Group-Centric Models for Secure and Agile Information Sharing

Ravi Sandhu^{1,2}, Ram Krishnan^{1},
Jianwei Niu^{1,2}, and William H. Winsborough^{1,2}

^{1} Institute for Cyber Security
^{2} Department of Computer Science
University of Texas at San Antonio
{ravi.sandhu,ram.krishnan}@utsa.edu,
niu@cs.utsa.edu, wwinsborough@acm.org

Abstract. To share information and retain control (share-but-protect) is a classic cyber security problem for which effective solutions continue to be elusive. Where the patterns of sharing are well defined and slow to change it is reasonable to apply the traditional access control models of lattice-based, role-based and attribute-based access control, along with discretionary authorization for further fine-grained control as required. Proprietary and standard rights markup languages have been developed to control what a legitimate recipient can do with the received information including control over its further discretionary dissemination. This dissemination-centric approach offers considerable flexibility in terms of controlling a particular information object with respect to already defined attributes of users, subjects and objects. However, it has many of the same or similar problems that discretionary access control manifests relative to role-based access control. In particular specifying information sharing patterns beyond those supported by currently defined authorization attributes is cumbersome or infeasible. Recently a novel mode of information sharing called group-centric was introduced by these authors. Group-centric secure information sharing (g-SIS) is designed to be agile and accommodate ad hoc patterns of information sharing. In this paper we review g-SIS models, discuss their relationship with traditional access control models and demonstrate their agility relative to these.

Keywords: DAC, Groups, LBAC, MAC, RBAC, Secure Information Sharing.

1 Introduction

The need to *share but protect* is one of the oldest and most challenging problems for trustworthy computing. Saltzer-Schroeder [1] identified the desirability and difficulty of maintaining “some control over the user of the information even after it has been released.” The ensuing three and half decades have further compounded the technical difficulties to the point where one may ask if it is even reasonable to seek solutions. The analog hole [2] wherein content is captured at the point it is rendered into human perceptible form and converted back into unprotected digital form highlights the intrinsic limits. At the same time our increasingly information-rich and information-dependent

I. Kotenko and V. Skormin (Eds.): MMM-ACNS 2010, LNCS 6258, pp. 55–69, 2010.

society needs to exploit *secure information sharing* (SIS) to fully benefit from the productivity, social and national security benefits of the ongoing cyber revolution.

SIS presents two major research challenges. The *containment challenge* is to ensure that protected information is accessible on the recipient's computer only as permitted by policy, including inability to make unprotected or less-protected copies. The latter has inherent limits such as the analog hole. Containment requires a trusted computing base on the recipient's machine and a mix of cryptography and access control, with the degree of assurance correlated with tamper-resistance. There is a rich literature on containment including the currently dominant TCG approach [3]. While high assurance is elusive and may remain so, there is consensus that low to medium assurance is within state-of-the-art.

In this paper, we assume that adequate assurance for containment is available commensurate with the application. We focus on the *policy challenge* of specifying, analyzing and enforcing SIS policies assuming adequate containment. A basic premise is that this requires new access control models that can integrate and go beyond earlier ones, have intuitive grounding and rigorous mathematical foundations, are usable by the ordinary citizen and enforceable in distributed systems. The paper will build upon a novel approach called Group-centric Secure Information Sharing (g-SIS) recently introduced by the authors [4,5,6]. Another basic premise is that the policy challenge in specifying and analyzing the intrinsic application policy should be clearly separated from enforcement policy issues that arise due to the realities and practicalities of a distributed system. Following [7,8,9] we call these respectively P-layer (for application policy) and E-layer (for enforcement policy) concerns. These premises are elaborated below.

Although many access control models have been published and analyzed, only three have received meaningful practical traction [10]. Discretionary access control (DAC) [11,12,13] enforces controls on sharing information at the discretion of the "owner" of the information but fails containment completely by allowing unprotected copies to be made. (Originator Control or ORCON [14,15,16,17] attaches policies from the original to the copies to fix this defect, but does not directly address the policy challenge.) Lattice-based access control (LBAC) [11,18,19,20] restricts information to flow in one direction in a lattice of security labels. Copies inherit the least upper bound of labels from the originals and remain contained. Information sharing in LBAC is essentially preordained in that information is either not shared or shared with everyone who has a sufficiently strong clearance. Any deviation from this pattern requires creation of a new label, which is not supported in existing LBAC models and breaks their existing mathematical foundations. Role-based access control (RBAC) [21,22] is designed to facilitate assigning permissions based on job function and such considerations. Although RBAC can be configured to enforce DAC and LBAC [23] it is not designed with information sharing in mind, so it does not directly address the containment or policy challenges. (Attribute-based access control models such as UCON [24] and XACML [25] use general attributes in addition to roles and security labels, but likewise do not directly address containment or policy.) This bears out the premise that new access control models are needed for SIS. At the same time these successful classic models embody intuitions and principles that are likely to be vital to a comprehensive solution.

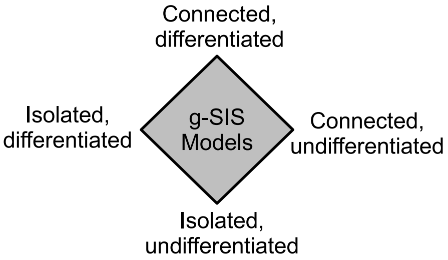

Connected, differentiated
Isolated, differentiated
g-SIS Models
Connected, undifferentiated
Isolated, undifferentiated

Fig. 1. A family of g-SIS models

The premise of sharply separating P- and E-layers builds on the much practised policy/mechanism separation principle first articulated in HYDRA [26]. P-layer specifications express a policy that is ideal in the sense that it ignores issues such as distributed authorization state, network latency, caching, and requirements for off-line use. E-layer specifications define authorization decisions that approximate those given by the ideal policy in a manner that provides the desired application-dependent balance between resource availability and timely propagation of authorization-state changes. They also include additional entities such as trusted authorization/revocation servers which are abstracted out at the P-layer.

This paper primarily focusses on P-layer aspects of g-SIS. In g-SIS, users and information come together in a group to facilitate sharing. Users gain access to group information by virtue of membership. Likewise information is made available to members by adding it to the group. Constituting a group as the unit of SIS provides many of the same benefits of using roles versus individual users for permission distribution. Two useful metaphors for a g-SIS group are a subscription service and a secure meeting room. Subscription disseminates information to subscribers who participate in blogs and forums. A meeting room brings people together to share information available in the room. The times at which users join and leave and at which objects are added and removed affect user authorizations both during and after periods of group membership. For example, in the much studied secure multicast problem [27] new members joining the group cannot access content added prior to joining (backward secrecy) and members leaving the group cannot access new content thereafter (forward secrecy). The requirements of a committee meeting room could allow members access to older information once they join (no backward secrecy). These metaphors further indicate the need for multiple groups. In the simplest case we can have multiple groups that are *isolated* or *independent* in that membership in one group has no impact on what a user can do in another group, whereas with *coupled* or *connected* groups such impact can occur. A theory of g-SIS thus needs to model and enable specification of such temporal and coupling interactions. Looking within a group we can distinguish *undifferentiated* versus *differentiated* groups. In an undifferentiated group user authorizations are undifferentiated once users are admitted into the group. Specifically, authorizations do not depend on attributes other than group membership (and associated temporal relations between users and objects as discussed above earlier). Combining these two characteristics of groups we have four possible cases shown in figure 1 for g-SIS models. In this figure the lowest class (isolated, undifferentiated) is included in all the higher classes; the highest class (connected, differentiated) includes all the others; and the two classes in the middle (isolated, differentiated) and (connected, undifferentiated) are incomparable in this respect.

Our prior work [5] primarily focussed on the isolated group model. In this paper, we outline our vision on building the connected, undifferentiated group model and compare it with classic access control models such as LBAC, Domain and Type Enforcement [28] and RBAC. We show that our proposed connected, undifferentiated group model can express such policies and conveniently handle more dynamic information sharing scenarios. The remainder of this paper is organized as follows. In section 2, we briefly review the isolated group model. In section 3, we discuss candidate inter-group relationships for the connected group model. We also discuss constructions of LBAC [20] and a read-write RBAC_0 model [22] and demonstrate the agility of the connected group model in relation to these. We conclude in section 4.

2 Background

Group-Centric models for secure information sharing (g-SIS) have been recently introduced [4,5,6]. In this paper we focus entirely on undifferentiated groups. There are then two classes of g-SIS models: isolated, undifferentiated (g-SIS^{i}) and connected, undifferentiated (g-SIS^{c}). For convenience we will henceforth drop explicit mention of undifferentiated and simply call these two classes isolated and connected respectively. In g-SIS^{i}, groups are isolated in the sense that they do not directly interact with each other. For instance, a user's membership in one group has no implication on her authorizations in other groups. Our prior work [4,5,6] focusses primarily on isolated g-SIS models. In g-SIS^{c}, groups may be related. For instance, user's membership in one group may be contingent upon her membership in another group or groups could be hierarchical where users in one group may dominate another group. In this section, we briefly review the core aspects of isolated g-SIS models. In the subsequent sections, we discuss candidate relationships in the connected group models.

In g-SIS^{i}, a group is established, for instance, between two or more organizations for a specific purpose. Users from these organizations may join, leave and possibly re-join the group. Similarly, objects from participating organizations may be added, removed and possibly re-added. Users in the group may read and write such group objects and potentially create new objects in the group. Such new objects typically represent intellectual property created as a result of collaboration between participating organizations. In such scenarios, authorizations in the group may depend upon various aspects such as the time at which a user joined and the time at which the object was added. Specifically, there is a requirement of simultaneous membership of a user and an object in order to be able to read/write the object.

g-SIS^{i} recognizes a range of group policies. For instance, in some scenarios, users may be authorized to access certain objects even after leaving the group. In another, a joining user may access objects added prior to her join time. Two metaphors highlight such scenarios: secure meeting room and subscription service. For the secure meeting room metaphor, consider a program committee meeting where participants discuss in a room. Suppose, Alice is a member whose paper is currently discussed. Typically, Alice steps out of the room for a brief period. During this period, Alice may retain access to discussions that occurred prior to the time at which she left the room. Further, on re-joining the room at a later period, her access to discussions resumes (except those that occurred during her period of absence). In another scenario (where Alice to had to step out of the room for reasons other than conflict-of-interest), discussions that occurred during Alice's absence may be recorded in a white board and she may access them on re-join.

For the subscription service metaphor, consider a secure multicast network which typically has a notion of backward and forward secrecy. When a node joins the multicast network, it cannot access data distributed on the network prior to join time (backward secrecy). When a node leaves the network, it cannot access data shared between other nodes after leave time (forward secrecy).

In general, there could be numerous variations of such policies in g-SIS^i. A g-SIS^i *specification* characterizes the precise conditions under which a user is authorized to perform a certain action (such read and write) on an object. All g-SIS^i specifications are required to satisfy a set of *core properties*. The core properties specify under what conditions it is appropriate for a specification to hold in the g-SIS^i model. We informally discuss these properties below (see [5] for a formal treatment).

Persistence Properties: This class of properties specifies that authorization may not change unless some authorization changing event occurs. In g-SIS^i, authorization changing events include a user joining and leaving a group and an object being added and removed from a group. Authorization (or Revocation) persistence property states that if a user is authorized (or not authorized) to access an object in a group, she will remain so unless one of the authorization changing event occurs.

Authorization Provenance: This class of properties is concerned about when authorization may begin to hold. As mentioned earlier, in certain scenarios, it is possible that a user may be able to access a group object even after leaving the group. (For instance, after the subscription ends, the user may retain access to articles that she had paid for.) This property states that a user's authorization to access an object may begin to hold for the *first time* only after a simultaneous period of group membership between the user and the object in question. Note that subsequent times at which the same authorization holds have no such requirement. Thus it is possible to construct a valid g-SIS^i specification in which after an initial overlapping period of user and object membership, the user may continue to remain authorized for that object even after leaving the group (or even after the object is removed from the group).

Bounded Authorization: This class of properties is concerned about what authorizations are allowed to hold during the non-membership periods of users/objects. For users, the property states that the set of objects that a user is authorized to access after she leaves the group cannot increase after leave time. (Note that she may lose access to such objects after leave time but she cannot gain access to new objects after leaving the group.) Similarly, for objects, the property states that the set of users authorized to access an object after it is removed from the group cannot increase after remove time.

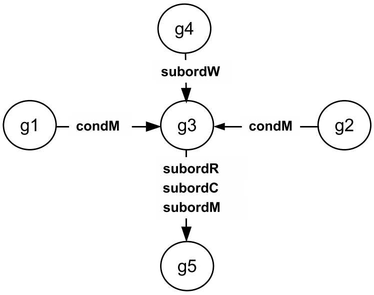


Fig. 2. A snapshot of relationships between various groups

In [5], we characterized a variety of useful authorization semantics for user join and leave operations and object add and remove operations. For instance, a *Strict* join to a group restricts a user's access to objects added to the group after join join time while a *Liberal* join allows access to all objects. We also developed a family of g-SIS^{i} specifications based on such authorization semantics and showed that they satisfy the core properties. In our follow on work, we have also shown that the core properties are logically consistent and mutually independent. We have further considered additional core properties in light of versioning support for object write. Here each object is composed of a growing set of versions and any specific version may be written to create a new version. Further, the core properties accommodate additional authorization changing operations such as update and object create.

3 Connected Group g-SIS Models

In this section, we introduce a connected g-SIS model (g-SIS^{c}) where groups are connected by some type of relationship. Before we discuss these relationships, it is important we distinguish the notion of user from that of a subject in access control. Typically, user a representation of a human being in the system (e.g. user id) and subjects represent processes (e.g. a word processing program) that a user may create to carry out various tasks. A user is typically trusted, within limits, in the system while a subject is not. For instance, a subject may be a trojan horse performing some hidden malicious activities such as a word processing program uploading contents to a remote server. Thus a user may create a subject with restricted privileges for containment purposes.

3.1 Inter-group Relationship Semantics in g-SIS^{c}

We discuss a few candidate inter-group relationships for the g-SIS^{c} model below:

1. Conditional Membership (condM): A conditional membership relation between two groups specifies that a users membership in one group is contingent upon her membership in another group. We define conditional membership relation to be reflexive. Transitivity and symmetry must be explicitly defined if required. Conditional membership requirements are common in collaboration scenarios. For instance, consider a collaboration group g3 established between two organizations

represented by groups g1 and g2 respectively (see figure 2). It is typical that every user in g3 is required to be a member of either g1 or g2. The definitions condM(g3,g1) and condM(g3,g2) can easily specify this requirement. Note that conditional membership is a relation defined between groups for users. It does not specify any direct requirement on subjects.

2. Subordination: The subordination relations, in general, characterize the notion of one entity dominating another. In g-SIS^c, we define a number of subordination relations where one group dominates another in different ways. Again, all of these relationships are reflexive by definition. Transitivity and symmetry must be explicitly defined if required.
   – Create Subordination (subordC): A subordC(g3,g5) definition states that users in group g3 may create subjects in group g5.
   – Read Subordination (subordR): A subordR(g3,g5) definition states that subjects in group g3 may read objects in group g5.
   – Write Subordination (subordW): A subordW(g4,g3) definition states that subjects in group g4 may write to objects in group g3.
   – Move Subordination (subordM): A subordM(g3,g5) definition states that subjects in group g3 may move to group g5. After moving to g5, the subject no longer resides in g3 which may result in losing access to objects in g3.

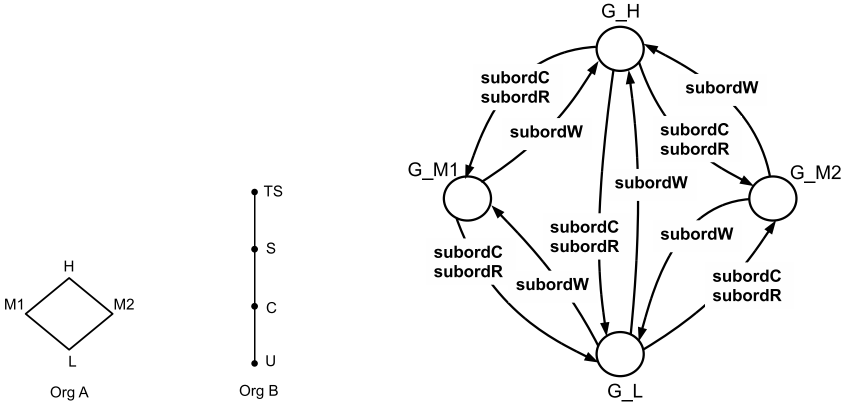
   Evidently, these subordination relations allow users in one group to read and write objects in another related group by means of their subjects.
3. Mutual Exclusion: Two groups may be specified to be mutual exclusive with respect to membership. That is a user (or an object) may not be a member of mutually exclusive groups at the same time. Furthermore, dynamic mutual exclusion can also be specified where a user may be a member of two mutually exclusive groups but cannot create subjects in the two groups at the same time.
4. Cardinality: There could be many different types of cardinality constraints. For instance, a group could have membership cardinality for users, subjects and objects. Furthermore, a cardinality restriction on the number of relationships that a group may have with other groups could be specified.

Figure 2 shows a snapshot of relationships established between different groups. An important aspect of g-SIS^c is that relationships may change over time as per the varying requirements of the information sharing or collaboration application.

3.2 Configuring LBAC Policies in g-SIS^c

In this section, we discuss how Lattice-Based Access Control [20] policies such as Bell-LaPadula [18] information flow policies can be easily configured using the relationships defined in g-SIS^c. We also demonstrate the agility of g-SIS^c by shoing how it addresses some of the limitations of LBAC models.

Figure 3(a) shows two sample Bell-LaPadula lattices for orgs A and B. The org A lattice has four security labels: L, M1, M2 and H. In LBAC, the domination relationship is reflexive, transitive and anti-symmetric. In this lattice, M1 and M2 dominate L and H dominates M1 and M2 (and L by transitivity). M1 and M2 are incomparable. As per standard terminology, users are assigned one of these four security *clearances* and



(a) A sample lattice for one-directional information flow.

(b) An equivalent g-SIS^c configuration of org A lattice.

Fig. 3. LBAC in g-SIS^c

objects are assigned one of these four *classifications*. Users may then create a subject with a clearance that is dominated by the user's clearance. A subject may read objects whose classifications are dominated by the subject's clearance. A subject may write to objects whose classifications dominate the subject's security clearance.

Figure 3(b) shows an equivalent construction of org A lattice in g-SIS^c. It consists of four groups G_L, G_M1, G_M2 and G_H representing the labels L, M1, M2 and H respectively. Read, write and subject create subordination relationships have been defined according to the specification of the org A lattice in figure 3(a). The subordination relationships are defined in such a manner that a group at the arrow end is subordinate to the group at the tail end. For instance, G_M1 is both create subject and read subordinate to G_H, while G_H is write subordinate to G_M1. Since the relationships are not transitive, we needed to define direct subordination relationships between G_H and G_L as shown in the figure.

Suppose orgs A and B in figure 3(a) need to collaborate on a mission. Specifically, suppose that org B wants to share all its S classified objects (but not its TS and C classified objects) with H cleared users in org A. This is not feasible by simple adjustments to the two lattices in figure 3(a).

Figure 4 shows a construction in g-SIS^c that allows such collaboration scenarios. By assigning a read subordination relation between groups G_H and G_S and groups G_H and G_C respectively, org B is able to allow H cleared org A users to read both S and C classified org B objects. If the subordRrelation is excluded between G_H and G_C, read access can be restricted to S cleared objects.[1] Note that other types of subordination relationships may be specified between org A groups and org B groups to realize

[1] It is true that information may flow from G_C to G_S and thus restricting org A users' access only to G_S may not be completely feasible. Nevertheless, only information that is explicitly copied from G_C to G_S by a subject is available to G_H users. G_H users do not have direct access to G_C objects.

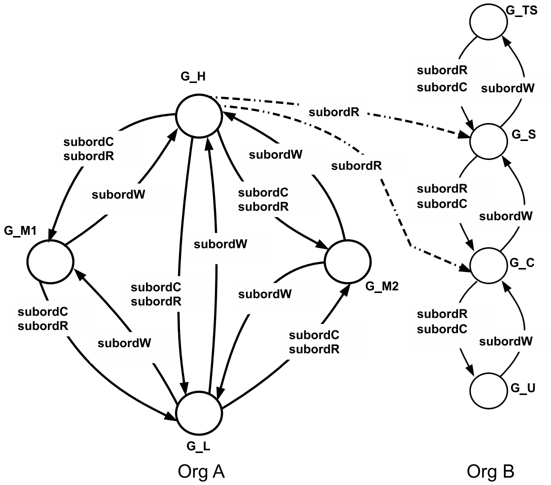


Fig. 4. Agile collaboration enabled by g-SIS^c

other interesting policies. For instance, G_L users may be allowed to write to G_TS objects by defining subordW(G_L, G_TS). These relationships are temporary and may be terminated or modified as collaboration evolves.

Now consider another collaboration scenario illustrated in figure 5. Suppose org A and org B need to collaborate on a mission. They establish groups G1 and G2. TS users/objects from org A and H users/objects from org B may join/be added to G1 (similarly for G2). Conditional membership relations between groups TS and G1 and groups H and G1 are respectively defined. This ensures that if a user leaves the source organization, her membership in G1/G2 is automatically terminated. New information may be created in G1 and G2 as a result of collaboration which may be exported to groups E1 and E2 respectively. The export operation may be performed only by special subjects that have administrative rights in the system. By defining a subordRrelationship between respective source organization groups and these export groups, we allow periodic updates about the mission to be communicated to users in source organizations.

3.3 Configuring Domain and Type Enforcement in g-SIS^c

Domain and Type Enforcement (DTE) (see [28] for example) assigns a subject to a specific domain and an object to a specific type and enforces information flow by specifying the read and write permissions in the form of a matrix. A classic example of the application of DTE is to address the problem of trusted pipelines. Suppose org A (figure 3(a)) needs to enforce that information may flow from L to H but only via M1 or M2.[2] This is not possible to achieve in classic LBAC. Due to the transitive nature

[2] For instance, before a subject at some clearance level may write to a print queue, the document needs to be sent to a trusted print queue manager that visibly stamps every page of the document to be printed with the correct label. In this scenario, the subject should not bypass the queue manager and write to the printer directly.

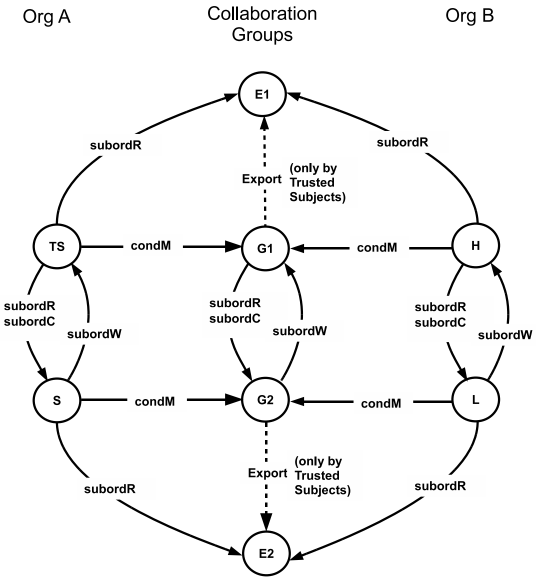
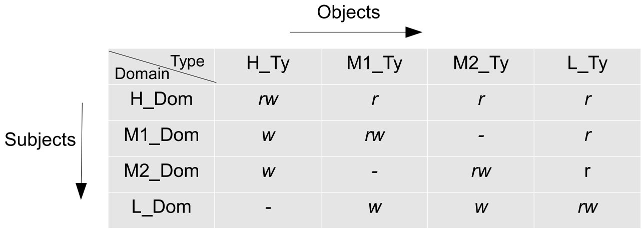


Fig. 5. A collaboration scenario between orgs A and B. The four groups in the middle column (G1, G2, E1 and E2) are established for collaboration between org A (groups in first column) and org B (groups in third column). Groups E1 and E2 are used for exporting new information created as a result of collaboration to G1 and G2 respectively. As indicated, the export operation may be performed only by trusted/administrative subjects.


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*

Fig. 6. A DTE matrix to enforce a trusted pipeline from L to H via M1 or M2 for org A lattice in figure 3(a). Note that a subject in L_Dom cannot write directly to objects in H_Ty.

of domination relation, subjects in L may directly write to objects in H (bypassing M1 and M2). In order to achieve this, DTE assigns subjects to domains (instead of security clearances) and objects to types (instead of classifications) and specifies the rights in the form a matrix as shown in figure 6. Note that, as per this matrix, a subject in L_Dom cannot directly write to H_Ty. However, L_Dom subjects may write, for instance, to M1_Ty and M1_Dom subjects may then read that object and write to H_Ty.

Figure 7 shows an equivalent g-SIS^c configuration for the DTE matrix in figure 6. Users join one of the four first level of groups (H_G, M1_G, M2_G and L_G). The second level of groups represent domains for subjects. A user in one of the first level groups

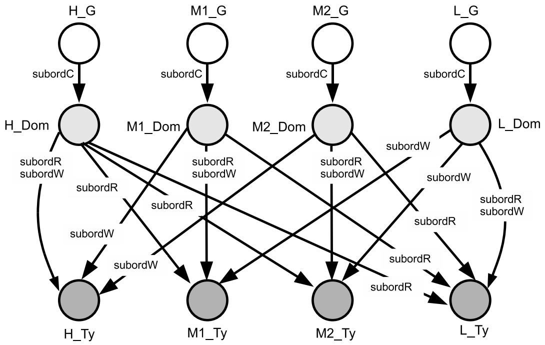


Fig. 7. An equivalent g-SIS^c configuration of the DTE matrix in figure 6. Users join one of the first level of groups (light gray). Users may create subjects in the second level groups representing domains. Objects belong to the third level groups (dark gray) representing types.

may create a subject in the second level domain groups as per the create subject subordination relation (subordC) defined between them. The third level of groups represent the types for objects. Read and write subordination relations are defined between the domain and type groups as per the DTE matrix in figure 6.[3]

3.4 Configuring RBAC Policies in g-SIS^c

In this section, we show the configuration of Role-Based Access Control (RBAC) models [22] in g-SIS^c. In RBAC, a set of roles are created which typically represent job functions of users (employees) in an organization. Each role is assigned with a set of abstract permissions (permission-role assignment) such as credit and debit and users are assigned to specific roles (user-role assignment). Users may activate any combination of roles assigned to them by creating a session. Sessions in RBAC are similar to subjects. The permissions available to a user in a session is the set of all permissions assigned to the set of roles activated in the session by the user. Users may dynamically activate and de-activate specific roles in the session for containment purposes. A family of models have been specified in the well-known RBAC96 [22]. RBAC_0 is the basic model described above. RBAC_1 supports role hierarchies (where a role inherits the permissions of other roles that it dominates). RBAC_2 supports constraints such as separation of duty and role cardinality. RBAC_3 supports all the features of RBAC_0, RBAC_1 and RBAC_2 models.

Here we only discuss the basic model, RBAC_0. g-SIS^c is a model for information sharing where read and write permissions to objects are of concern. However RBAC supports abstract permissions (to accommodate varied permissions in an organization) and hence it is not feasible to directly configure RBAC policies in g-SIS^c. For the

[3] The figure outlines the approach for the construction but excludes some finer details. For instance, users/objects may not be members of domain groups and hence we need a user/object membership cardinality constraint on those groups. Similarly, users cannot join more than one of the first level groups which requires a mutual exclusion constraint between those groups.

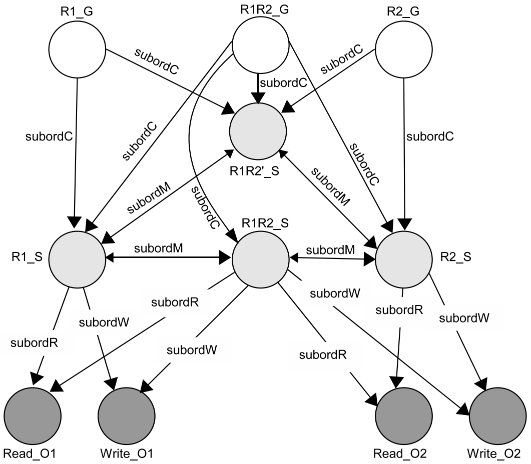


Fig. 8. An equivalent g-SIS^c configuration for the RBAC$_0^{rw}$ model

purpose of our construction, we consider an RBAC$_0$ model with only read and write permissions to objects. Thus for every object, we have two permissions: one to read and the other to write that object. We denote this read-write RBAC$_0$ model as RBAC$_0^{rw}$.

Consider two roles R1 and R2 and two objects O1 and O2. As mentioned earlier, we have two permissions (read and write) for each object, resulting in a total of four permissions. Suppose R1 is assigned permissions to read and write O1 and R2 is assigned permissions to read and write O2. Figure 8 shows an example construction of RBAC$_0^{rw}$ model with two roles R1 and R2 and two objects O1 and O2 in g-SIS^c. The first level of unshaded groups (R1_G, R2_G and R1R2_G) represent groups for user-role assignment. A user may be a member of one of these groups. For instance, users in R1_G have role R1 while users in R1R2_G are assigned to roles R1 and R2. The second level of light-gray groups represent sessions. Note that the group R1R2′_S represents activating a session with no roles assigned. The second level of groups are related to unshaded groups using subordCrelations specifying the rules for subject creation (similar to session in RBAC$_0^{rw}$). Note that subjects may move between the light-gray groups as per the subordMrelation defined. This allows users to activate and de-activate a role dynamically.[4] Finally, the last level of dark-gray groups represent object permissions. Groups Read_O1 and Write_O1 and Read_O2 and Write_O2 represent permissions for objects O1 and O2 respectively. These groups are related to the light-gray groups as per the requirements of permission-role assignment. In the figure, roles R1 and R2 have read and write permissions to objects O1 and O2 respectively. Thus users assigned to both roles R1 and R2 have read and write permissions to both objects O1 and O2. The subordRand subordWrelations defined in figure 8 reflect this configuration.

[4] Again, constraints are necessary for a complete construction. For instance, a subject may move from R1_S or R2_S to R1R2_S only if the user who owns the subject is a member of R1R2_G. Additional constraints are also necessary to ensure that users are not assigned to more than one of R1_G, R2_G and R1R2_G.

4 Conclusion and Discussion

We presented some of design choices for a connected, undifferentiated group g-SIS model and demonstrated its agility with respect to the ease with which changes to information flow/sharing pattern in classic LBAC models can be efficiently handled. We also showed an equivalent representation of an RBAC model with read-write permissions. Because of this result and as per [23], we claim it is feasible to configure Discretionary Access Control policies in g-SIS^c. This positive result allows a system to use the same trusted computing base to configure any of these policies. Prior work on non-transitive information flow in the literature (see [29] for example) is relevant in this context. However, g-SIS is far richer and brings in additional concepts such as subject creation and movement subordination. Furthermore, g-SIS accommodates various useful semantics for group operations such as join and leave for users and add and remove for objects as illustrated in [4,5].

Another area of related work is that of Dynamic Coalition (see for example [30,31]). This problem is concerned about forming a coalition amongst different organizations, for instance, in response to a crisis. Most of the security research in this domain has been carried out in the enforcement or E-layer with the exception of a few. (For instance, in [32,33], the authors focus on enriching role-based access control to address the challenges involved in dynamic coalition.) While dynamic coalition is a very broad and large-scale problem, the focus of g-SIS models is more on information sharing. Specifically, it focusses on read and write permissions to objects and containing subject level information flow. We believe that g-SIS policy models can be beneficially used in dynamic coalition scenarios.

Our future work involves formal specification and analysis of a connected group g-SIS model. In our prior work [4,5,6,34], we have formally specified and analyzed an isolated group g-SIS model. We are exploring candidate core security properties for the connected g-SIS model similar to those of the isolated model. A major challenge in the connected model is that relationships are not static like that of LBAC models. Modern information sharing scenarios are dynamic and inter-group relationships change over time. This complicates information flow analysis in the connected model. For instance, information may flow from group g1 to g3 even if g1 and g3 never existed at the same time (it may currently flow from g1 to g2 and from g2 to g3 in the future). Thus, unlike LBAC, information flow properties tend to be temporal in nature in g-SIS^c.

Acknowledgments

The authors are partially supported by grants from NSF, AFOSR MURI, THECB, State of Texas Emerging Tech. Fund and Intel Corp.

References

1. Saltzer, J., Schroeder, M.: The protection of information in computer systems. Proceedings of IEEE 63(9), 1278–1308 (1975)
2. Wikipedia: Analog hole (September 2009) (Online; accessed December 15, 2009)

3. TCG: TCG specification architecture overview (August 2007), `http://www.trustedcomputinggroup.org`
4. Krishnan, R., Sandhu, R., Niu, J., Winsborough, W.: A conceptual framework for group-centric secure information sharing. ACM Symposium on Information, Computer and Comm. Security (March 2009)
5. Krishnan, R., Sandhu, R., Niu, J., Winsborough, W.H.: Foundations for group-centric secure information sharing models. In: Proc. of ACM Symposium on Access Control Models and Technologies (2009)
6. Krishnan, R., Sandhu, R., Niu, J., Winsborough, W.: Towards a framework for group-centric secure collaboration. In: Proceedings of IEEE International Conference on Collaborative Computing (2009)
7. Krishnan, R., Sandhu, R., Ranganathan, K.: PEI models towards scalable, usable and high-assurance information sharing. In: ACM Symposium on Access Control Models and Technologies (SACMAT 2007), pp. 145–150. ACM, New York (2007)
8. Sandhu, R.: The PEI framework for application-centric security. In: Proceedings of 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (2009)
9. Sandhu, R., Ranganathan, K., Zhang, X.: Secure information sharing enabled by trusted computing and PEI models. In: Proc. of ACM Symp. on Inf. Computer and Comm. Security, pp. 2–12 (2006)
10. Sandhu, R., Samarati, P.: Access control: Principles and practice 32(9), 40–48 (1994)
11. OrangeBook: Trusted Computer System Evaluation Criteria. DoD National Computer Security Center (December 1985)
12. Graham, G., Denning, P.: Protection-principles and practice. In: Proceedings of the AFIPS Spring Joint Computer Conference, vol. 40, pp. 417–429 (1972)
13. Lampson, B.: Protection. ACM SIGOPS Operating Systems Review 8(1), 18–24 (1974)
14. Graubart, R.: On the Need for a Third Form of Access Control. In: Proceedings of the 12th National Computer Security Conference, pp. 296–304 (1989)
15. McCollum, C., Messing, J., Notargiacomo, L.: Beyond the pale of MAC and DAC - defining new forms of access control. In: Proceedings of the 1990 IEEE Symposium on Security and Privacy, pp. 190–200 (1990)
16. Abrams, M., Heaney, J., King, O., LaPadula, L., Lazear, M., Olson, I.: Generalized Framework for Access Control: Towards Prototyping the ORGCON Policy. In: Nat. Comp. Sec. Conf. (1991)
17. Park, J., Sandhu, R.: Originator control in usage control. In: Policies for Distrib. Syst. and Networks (2002)
18. Bell, D., La Padula, L.: Secure computer systems: Unified exposition and multics interpretation. Technical Report ESD-TR-75-306 (1975)
19. Denning, D.: A Lattice Model of Secure Information Flow. Communications of the ACM 19(5), 236–243 (1976)
20. Sandhu, R.: Lattice-Based Access Control Models. IEEE Computer 26(11), 9–19 (1993)
21. Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, D., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. on Inf. and Syst. Security (TISSEC) 4(3), 224–274 (2001)
22. Sandhu, R., Coyne, E., Feinstein, H., Youman, C.: Role-Based Access Control Models. IEEE Computer, 38–47 (1996)
23. Osborn, S., Sandhu, R., Munawer, Q.: Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. ACM Trans. on Inf. and Syst. Security 3(2), 85–106 (2000)
24. Park, J., Sandhu, R.: The UCON$_{ABC}$ usage control model. ACM Transactions on Information and System Security (TISSEC) 7(1), 128–174 (2004)

25. XACML: OASIS eXtensible Access Control Markup Language (April 2009), `http://www.oasis-open.org/committees/xacml/`
26. Levin, R., Cohen, E., Corwin, W., Pollack, F., Wulf, W.: Policy/mechanism separation in Hydra. In: 5th ACM Symposium on Operating Systems Principles, pp. 132–140 (1975)
27. Rafaeli, S., Hutchison, D.: A survey of key management for secure group communication. ACM Computing Surveys, 309–329 (September 2003)
28. Badger, L., Sterne, D.F., Sherman, D.L., Walker, K.M., Haghighat, S.A.: Practical domain and type enforcement for unix. In: SP 1995: Proceedings of the 1995 IEEE Symposium on Security and Privacy, Washington, DC, USA, p. 66. IEEE Computer Society, Los Alamitos (1995)
29. Foley, S.N.: A model for secure information flow. IEEE Symposium on Security and Privacy, 248–258 (1989)
30. Phillips Jr., C.E., Ting, T., Demurjian, S.A.: Information sharing and security in dynamic coalitions. In: SACMAT 2002: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, pp. 87–96. ACM, New York (2002)
31. Shands, D., Jacobs, J., Yee, R., Sebes, E.: Secure virtual enclaves: Supporting coalition use of distributed application technologies. ACM Transactions on Information and System Security (TISSEC) 4(2), 103–133 (2001)
32. Freudenthal, E., Pesin, T., Port, L., Keenan, E., Karamcheti, V.: drbac: Distributed role-based access control for dynamic coalition environments. In: ICDCS 2002: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS2002), Washington, DC, USA, pp. 411–420. IEEE Computer Society, Los Alamitos (2002)
33. Cohen, E., Thomas, R.K., Winsborough, W., Shands, D.: Models for coalition-based access control (CBAC). In: SACMAT 2002: Proceedings of the Seventh ACM Symposium on Access Control Models and Technologies, pp. 97–106. ACM, New York (2002)
34. Krishnan, R., Niu, J., Sandhu, R., Winsborough, W.: Stale-safe security properties for group-based secure information sharing. In: Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering, pp. 53–62. ACM, New York (2008)