A Hybrid Enforcement Model for Group-Centric Secure Information Sharing (g-SIS)

Ravi Sandhu
Executive Director and Endowed Professor
Institute for Cyber Security
University of Texas at San Antonio

August 2009

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

Co-authored with
Ram Krishnan, PhD Candidate, George Mason University

Outline

- Motivation for g-SIS
- g-SIS Enforcement Architecture
- Micro vs Super-distribution in g-SIS
- Hybrid g-SIS Architecture
- Comparison
- Conclusion

Secure Information Sharing (SIS)

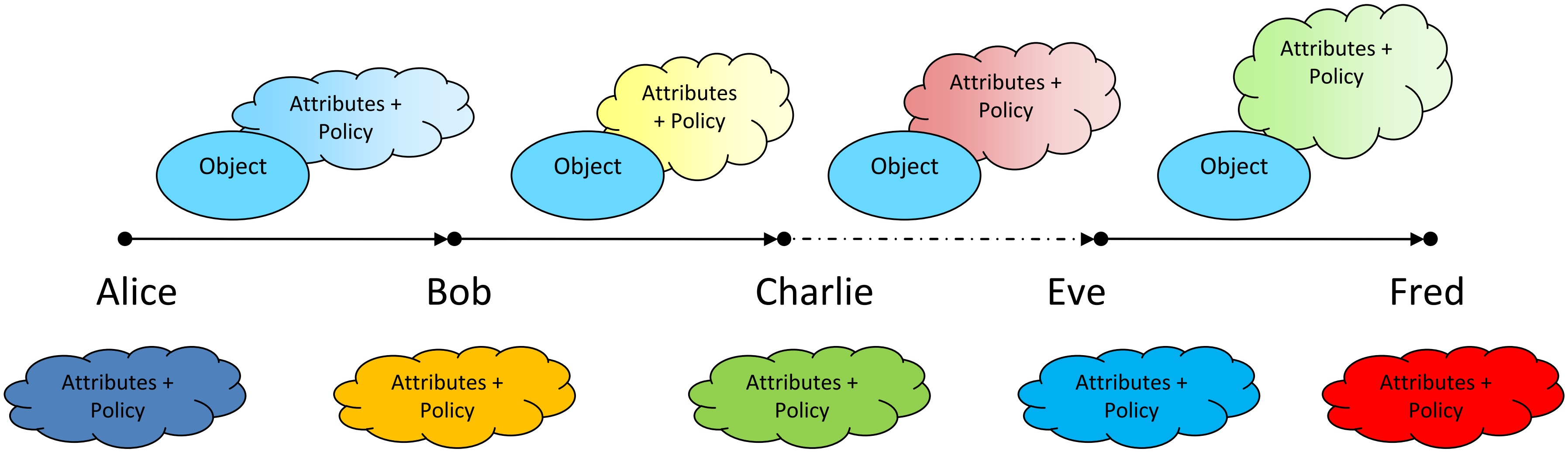
- SIS: Share *but* protect
- Traditional models capture important SIS aspects BUT have serious shortcomings
  - Discretionary Access Control (owner control)
    - Too fine-grained, lacks copy/usage control
  - Lattice-Based Access Control (information flow)
    - Too rigid, coarse-grained and binary
  - Role-Based Access Control (effective administration)

    Attribute-Based Access Control (implicit/automated administration)

    Usage Control (mutable attributes, continuous enforcement, obligations)
    - Do not directly address information sharing
- Primary issues
  - Copy/usage control
  - Manageability
  - Purpose

Dissemination-Centric Sharing

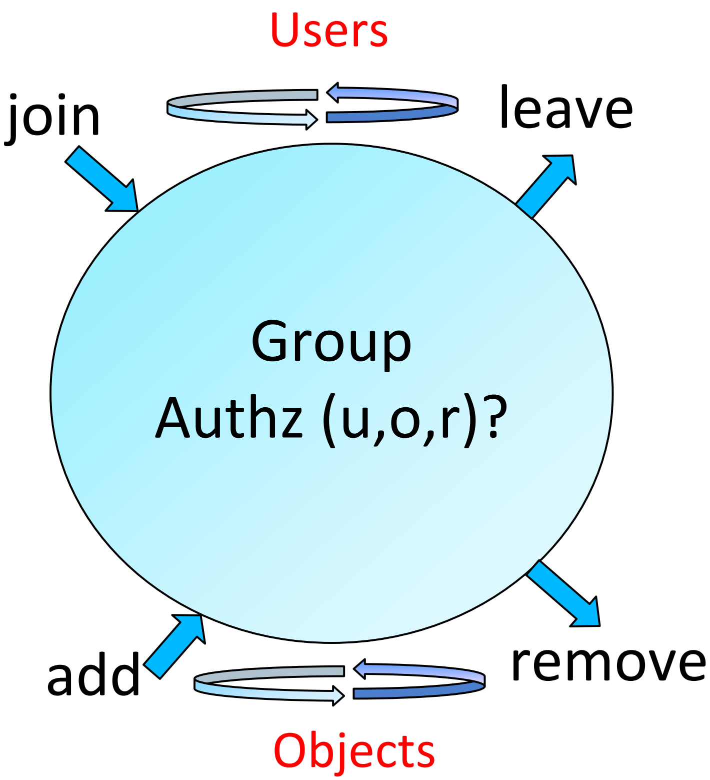
- Extensive research in the last two decades
  - ORCON, DRM, ERM, XrML, ODRL, etc.
- Copy/usage control: major attention
- Manageability and purpose: hardly any attention



Dissemination Chain with Sticky Policies on Objects

Group-Centric Sharing (g-SIS)

- Brings users & objects together in a group
  - Focus on manageability and purpose
  - Co-exists with dissemination-centric
  - Two metaphors
    - Secure Meeting Room (E.g. Program committee)
    - Subscription Model (E.g. Secure multicast)
- Operational aspects
  - Group characteristics
    - E.g. What core properties are required of all groups?
  - Group operation semantics
    - E.g. What precisely is authorized by join, add, etc.?
  - Is there additional structure within the group
    - E.g. Security levels, roles, sub-groups?
- Administrative aspects
  - E.g. Who authorizes join, add, etc.?
- Multiple groups
  - Inter-group relationship



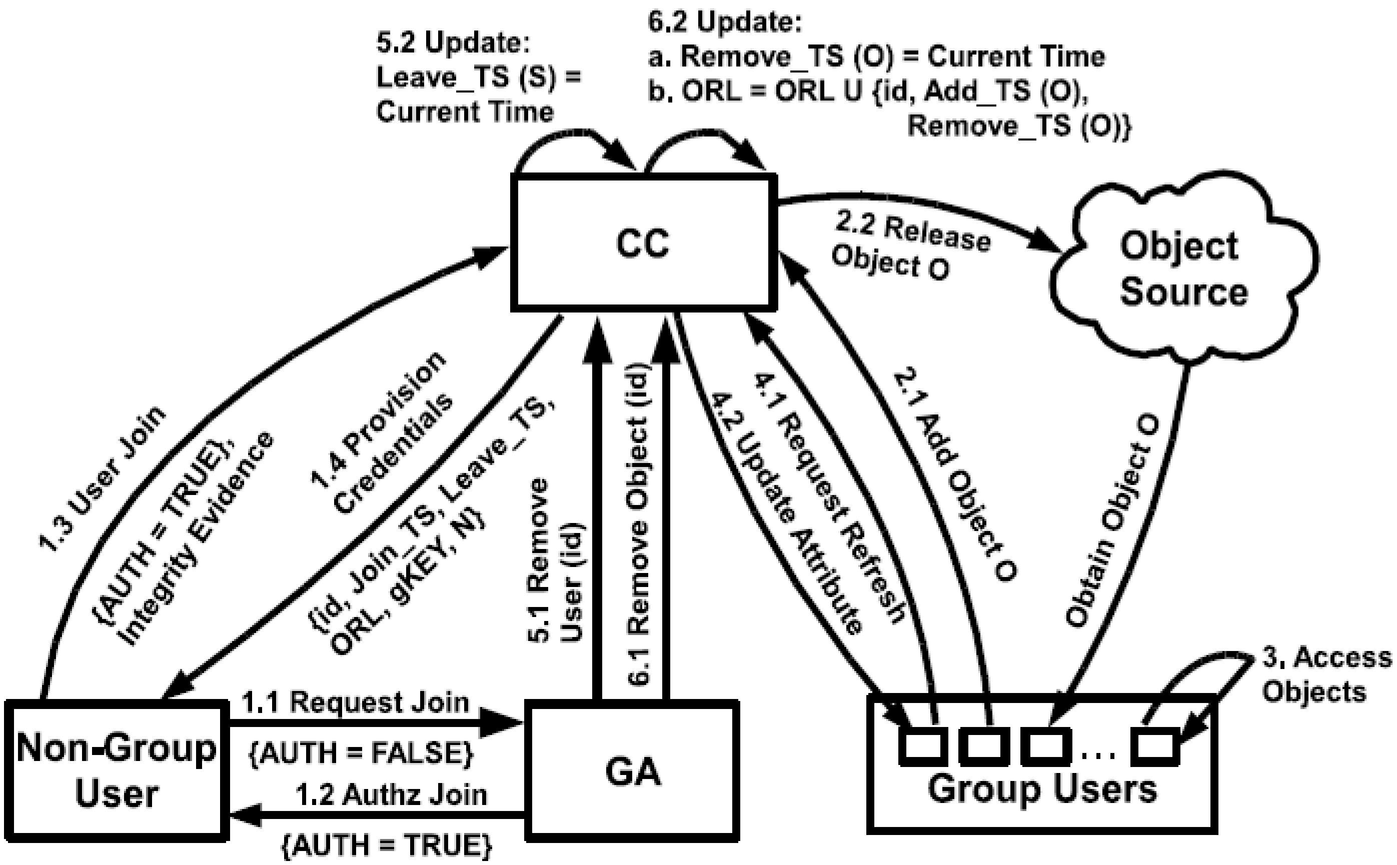
Roles vs Groups in SIS

- Roles
  - Users get same set of privileges on role assignment
  - Temporal aspects of roles have been studied
    - E.g., when can a role can be activated, what pre-requisite roles need to be activated first
- Groups
  - Privileges may differ with time of join, leave, etc.
  - Groups are a unit of purpose-oriented sharing
  - Inter-group relationship differ from that of roles

g-SIS Architecture

Key Features:

- Trusted Clients
- Offline Access

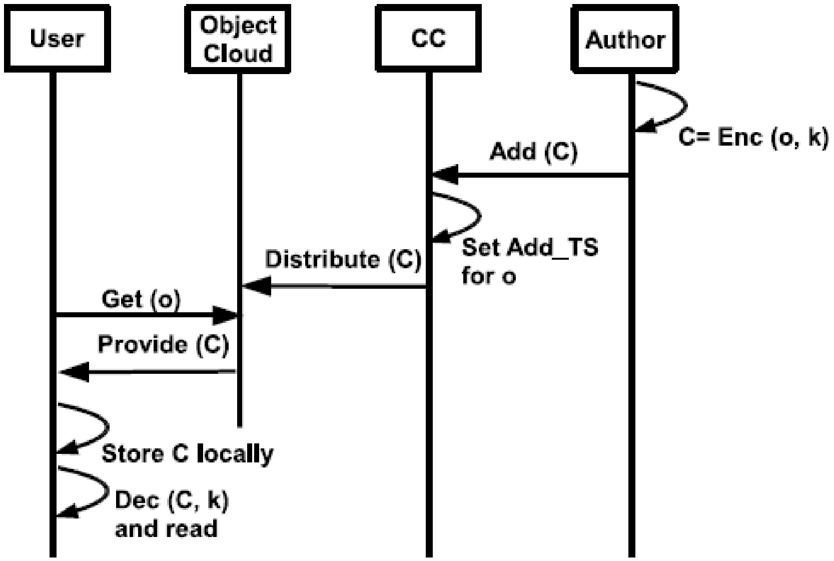


User Attributes: {id, Join_TS, Leave_TS, ORL, gKey, usageCount}

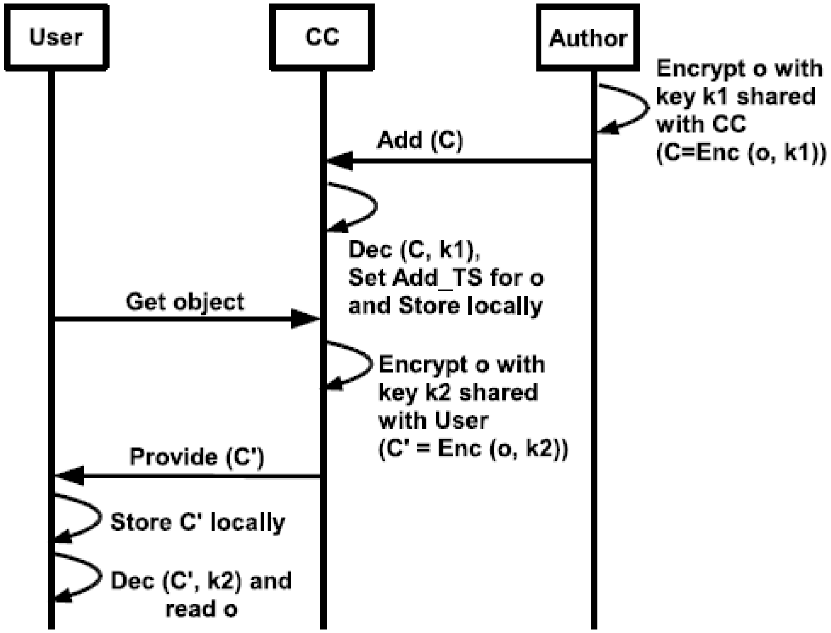
Object Attributes: {id, Add_TS}

Policy: $\mathrm{Authz}(u, o, read) \rightarrow o \notin \mathrm{ORL}(u) \wedge \mathrm{Leave\_TS}(u) = \mathrm{NULL}$
$\wedge\, \mathrm{Join\_TS}(u) \leq \mathrm{Add\_TS}(o)$

Super vs Micro-distribution in g-SIS



Super-Distribution (SD)


User
CC
Author
Encrypt o with key k1 shared with CC (C=Enc (o, k1))
Add (C)
Dec (C, k1), Set Add_TS for o and Store locally
Get object
Encrypt o with key k2 shared with User (C' = Enc (o, k2))
Provide (C')
Store C' locally
Dec (C', k2) and read o

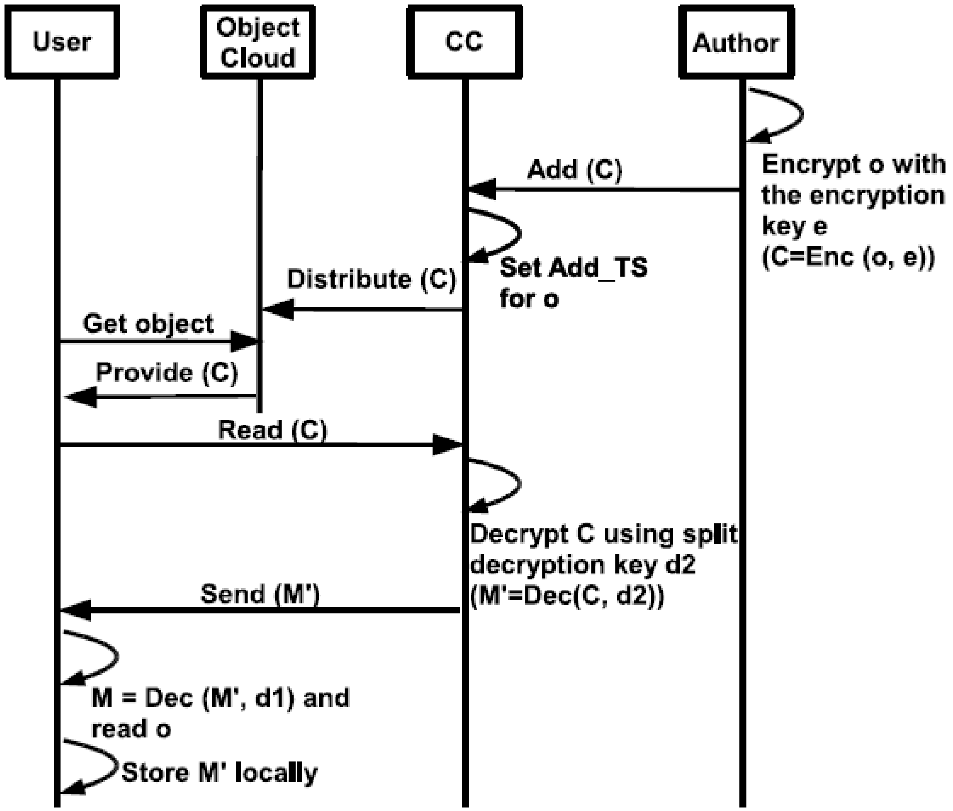
Micro-Distribution (MD)

- Scalability/Performance
  - SD: Encrypt once, access where authorized
  - MD: Custom encrypt for each user on initial access
- Assurance/Recourse
  - SD: Compromise one client, compromise group key
  - MD: Compromise of one client contained to objects on that client

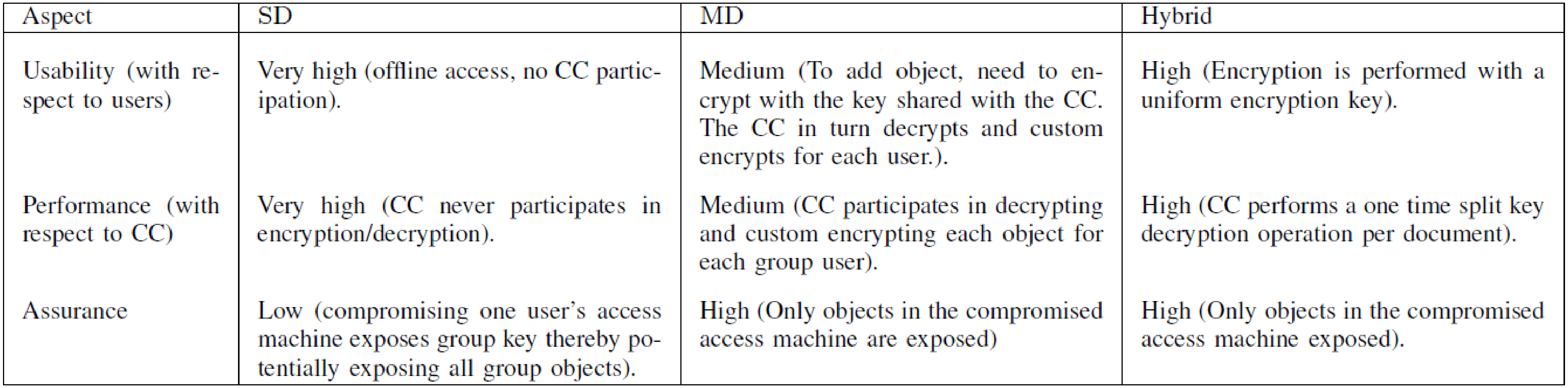
Hybrid Approach

- Split-key RSA
  - Decryption key split into two parts
  - Different split for each group user
  - One split held by CC, other split shared with user

$$
\begin{aligned}
e * d &= 1 \; mod \; \varphi(n) \\
d1 * d2 &= d \; mod \; \varphi(n) \\
C &= M^{e} \; mod \; n \\
{(M)^{d1}}^{d2} \; mod \; n &= \\
{(M)^{d2}}^{d1} \; mod \; n &= \\
(M)^{d1*d2} \; mod \; n &= \\
M^{d} \; mod \; n &
\end{aligned}
$$


User
Object Cloud
CC
Author
Encrypt o with the encryption key e (C=Enc (o, e))
Add (C)
Set Add_TS for o
Distribute (C)
Get object
Provide (C)
Read (C)
Decrypt C using split decryption key d2 (M'=Dec(C, d2))
Send (M')
M = Dec (M', d1) and read o
Store M' locally

Comparison



Aspect	SD	MD	Hybrid
Usability (with respect to users)	Very high (offline access, no CC participation).	Medium (To add object, need to encrypt with the key shared with the CC. The CC in turn decrypts and custom encrypts for each user.).	High (Encryption is performed with a uniform encryption key).
Performance (with respect to CC)	Very high (CC never participates in encryption/decryption).	Medium (CC participates in decrypting and custom encrypting each object for each group user).	High (CC performs a one time split key decryption operation per document).
Assurance	Low (compromising one user's access machine exposes group key thereby potentially exposing all group objects).	High (Only objects in the compromised access machine are exposed)	High (Only objects in the compromised access machine exposed).

SD – Super-Distribution
MD – Micro-Distribution

Conclusion

- Group-Centric vs Dissemination-Centric Sharing
- g-SIS Enforcement Architecture
  - Super-Distribution (SD) vs Micro-Distribution (MD)
  - Hybrid approach using public key cryptography with split private keys
- Hybrid approach offers a mix of
  - Usability and performance advantages of Super-Distribution
  - Better compromise containment of Micro-Distribution