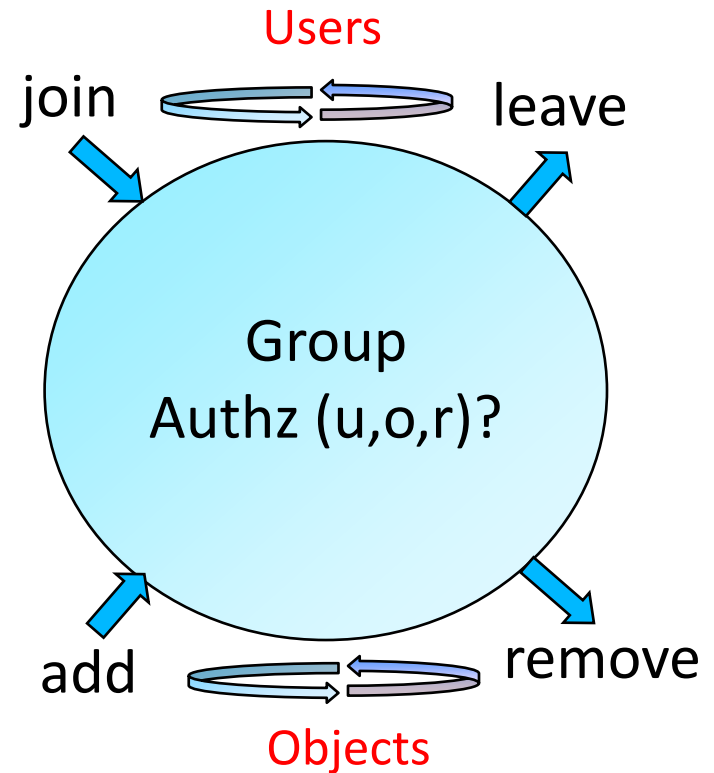
- Extensive research in the last two decades
  - ORCON, DRM, ERM, XrML, ODRL, etc.
- Copy/usage control has received major attention
- Manageability problem largely unaddressed



Dissemination Chain with Sticky Policies on Objects

# Group-Centric Sharing (g-SIS)

- Brings users & objects together in a group
  - Focuses on manageability using groups
  - Co-exists with dissemination-centric
  - Two metaphors
    - Secure Meeting Room (E.g. Program committee meeting)
    - Subscription Model (E.g. Secure multicast)
- Operational aspects
  - Group characteristics
    - E.g. Are there any core properties?
  - Group operation semantics
    - E.g. What is authorized by join, add, etc.?
  - Read-only Vs Read-Write
- Administrative aspects
  - E.g. Who authorizes join, add, etc.?
  - May be application dependant
- Multiple groups
  - Inter-group relationship



# Roles Vs Groups in SIS

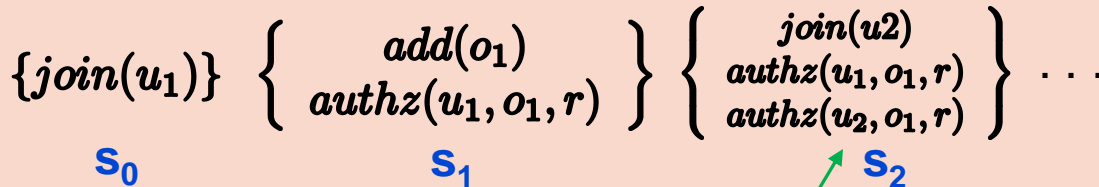
- Roles
  - Users get same set of privileges on role assignment
  - Does not consider timing of assignment/activation
  - Temporal RBAC considers specific timing aspects
    - E.g. authorizations for when a role can be activated
- Groups
  - Privileges may differ with time of join, leave, etc.
  - Sharing is promoted within and across groups
  - Inter-group relationship may differ from that of roles

# Formalization of g-SIS

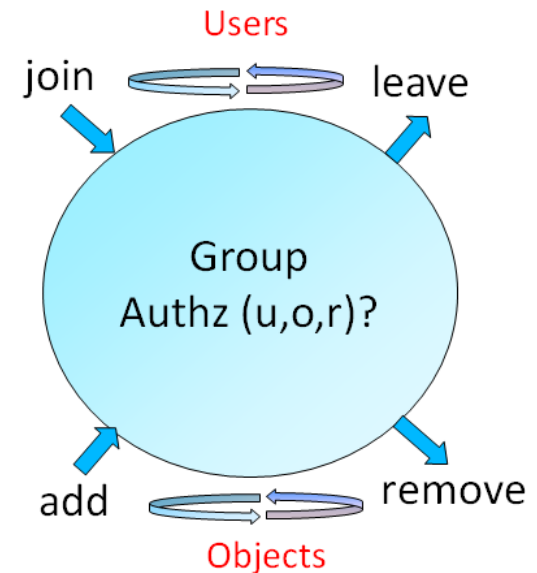
# Terminology

- A *state* in g-SIS is a function from predicates to  $\{\text{True}, \text{False}\}$ 
  - Predicates include join, leave, add and remove
  - Authorization depends on type of join, leave, add and remove
- A *trace* is an infinite sequence of states

A sample g-SIS trace



May depend on type  
of  $join(u_2)$  and  $add(o_1)$





# Notations

- Use Join, Leave, Add and Remove to refer to some respective event type occurring

$$\text{Join}(u) = (\text{join}_1(u) \vee \text{join}_2(u) \vee \dots \vee \text{join}_m(u))$$

$$\text{Leave}(u) = (\text{leave}_1(u) \vee \text{leave}_2(u) \vee \dots \vee \text{leave}_n(u))$$

$$\text{Add}(o) = (\text{add}_1(o) \vee \text{add}_2(o) \vee \dots \vee \text{add}_p(o))$$

$$\text{Remove}(o) = (\text{remove}_1(o) \vee \dots \vee \text{remove}_q(o))$$

- Drop the parameters for convenience

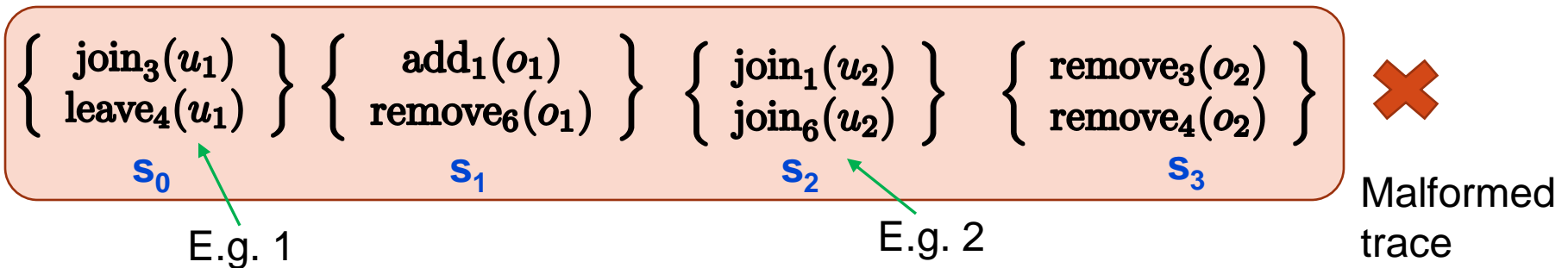
$$\text{Authz} \rightarrow (\text{Join} \wedge (\neg(\text{Leave} \vee \text{Remove})))$$

≡

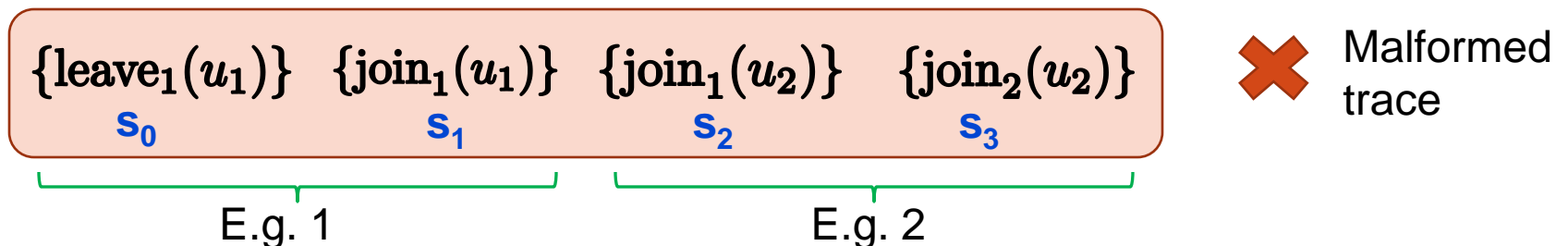
$$\forall u \in U. \forall o \in O. \text{Authz}(u, o, r) \rightarrow (\text{Join}(u) \wedge (\neg(\text{Leave}(u) \vee \text{Remove}(o))))$$

# Well-Formed Traces

- Multiple events cannot occur in a state for the same user (or object)
  - E.g. 1 User cannot join and leave in the same state
  - E.g. 2 Two types of join cannot occur in the same state



- User events should occur alternatively beginning with a join event
  - E.g. 1 leave cannot occur before join
  - E.g. 2 join should be followed by a leave before another join



# LTL Specification of Well-Formed Traces

$$\tau_0 = \Box(\neg(\text{Add} \wedge \text{Remove}) \wedge \neg(\text{Join} \wedge \text{Leave}))$$

---

$$\begin{aligned}\tau_1 = & \forall i, j \Box((i \neq j) \rightarrow \neg(\text{join}_i \wedge \text{join}_j)) \wedge \\ & \forall i, j \Box((i \neq j) \rightarrow \neg(\text{leave}_i \wedge \text{leave}_j)) \wedge \\ & \forall i, j \Box((i \neq j) \rightarrow \neg(\text{add}_i \wedge \text{add}_j)) \wedge \\ & \forall i, j \Box((i \neq j) \rightarrow \neg(\text{remove}_i \wedge \text{remove}_j))\end{aligned}$$

---

$$\begin{aligned}\tau_2 = & \Box(\text{Join} \rightarrow \bigcirc(\neg\text{Join} \mathcal{W} \text{Leave})) \wedge \\ & \Box(\text{Leave} \rightarrow \bigcirc(\neg\text{Leave} \mathcal{W} \text{Join})) \wedge \\ & \Box(\text{Add} \rightarrow \bigcirc(\neg\text{Add} \mathcal{W} \text{Remove})) \wedge \\ & \Box(\text{Remove} \rightarrow \bigcirc(\neg\text{Remove} \mathcal{W} \text{Add}))\end{aligned}$$

---

$$\tau_3 = \Box(\text{Leave} \rightarrow \blacklozenge\text{Join}) \wedge \Box(\text{Remove} \rightarrow \blacklozenge\text{Add})$$

# g-SIS Specification (Syntactic Correctness)

- Defines precisely when authorization holds
- A g-SIS specification is syntactically correct if
  - Stated in terms of user and object operations
  - Satisfies well-formedness constraints

$$\gamma = \forall u \in U. \forall o \in O. \Box(\text{Authz}(u, o, r) \leftrightarrow \psi(u, o)) \wedge \bigwedge_{0 \leq i \leq 3} \tau_i$$

specified using join, leave, add  
and remove (but not authz)

Well-formedness  
constraints

- A g-SIS specification is semantically correct if it satisfies following core properties

# Core g-SIS Properties

- Persistence
  - Authorization cannot change if no group event occurs

$$\varphi_0 = \Box(\text{Authz} \rightarrow (\text{Authz } \mathcal{W} (\text{Join} \vee \text{Leave} \vee \text{Add} \vee \text{Remove})))$$

$$\varphi_1 = \Box(\neg\text{Authz} \rightarrow (\neg\text{Authz } \mathcal{W} (\text{Join} \vee \text{Leave} \vee \text{Add} \vee \text{Remove})))$$

- Provenance
  - Authorization can begin to hold only after a simultaneous period of user and object membership

$$\varphi_2 = (\neg\text{Authz } \mathcal{W} (\text{Authz} \wedge (\neg\text{Leave } \mathcal{S} \text{Join}) \wedge (\neg\text{Remove } \mathcal{S} \text{Add})))$$



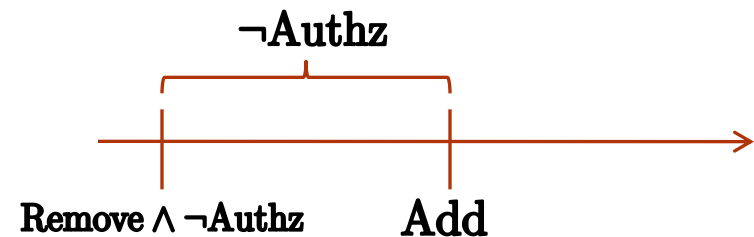
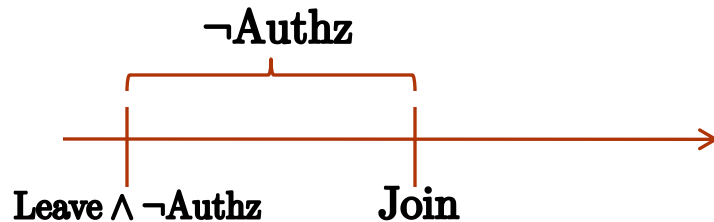
# Core g-SIS Properties (contd)

- Bounded Authorization

- Authorization cannot grow during non-membership periods

$$\varphi_3 = \Box((\text{Leave} \wedge \neg\text{Authz}) \rightarrow (\neg\text{Authz} \mathcal{W} \text{Join}))$$

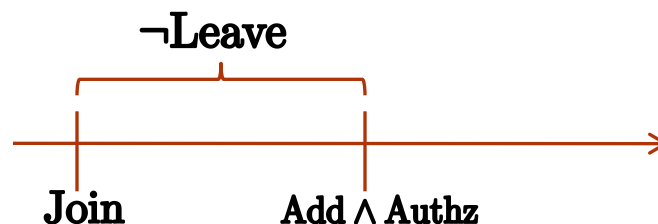
$$\varphi_4 = \Box((\text{Remove} \wedge \neg\text{Authz}) \rightarrow (\neg\text{Authz} \mathcal{W} \text{Add}))$$



- Availability

- After add, authorization should hold for all existing group users

$$\varphi_5 = \Box(\text{Join} \rightarrow ((\text{Add} \rightarrow \text{Authz}) \mathcal{W} \text{Leave}))$$



# g-SIS Specification (Semantic Correctness)

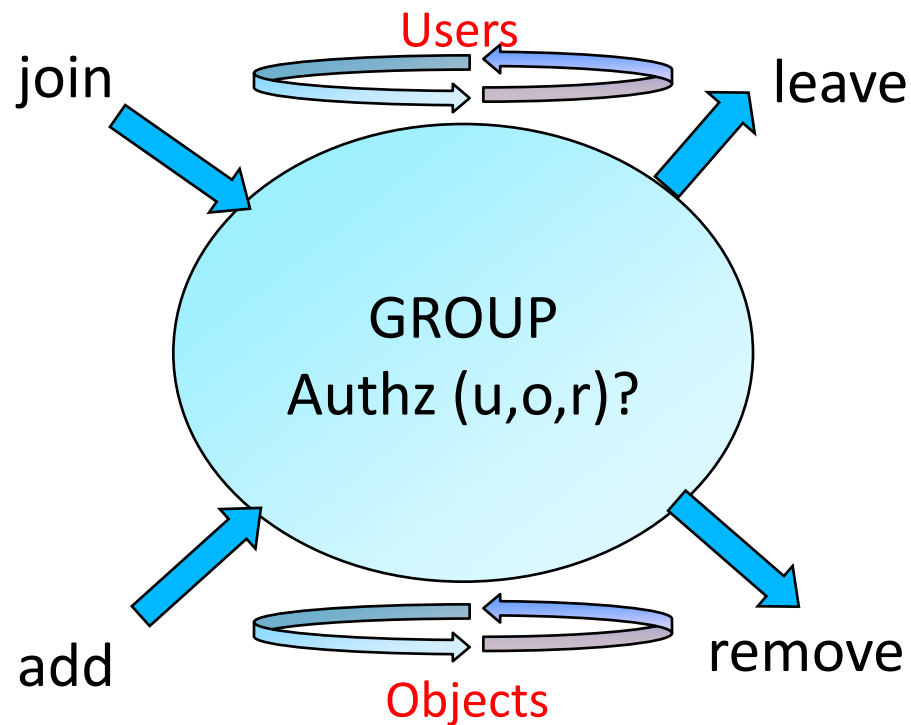
- Semantically correct if it satisfies the core g-SIS properties

$$\gamma \models \bigwedge_{0 \leq i \leq 5} \varphi_i$$

- Syntactic correctness

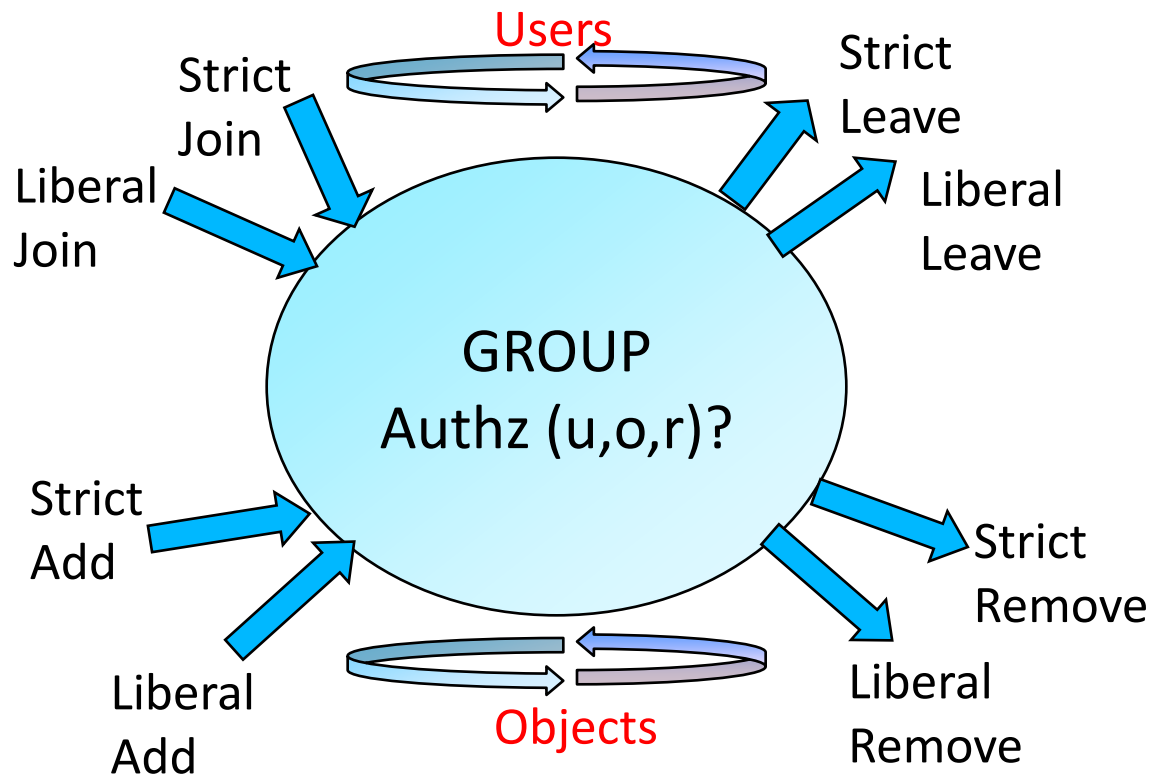
$$\gamma = \forall u \in U. \forall o \in O. \Box(\text{Authz}(u, o, r) \leftrightarrow \psi(u, o)) \wedge \bigwedge_{0 \leq i \leq 3} \tau_i$$

# g-SIS Operation Semantics



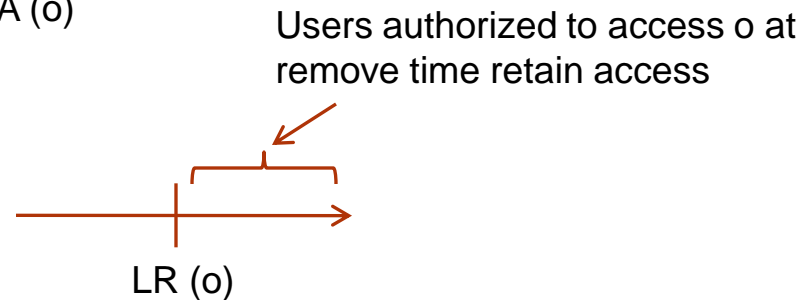
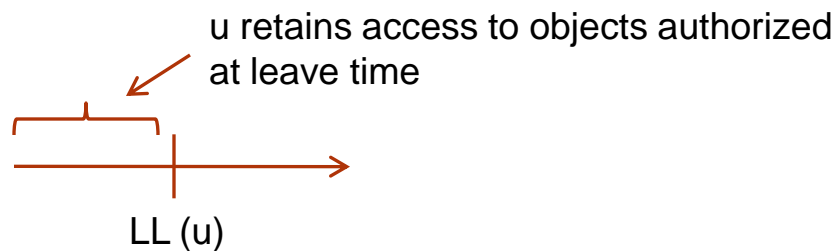
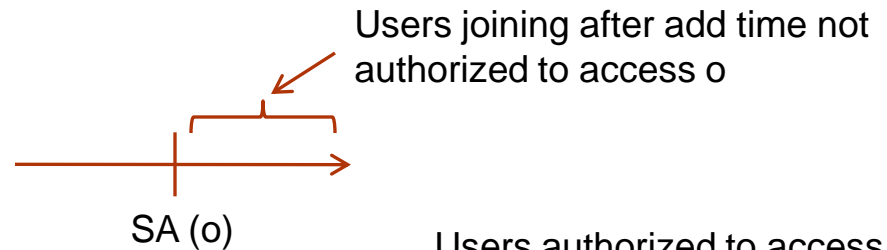
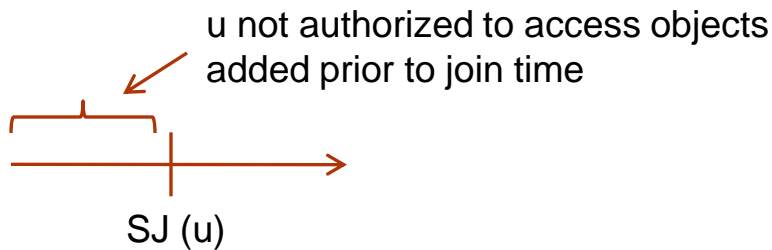


# g-SIS Operation Semantics



# Group Operation Semantics

- Membership semantics
  - Considers authorizations enabled by Join and Add
    - And those disabled by Leave and Remove
  - Strict Vs Liberal operations
    - User operations (SJ, LJ, SL, LL)
    - Object operations (SA, LA, SR, LR)



# Group Operation Semantics (contd)

- Membership Renewal Semantics
  - Considers authorizations from past membership period(s)
- Lossless Vs Lossy Join
  - Lossless: Authorizations from past membership period not lost
  - Lossy: Some authorizations lost at rejoin time
- Restorative Vs Non-Restorative Join
  - Restorative: Authorizations from past membership restored
  - Non-Restorative: Past authorizations not restored at rejoin time
- Gainless Vs Gainful Leave
- Restorative Vs Non-Restorative Leave

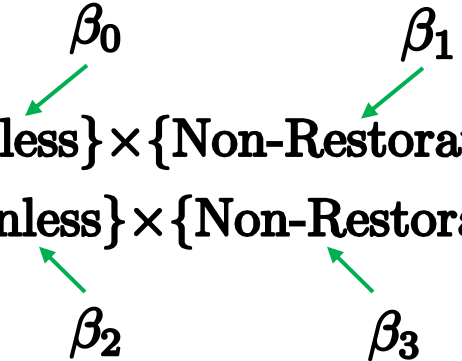
## LTL spec for Membership and Membership Renewal Properties (contd)

Operation	Explanation	Property
Strict Join (SJ)	Only objects added after join time can be accessed	$\alpha_0 = \Box(\text{Authz} \rightarrow \blacklozenge(\text{Add} \wedge (\neg \text{Leave } \mathcal{S} \text{ join}_i)))$
Liberal Join (LJ)	Can access objects added before and after join time	There exists a well-formed trace that does not satisfy $\alpha_0$
Strict Leave (SL)	Lose access to all objects on leave	$\alpha_1 = \Box(\text{Authz} \rightarrow (\neg \text{leave}_i \mathcal{S} \text{ Join}))$
Liberal Leave (LL)	Retain access to objects authorized before leave time	There exists a well-formed trace that does not satisfy $\alpha_1$
Strict Add (SA)	Only users who joined prior to add time can access	$\alpha_2 = \Box(\text{add}_i \rightarrow (\neg \blacklozenge \text{Join} \rightarrow (\neg \text{Authz } \mathcal{W} \text{ Add})))$
Liberal Add (LA)	Users who joined before or after add time may access	There exists a well-formed trace that does not satisfy $\alpha_2$
Strict Remove (SR)	All users lose access on remove	$\alpha_3 = \Box(\text{remove}_i \rightarrow (\neg \text{Authz } \mathcal{W} \text{ Add}))$
Liberal Remove (LR)	Users who had access at remove time retain access	There exists a well-formed trace that does not satisfy $\alpha_3$

Operation	Explanation	Property
Lossless Join	Authorizations prior to join time is not lost	$\beta_0 = \Box(((\text{Join} \wedge \neg \text{Remove} \wedge \ominus \text{Authz}) \rightarrow \text{Authz}))$
Lossy Join	Authorizations from prior to join may be lost	There exists a well-formed trace that does not satisfy $\beta_0$
Non-Restorative Join	Authorizations from past membership periods not explicitly restored	$\rho_1 = (\text{join}_i(u1) \wedge \text{join}_i(u2) \wedge$ $\text{Authz}(u1, o, r) \wedge \neg \text{Authz}(u2, o, r))$ $\rho_2 = \ominus (\text{Authz}(u1, o, r) \wedge \neg \text{Authz}(u2, o, r))$ $\beta_1 = \forall i \Box(\rho_1 \rightarrow \rho_2)$
Restorative Join	Authorizations from past membership may be restored	There exists a well-formed trace that does not satisfy $\beta_1$
Gainless Leave	Authorizations that never held during most recent membership period cannot be obtained	$\beta_2 = \Box(((\text{Leave} \wedge (\neg \text{Join } \mathcal{U} (\text{Authz} \wedge \neg \text{Join}))) \rightarrow$ $\ominus ((\neg \text{Authz} \wedge \neg \text{Join}) \mathcal{S} (\text{Authz} \wedge (\neg \text{Join } \mathcal{S} \text{ Join}))))$
Gainful Leave	New authorizations may be granted at Leave time	There exists a well-formed trace that does not satisfy $\beta_2$
Non-Restorative Leave	Authorizations that the user had prior to joining the group are not explicitly restored	$\beta_3 = \Box(\text{Leave} \wedge \text{Authz} \rightarrow \ominus \text{Authz})$
Restorative Leave	Authorizations from prior to join time may be restored	There exists a well-formed trace that does not satisfy $\beta_3$

# The $\pi$ -System g-SIS Specification

- $\pi$ -system is a g-SIS specification
  - Allows all membership ops (Strict and Liberal user/object ops)
  - Allows only selected membership renewal ops
    - Lossless and Non-Restorative Join
    - Gainless and Non-Restorative Leave

$$\begin{aligned}\forall i. \text{Type}(\text{join}_i) &\in \{\text{SJ}, \text{LJ}\} \times \{\text{Lossless}\} \times \{\text{Non-Restorative}\} \\ \forall i. \text{Type}(\text{leave}_i) &\in \{\text{SL}, \text{LL}\} \times \{\text{Gainless}\} \times \{\text{Non-Restorative}\} \\ \forall i. \text{Type}(\text{add}_i) &\in \{\text{SA}, \text{LA}\} \\ \forall i. \text{Type}(\text{remove}_i) &\in \{\text{SR}, \text{LR}\}\end{aligned}$$


# The $\pi$ -System g-SIS Specification (contd)

$\pi$ -system g-SIS Specification:

$$\pi = \square(\text{Authz} \leftrightarrow \lambda_1 \vee \lambda_2) \wedge \bigwedge_{0 \leq j \leq 3} \tau_j$$

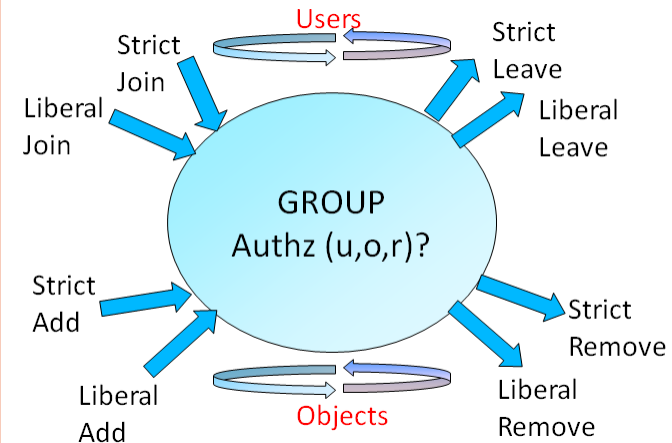
$$\lambda_1 = ((\neg \text{SL} \wedge \neg \text{SR}) \mathcal{S} ((\text{SA} \vee \text{LA}) \wedge ((\neg \text{LL} \wedge \neg \text{SL}) \mathcal{S} (\text{SJ} \vee \text{LJ}))))$$

$$\lambda_2 = ((\neg \text{SL} \wedge \neg \text{SR}) \mathcal{S} (\text{LJ} \wedge ((\neg \text{SR} \wedge \neg \text{LR}) \mathcal{S} \text{LA})))$$

Well-formed traces

Add after Join

Add before Join



Entailment Theorem: The  $\pi$ -system entails the Core g-SIS properties

$$\pi \models \left( \bigwedge_{0 \leq q \leq 5} \varphi_q \wedge \bigwedge_{0 \leq r \leq 3} \beta_r \right)$$

Core properties

Membership Renewal Properties

# Verification Using Model Checker

- Model allows join, leave, add and remove to occur concurrently, non-deterministically and in any order

$$\pi \rightarrow \bigwedge_{0 \leq q \leq 5} \varphi_q \wedge \bigwedge_{0 \leq r \leq 3} \beta_r$$

- The above implication is used as the LTLSPEC
- The model checker generates a counter-example if the specification is false
- Used the open-source NuSMV model checker

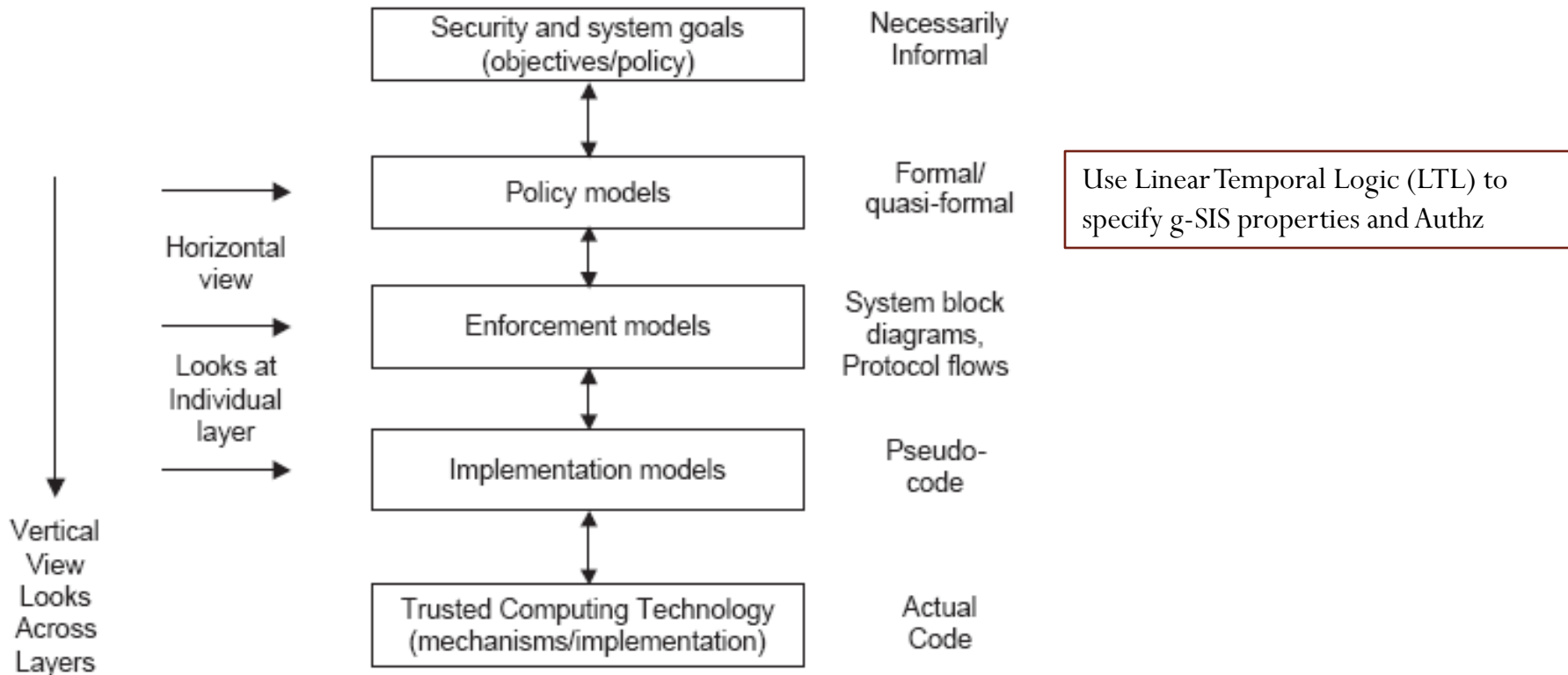
# Conclusion

- Group-Centric Vs Dissemination-Centric SIS
- Core g-SIS properties
- Various group operation semantics
- g-SIS specification using LTL
- Entailment theorem
- Ongoing work
  - Read-Write model with versioning
  - Multiple groups



# Backup

# PEI Framework for Secure Systems Design



Formal Specification using LTL allows:

1. Precise, Concise expression of state sequence properties
2. Enables automated verification of properties

# Linear Temporal Logic (summary)

- Next  $p$  ( $\bigcirc p$ )
    - Formula  $p$  holds in the next state
  - Henceforth  $p$  ( $\square p$ )
    - Starting from current state,  $p$  will continuously hold in all the future states
  - $p$  until  $q$  ( $p \mathcal{U} q$ )
    - $q$  will occur sometime in the future and  $p$  will hold at least until the first occurrence of  $q$
  - $p$  unless  $q$  ( $p \mathcal{W} q$ )
    - $p$  holds either until the next occurrence of  $q$  or if  $q$  never occurs, it holds throughout
- 
- Previous  $p$  ( $\ominus p$ )
    - Formula  $p$  held in the previous state
  - Once  $p$  ( $\blacklozenge p$ )
    - Formula  $p$  held at least once in the past
  - $p$  since  $q$  ( $p \mathcal{S} q$ )
    - $q$  happened in the past and  $p$  held continuously from the position following the last occurrence of  $q$  to the present