PROJECT SUMMARY

Overview:

The need to share information while confining it to authorized recipients is one of the most challenging problems in cybersecurity. Toward this end, this project focuses on the policy problem of specifying, analyzing and enforcing information sharing policies. The PI recently introduced a novel approach called Group-centric Secure Information Sharing (g-SIS), and developed an initial set of models, analysis and case studies. A fundamental philosophy of g-SIS is to bring users and information together in a group--an intuitive construct for information sharing. A good metaphor for a group is that of a secure meeting room, where people and information are brought in, and taken out over time. The PI has studied temporal interactions of people and information in a single group, and consequent policy-based impact on authorization. This project will develop these promising early results into a mature theory of multiple, connected groups with broad, real-world application. The objective of g-SIS is to support agile responsiveness to shifting circumstances. To ensure practical relevance, the PI, in collaboration with the Center for Infrastructure Assurance and Security (CIAS) at UTSA, will extensively study and pursue real-world information sharing scenarios faced by communities across the nation. On the educational and outreach front, this project strongly ties in with curriculum development, working with students from under-represented groups within UTSA and high schools in San Antonio, and launching multi-disciplinary teams of under-graduate UTSA engineering and business school students, in entrepreneurial pathways.

Intellectual Merit :

The project’s research methodology separates policy layer specifications (P-layer) of g-SIS models from the enforcement layer (E-layer). P-layer assumes idealized, accurate access control and ignores issues such as decentralization, network latency and local caching, which impact authorization enforcement. At the P-layer this project will develop multiple, connected group models based on relationships such as subordination, conditional membership, mutual exclusion, etc. E-layer specifications incorporate the inevitable distributed nature of maintaining authorization state in a realistic system. A progression of specifications will be developed from P-layer to E-layer, culminating in a specification that can be implemented by a reasonably competent programmer. The P-layer specifications are necessarily mathematically intricate whereas E-layer specifications are intuitively comprehensible to system architects and developers. Along this progression, this project will employ rigorous formal methods to ensure that specifications at each layer are consistent with respect to authorization decisions. A novel goal is to develop a policy-based approach for safe approximation of E-layer specifications to that of P-layer. Previously, the PI has analyzed single-group P/E-layer models including stale-safe approximation. This project will study alternate distributed E-layer specifications including their stale-safety and other novel safe approximation properties. A proof-of-concept implementation, and evaluation will be carried out on an OpenStack-based cloud computing platform. The research methodology of progressive specification leading to an easily implementable specification is applicable to models beyond g-SIS.

Broader Impacts :

This research addresses challenging topics important to national priorities and interests of an information-rich society. A number of activities planned in this project will have a broader impact at various levels. The PI will work closely with CIAS toward prototyping and utilizing the developed cloud-based information sharing platform for use in its ongoing community cybersecurity information sharing efforts. This has the potential for national impact. In collaboration with the Center for Innovation and Technology Entrepreneurship at UTSA, the PI will work with multiple teams of undergraduate UTSA students toward entrepreneurial pathways, while leveraging the outcomes of this project. UTSA is a research-intensive, Hispanic serving, and Minority institution with more than 60% of its undergraduate student body from underrepresented groups, many of these being first-generation college students. Thus this activity has the potential for a tremendous impact on the UTSA undergraduate student body. Finally, this project incorporates strong curriculum development within UTSA, and outreach activities involving high school students in San Antonio.

C Project Description

C.1 Motivation and Objectives

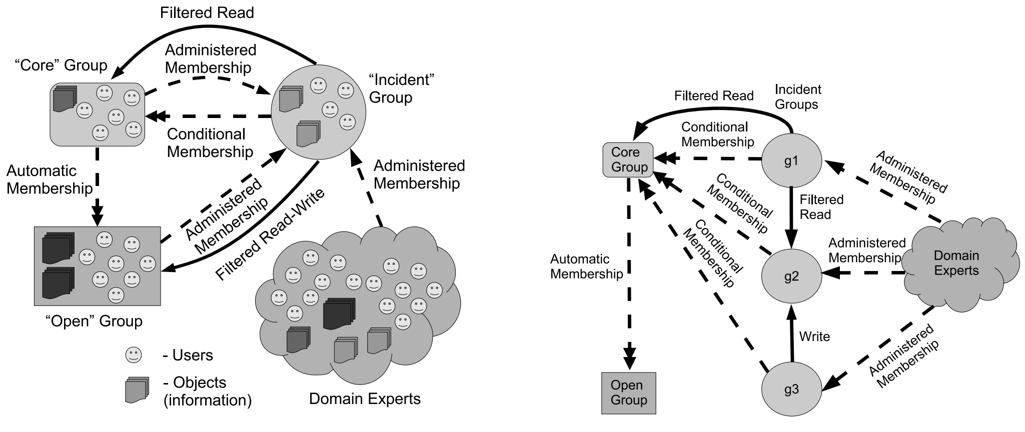
The need to share information while protecting it is one of the oldest and most challenging problems in cyber security. As early as 1975, Saltzer-Schroeder [44] identified the desirability and difficulty of maintaining "some control over the user of the information even after it has been released." This problem has only compounded over the last decades with the advancement of information technology. At the same time, our increasingly information-rich and information-dependent society needs to leverage *secure information sharing* (SIS) to fully benefit from the productivity, and social and national security benefits of the ongoing cyber revolution. The focus of this project is on the *policy challenge* of specifying, analyzing and enforcing SIS policies. A basic premise is that this requires models that have intuitive grounding and rigorous mathematical foundations, are usable by a typical netizen, and enforceable in distributed systems.

This project proposes the notion of a *group* as an intuitive construct for information sharing. A good metaphor is that of a secure meeting room. People join and leave (who may later re-join) the room, and similarly information is added to, and removed (which may be re-added) from the room. During the course of the meeting, new information may be created and exported out of the room. Furthermore, information in the room may be updated, leading to different versions. 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 instance, in order to be initially authorized, a notion of simultaneous presence between the user and information is an inherent property of a meeting room. The PI has studied such temporal issues that affect authorization in a single group (i.e., where different groups are unrelated to, and independent from each other in the context of authorization) extensively [31]. In this proposal, this approach is referred to as Group-Centric Secure Information Sharing (g-SIS). This project will develop these promising early results into a mature theory of multiple, connected groups with broad, real-world application. The inter-relationships between different groups and corresponding users' actions across those groups require a careful model design so as to be usable in real-world circumstances.

Motivational Scenario: Consider a scenario where organizations share cyber incident related information among each other for being better prepared for a cyber attack [3]. UTSA's Center for Infrastructure Assurance and Security (CIAS) [1] has decade long experience with respect to such SIS related activities in communities across the nation. The CIAS conducts cyber security exercises that emphasize the importance of protecting a community's critical infrastructure through awareness, information sharing, and training.

The scenario is developed in the context of the life cycle of a cyber security incident in a community (a city, a county, or a state), as illustrated in figure 1(a). This scenario is from real-world experiences of the CIAS Director, Dr. Greg White, based on an interview conducted by the PI. A basic premise is that several different community organizations need to be involved in managing a community scale cyber security incident, including recognizing its existence. These include local government organizations, e.g., police, fire, health, quasi-government organizations, e.g., energy and power utilities, and private organizations, e.g., telecommunications, internet and financial service providers, and so on. In the steady state, two "permanent" groups, the open and core groups, that enable SIS across these organizations are proposed. The notion of the open group is that it is open to voluntary participation by employees. The open group fosters unmoderated, but contained information sharing. Information shared in the open group is likely to be of relatively low sensitivity. The core group, on the other hand, enables sharing more sensitive information, and its membership is administered. Perhaps, each organization gets to nominate a few participants as appropriate. Members of the core group are automatically enrolled in the open group for convenience.

When a cyber incident occurs or is suspected, an "incident" group is established, which would focus on that particular incident. Members of the incident group are selected administratively from the core group. The criteria for selection could involve special expertise or representation of a particular sector such as telecommunications. Membership in the incident group is conditional on continued membership



(a) Cyber-incident life-cycle

(b) Dynamic group relationship

Figure 1: Community Cyber Incident Management.

in the core group. Additional expertise from outside the core group may be appropriate within the incident group—such as independent domain experts. The incident group is provided a filtered read capability into the core group which enables selective access to core group information. Further, the incident group has a filtered read/write capability to the open group. This can allow solicitation of particular expertise or information from that group, for example. Over time the incident group may dissolve or persist depending on circumstances. Figure 1(b) illustrates multiple incident groups, which may start of as independent, but develop relationships over time, indicated by the filtered read from g1 to g2, and the write from g3 to g2. The goal of this project is to capture the dynamism of SIS exemplified by this community-based scenario, and build models with strong mathematical foundations and intuitive appeal. The motivation for g-SIS is agile dynamism and evolution in response to real-world events and requirements.

Specific Research Objectives: A summary of the research objectives of this project is provided below.

1. Design g-SIS policy models, specify security properties, and perform attendant analysis. This objective focusses on developing formal models for connected groups, based on well-laid out design principles. Global security properties of the models will be specified along with formal security analysis. This task will design and select suitable relationships among groups that facilitate administration. For instance, membership in one group might confer rights to access objects in other groups. Alternatively, membership in one group may be contingent on membership in another.

2. Design enforcement level specifications and "stale-safe" security properties. A basic premise of the approach is that the policy problem in specifying and analyzing the intrinsic application policy (P-layer) should be clearly separated from enforcement policy (E-layer) issues that arise due to the realities and practicalities of a decentralized, and distributed system (as first articulate in HYDRA [34]). The overarching goal of this task is to design E-layer specifications that are approximate to that of the ideal P-layer, while maintaining a measure of "stale-safety"–i.e., consistency between P- & E-layers.

3. Develop a prototype of g-SIS in OpenStack cloud platform. This objective focusses on developing a prototype of the g-SIS E-layer specifications in the OpenStack infrastructure as a service cloud platform [7]. The prototype will be based on the STIX-TAXII-CybOX framework [5] that defines a set of standards and protocols for cyber threat information sharing, recently introduced by the DHS [2] and MITRE [6].

4. Conduct rigorous evaluation. This objective focuses on a rigorous evaluation of various aspects of this project including design adequacy, usability, performance and practical adoption. Mechanisms used include live scenarios from CIAS, and leveraging Amazon Mechanical Turk for usability analysis.

Specific Education and Outreach Objectives:

1. Launch multi-disciplinary teams of UTSA students onto entrepreneurial pathways. This project lends itself to designing products and services that are of great interest to private and public sector organizations. The Center for Innovation and Technology Entrepreneurship (CITE) at UTSA is an NSF designated

Innovation Corps (I-Corps) Site. CITE hosts technology venture competitions for teams of undergraduate engineering and business school students as part of their Senior Design projects. Most of these students (>60%) are from underrepresented minority groups, many of whom are first-generation college students. In collaboration with CITE, the PI plans to mentor multiple teams of these students on topics related to this project over the next 5 years, and help them launch into entrepreneurial pathways. The PI is excited about the significant impact it can have on UTSA's underrepresented student body.

2. Enhance participation from traditionally underrepresented groups in science and engineering. In collaboration with the Center for Excellence in Engineering Education (CE3) at UTSA, the PI plans to recruit underrepresented students in a targeted manner to participate in this research project. The PI will offer independent study courses and targeted implementation projects to additional students who cannot be directly funded by this project. CE3's primary objective is to increase retention and graduate rates in the College of Engineering at UTSA. In addition, the PI will work with underrepresented high school students to increase awareness of security issues, cloud computing, etc. via summer camps offered by UTSA's Interactive Technology Experience Center (iTEC).

4. Develop a strong curriculum on cybersecurity in the PI's department at UTSA. Finally, the PI plans to develop a curriculum on cybersecurity and cloud computing in his department at UTSA. Currently, he teaches "Introduction to computer and network security" (EE 5453). Based on this project, the PI will enhance this curriculum with topics such as formal methods in designing and analyzing security policies, and build a course on cloud computing security based on the open-source cloud platform, OpenStack.

PI Qualifications: The PI has a strong background in utilizing formal methods to characterize security models and verifying security properties against those models [31]. The PI also has formally studied attribute-based access control [24, 25] (ABAC) and online social networks [19]. These experiences will support executing the tasks involved in meeting research objectives #1 and #2. At UTSA, the PI has built an infrastructure as service (IaaS) cloud laboratory for his research using OpenStack. In addition, the PI has explored collaboration in multi-cloud computing environments [46], and has co-advised a PhD student who investigated ABAC aspects in IaaS in his thesis [22]. These experiences will support executing research objective #3. Finally, collaboration efforts with CIAS will support aspects of objective #4.

The PI is currently funded by an NSF grant CNS-1423481–"TWC:Small: Attribute-Based Access Control for Infrastructure as a Service Cloud" that investigates foundations for a formal theory of ABAC with IaaS as the application domain. This CAREER proposal is highly synergistic with CNS-1423481 as it seeks to investigate formal aspects of group-based information sharing models. While at a theoretical level, ABAC can simulate most other forms of access control, its application to dynamic information sharing is not understood.

On the education side, the PI has strived to integrate research and education since joining UTSA in 2010. As a recognition, the PI is a recent recipient of 2015 University of Texas System Regents' Outstanding Teaching Award. Using a seed grant from Intel Inc., the PI has developed a curriculum on building security services in cloud. This led to the development of a new introductory undergraduate course on cloud computing that was offered by the PI in Spring 2013 (EE 4953: Introduction to Cloud Computing), in which students developed Facebook-like social networking applications on Heroku [4]. The PI has also mentored a number of underrepresented students both within and outside UTSA, and lectured at high schools in San Antonio. These experiences will help execute the educational and outreach objectives.

C.2 Technical Approach

As discussed earlier, in g-SIS, groups can be isolated, or connected. Isolated groups are independent from each other in the sense that a user in one group has no authorization to access information in another, as a consequence of membership in that of another. If groups are connected via certain inter-group relationships such as conditional membership, subordination, mutual exclusion, prerequisite memberships, cardinality, etc., a user's membership in one group can affect her authorization to information in that of another.

To manage complexity, this project will incrementally develop a progression of g-SIS models. At each step, it will develop a sequence of Policy/Enforcement layer (P/E-layer) specifications and properties, and attendant formal analysis. Fig. 2 illustrates the specifications, discussed in detail below.

Stateless vs. stateful specifications. At the P-layer, a collection of specifications, including stateless and stateful specifications of the isolated and connected group models will be developed. A specification is *stateless* if a user's authorization to an object is specified based solely on the history of user and object actions. The stateless specification focuses on the history of actions that lead up to authorization decision. By contrast, the *stateful* specification introduces data structures into states that record sufficient information about the history to support making efficient authorization decisions. A major research task here is to formally show that the stateful specification is *authorization equivalent* to the stateless specification. That is, a user will be authorized to access an object in the stateful specification if and only if the stateless specification would authorize the same user action given the same group-action history. The stateless and stateful isolated models are denoted SL^{i} and SF^{i} respectively. The stateless and stateful connected group models are denoted SL^{c} and SF^{c} respectively.

	Isolated model (g-SIS$^{\mathrm{i}}$)	**Connected model (g-SIS$^{\mathrm{c}}$)**
P-Layer Specifications	Stateless Isolated (SL^{i}) Stateful Isolated (SF^{i})	Stateless Connected (SL^{c}) Stateful Connected (SF^{c})
E-Layer Specifications	Stateful Isolated Distributed ($\mathrm{SF}^{\mathrm{i}}_{\mathrm{d}}$)	Stateful Connected Distributed ($\mathrm{SF}^{\mathrm{c}}_{\mathrm{d}}$)

Figure 2: g-SIS Roadmap

Separation of stateful specification from that of stateless is an efficient design methodology [31], increasing the level of assurance that designs meet the intended security objectives. By focusing initially on a stateless specification, we avoid being distracted by issues surrounding mechanism for supporting authorization decisions. It is also convenient to design, specify, and check, in the stateless context, global "core" properties that the specification should satisfy.

Distributed E-layer specifications. Both stateless and stateful P-layer specifications are developed assuming an idealized, centralized setting and abstracts out the realities of enforcing policies in a distributed system. The next level of E-layer specifications will incorporate the distributed nature of maintaining authorization state in a realistic system. Specifically, distributed stateful specifications for isolated and connected group models, denoted $\mathrm{SF}^{\mathrm{i}}_{\mathrm{d}}$ and $\mathrm{SF}^{\mathrm{c}}_{\mathrm{d}}$ respectively, will be developed. By incorporating details of interaction between distributed authorization system components, the distributed specifications provide sufficient detail for a programmer to implement the system directly from them.

A natural concern is how distributed stateful specifications in the E-layer compare to the P-layer stateful specifications: what are the relationships between $\mathrm{SF}^{\mathrm{i}}_{\mathrm{d}}$ and SF^{i}, and between $\mathrm{SF}^{\mathrm{c}}_{\mathrm{d}}$ and SF^{c}? The $\mathrm{SF}^{\mathrm{i}}_{\mathrm{d}}$ (resp. $\mathrm{SF}^{\mathrm{c}}_{\mathrm{d}}$) specification defines authorization decisions that approximate those of the SF^{i} (resp. SF^{c}) in a manner that provides the desired application-dependent balance between resource availability and timely propagation of authorization-state changes in a distributed system. One approach for addressing this issue is to specify a number of *stale-safe* security properties of distributed systems that have the effect of constraining the degree to which decisions made by these distributed systems, approximate those made by the corresponding idealized (stateful) systems. For example, one of the stale-safe properties that the PI has proposed in the past [29] requires that a user be authorized to access an object (information) only if the authorization was confirmed in the *recent past*. A major research task is to generalize such stale-safe properties to show that the distributed, stateful specifications approximate the corresponding idealized, P-layer stateful specifications, in accordance with appropriate stale-safe properties.

Theoretical underpinnings. This research will make extensive use of formal methods to develop models and to analyze and verify security properties. For developing stateless specifications, we plan to use first-order linear temporal logic (FOTL) [40], a natural candidate because it specifies authorization based on the history of various actions. These actions include user join and leave, object add and remove, and a number of group operations such as group creation and termination, and inter-group relationship set up and tear down. The PI's prior work in this area affirms the effectiveness of FOTL for this purpose [31].

For stateful specifications, we plan to employ a combination of FOTL and extended finite state machines (EFSMs) [37, 38]. To verify stateless and stateful authorization equivalence, and to show stale safety between specifications, a combination of manual and automated proof techniques will be used. An automated technique that is planned to be leveraged the most is model checking [14, 15], which takes an FSM (finite state machine) and a propositional temporal logic formula and determines whether the FSM behavior satisfies the formula. By appropriately choosing the EFSM and formula, this technique can also be used to verify relationships between two EFSMs or between two propositional temporal formulas. Because it suffers from the state-space explosion, model checking is directly applicable only when the state space is finite (and relatively small), and when the formula is propositional. When domains over which variables range are finite, first-order formulas can be translated to propositional formulas.

Unfortunately, even when the numbers of groups, users, subjects, and objects are very small, direct verification via model checking is often infeasible due to explosion of possible states. It will be necessary to use additional techniques. The PI will explore using approaches such as decomposing the specifications [26, 36, 41]. In this approach, one manually proves that relationships that can be shown to hold via automated means between the components, together give the desired results for the full specifications. Moreover, because the aim is to specify systems of unbounded size, domains and state spaces are not generally finite. For those specifications in which groups are isolated, the authorization of a user to access an object depends only on their action history with respect to a group they both are in. The PI will explore using more general "small model theorems," which guarantee that specifications that are given by (or are equivalent to) FOTL formulas having certain syntactic structures can be model checked using domains of bounded size, the results generalize to unbounded domains [53].

When working with coupled groups, more powerful techniques must be developed. One approach to tackle this problem builds on [33], which develops static analysis techniques for inferring safety properties. The approach uses abstraction based on 3-valued logic, combined with a rich suite of operations that can be used to control the degree of precision with which the abstract representations characterize concrete structures. This gives the opportunity to balance the analysis's performance and its precision.

C.3 Related work

P-Layer Achieving security in dynamic coalitions is a broad and complex problem, which has been extensively studied. (See [9, 17, 21, 27, 28, 35, 39, 45, 49] to name a few.) The intention of g-SIS is that the security models already in use by participating organizations need not be necessarily integrated with the g-SIS system. Instead, g-SIS should be orthogonal and complementary to those deployed models. This is sharply distinguished from those of prior work, such as in the context of dynamic coalitions, which generally focus on integrating security infrastructure across organizational boundaries. For instance, the work of Shands *et al.* [45] on secure virtual enclaves, focusses specifically on building a middleware infrastructure amongst participating organizations while being agnostic to the security policy. Cohen *et al.* [17] have proposed a complex role-based model for integrating security infrastructure of multiple organizations in a coalition. Furthermore, prior work on dynamic coalition focuses on sharing static resources such as a service or a computing facility. g-SIS focuses explicitly on information or object sharing, the life-cycle of which is highly dynamic. Information flow is also a major concern in such a scenario. g-SIS models explicitly handle information flow issues by distinguishing users (people) from software subjects to restrict permissions, thereby containing unpredictable information flow. The work of Li *et al.* (RT [35]) falls in the domain of dynamic coalitions. RT can be a powerful administrative model for g-SIS. In the proposed research section, administrative models for g-SIS and RT are discussed in further detail.

E-Layer The proposed research work on the notion of an enforcement policy that safely approximates an ideal P-layer specification to a distributed E-layer specification is novel. The work of Lee *et al.* [32] on safety and consistency in trust management systems is the closest that can be found in the literature, but focuses exclusively on credentials (attribute certificates). Within the scope of a trust negotiation protocol, it focuses on obtaining fresh information about the revocation status of credentials to avoid staleness. Another set of related works concern authorization recycling by Crampton et al [18, 50], which are specific to a form of access control such as Bell-LaPadula or RBAC. However, as will be discussed later, our formalism is not restricted to a specific domain or a type of credential, and is based on the notion of a "refresh time," that is the time when an attribute value was known to be accurate. Because Lee *et al.* admit only attribute certificates as carriers of attribute information, there is no notion of refresh time in their framework. Furthermore, our enforcement policy specification is intended to support automatic verification techniques such as model checking, which is critical for practical adoption as an E-layer specification, since subsequent design changes can be rapidly verified against the enforcement policies.

C.4 Proposed Research

This section describes in detail each of the proposed research tasks. A brief overview of the formal constructs concerning isolated g-SIS (g-SIS^i) [31] is discussed below. As mentioned earlier, in g-SIS^i, users join and leave (and possibly re-join) a group, and objects may be added, removed (and possibly re-added) into the group (figure 3). A set of *core properties* can be defined that are required to be satisfied by any g-SIS^i specification. Core properties are designed to focus on one global system property at a time, which allows careful consideration of security properties that are necessary to hold in any g-SIS^i specification.

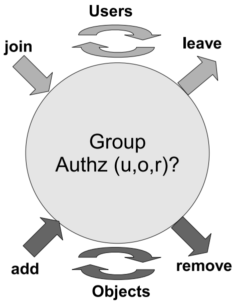


Figure 3: Group Sharing

Authorization Provenance—An Example Core Property. Intuitively, a user should not be authorized to read an object until a point at which both the user and object are simultaneously group members. If a user has authorization to read an object in a particular group (denoted Authz) in some state in any given trace, then there was an overlapping period of user and object membership in that group at least once in the past. This can be succinctly stated in FOTL[1] as, $\forall u.\forall o.\forall g.\forall r.$

$$(\neg\mathrm{Authz}(u,o,g,r)\,\mathcal{W}\,(\mathrm{Authz}(u,o,g,r)\wedge(\neg\mathrm{Leave}(u,g)\,\mathcal{S}\,\mathrm{Join}(u,g))\wedge(\neg\mathrm{Remove}(o,g)\,\mathcal{S}\,\mathrm{Add}(o,g))))$$

That is, starting from the initial state, Authz (for a specific user, object, right) cannot begin to hold until the user and the object become simultaneous members in the group. Note that once Authz begins to hold, simultaneous membership is not required in some future state for Authz to start holding again. A set of such properties are defined, and shown to be logically independent and consistent. A trivial example of an FOTL-based stateless specification for g-SIS^i that satisfies such core properties is as follows:

$$\square(\mathrm{Authz}(u,o,r)\leftrightarrow(\neg\mathrm{Leave}(u,g)\wedge\neg\mathrm{Remove}(o,g))\,\mathcal{S}\,(\mathrm{Add}(o,g)\wedge(\neg\mathrm{Leave}(u,g)\,\mathcal{S}\,\mathrm{Join}(u,g))))$$

It states that a user is allowed to read an object if the user and object are current members in the group, and the object was added at a time the user was a member. In order to develop a stateful specification for the above stateless specification, it is necessary to maintain user and object attributes that kept track of the time at which user joined ($\mathrm{Join\_TS}$) or left ($\mathrm{Leave\_TS}$), and the time at which objects were added ($\mathrm{Add\_TS}$) or removed ($\mathrm{Remove\_TS}$). Thus authorization in the stateful specification is given by:

$$(\mathrm{Authz}(u,o,r)\leftrightarrow\mathrm{Join\_TS}(u,g)\leq\mathrm{Add\_TS}(o,g)\wedge\neg\mathrm{Leave\_TS}(u,g)\wedge\neg\mathrm{Remove\_TS}(o,g))$$

By using model checking [16] and manual analysis, it can be shown that the stateful specification of g-SIS^i is *authorization equivalent* to its stateless specification. Another approach for showing equivalence is to treat them as 2 different specifications, and compare their expressive power using the framework developed by Tripunitara and Li [48].

[1] The formula ($p\ \mathcal{S}\ q$) states that at some trace index less than or equal to the current one, q held, and since that point, p has held at every following index. The formula ($p\ \mathcal{W}\ q$) states that p holds at each subsequent trace index unless and until a point is reached at which q holds. The formula ($\square\ p$) means that p holds in the current, and all future states.

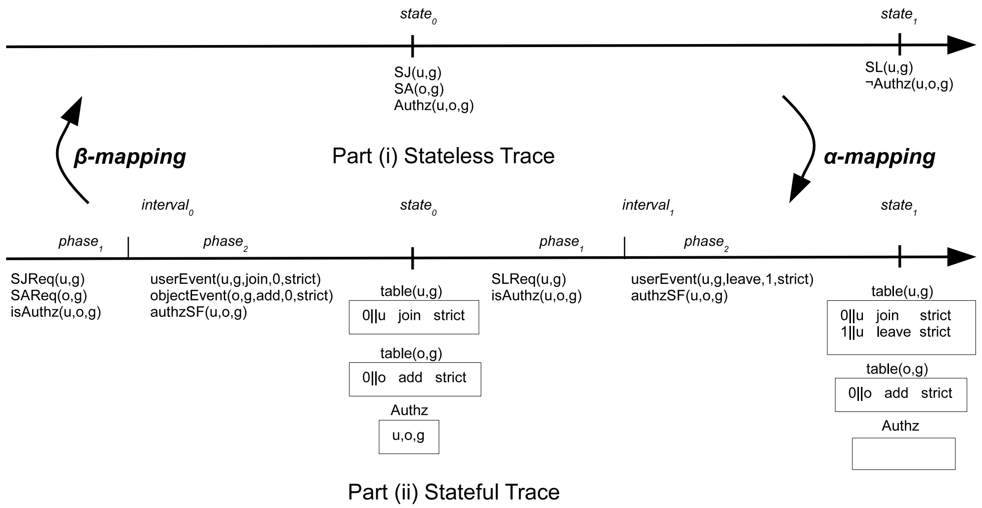


Figure 4: α and β mapping. Part(i) shows a sample stateless trace and part(ii) shows a corresponding stateful trace. Note that the stateful trace generates the required action and authorization requests during the time interval leading up to the state.

C.4.1 P-layer Research Tracks

P0: Define a set of guiding principles for g-SIS model design. This task will develop a core set of principles that will guide the design of connected g-SIS models (g-SIS^c). A few candidates below:

Principle 1. Each group's admins should maintain autonomy over its constituents (i.e., its users and objects), and what objects in that group are available to members of external groups. This principle recognizes that the access control models within each group can be completely independent from that of others. Within each group, access decisions may be made based on roles, attributes, etc. of the user and object in question.
Principle 2. In g-SIS^c, global constraints for information access need to be specifiable by an administrator, and preserved, as relationships between groups are established and disbanded. Examples of such constraints include, group g1's users should never be able to access objects in group g2, or user u1 of group g1 should never get access to objects in g2. Many of these properties are safety and liveness properties [8, 47].
Principle 3. To support review functionality, authorization effects of inter-group operations should be efficiently verifiable. When an admin establishes a new type of relationship between two groups, its effect on authorizations of specific users on specific objects of specific groups should be verifiable efficiently.
Principle 4. The g-SIS^c model should allow for distributed administration. That is, administrators of different groups should be able to perform their inter-group relationship establishment and disbanding operations without any dependency on administrators of relevant groups, so long as the global constraints are obeyed. In a sense, this principle works against Principle 3, since reviewing effects of distributed changes is harder. A major research challenge is to design g-SIS^c while balancing these two requirements.
P1: Defining and verifying ideal stateful specification—isolated case Following the outline in section C.2, the next step is to specify and verify a stateful specification of the isolated g-SIS system (g-SIS^i). The stateful specification takes the form of a EFSM that maintains data structures summarizing sufficient information about the action history to support making user authorization decisions.

Figure 4 illustrates the distinction between centralized stateless and stateful specifications for a simple example of a g-SIS^i specification. A Strict Join (SJ) operation allows users to only access objects added after she joins the group, while a Strict Add (SA) operation restricts the added object to be accessible only to the current members of the group at add time. In the figure, the stateless trace keeps track of such join, and add events. The stateful trace maintains data structures (e.g. relations with time-stamped tuples of events) at each state that can be queried to make authorization decisions. Thus, in addition to keeping track of whether users and objects are currently members of groups, it is necessary to keep additional information such as the relative order in which users and objects were added, since some versions of the join and add operations enable user access only to objects added after the user joined. Another example data structure keeps track of whether a user had access to an object prior to leaving the group, since some versions of the leave operation permit user access to objects if they had access when they left.

The correctness of an EFSM with respect to the FOTL specification of authorization is expressed at the level of traces. At each state in a trace, authorization is given by the interpretation given to the state predicate, Authz. An EFSM is correct wrt a stateless specification, π, if the EFSM generates exactly those traces that satisfy π (β-mapping the figure 4). This ensures the two formalisms agree about when each user access is authorized. Additionally, it ensures that the two agree about which group-action sequences are legal. The stateless specification, π, includes sub-formulas that require group-action sequences to be *well formed*, which captures requirements such as a user can leave a group only if he is currently a member.

The strategy for showing correctness combines automated and manual techniques. First model checking is used to verify that over small domains (carriers), all traces generated by the stateful specification satisfy that of the stateless. This tells that traces generated by the EFSM are well formed and always agree with π as to which accesses would be authorized. Second, manual techniques are used to generalize this result to countably infinite carriers. Intuitively, this generalization follows because the actions and authorizations of each user and object are independent of actions and authorizations of other users and objects. Third, manual techniques are again used to show that each action sequence that satisfies the well-formedness requirements of π traces can be generated by the EFSM (α-mapping in figure 4).

P2: Administration model specification—isolated case Administrative rights in the g-SIS context concern privileges to perform group operations, including creation and destruction of groups, permitting users to join and leave, and adding and removing objects. Many possible administrative models could be appropriate in various scenarios. Furthermore, it seems likely that existing authorization systems would satisfy the needs of many deployments. For example, one natural model that seems likely to be widely useful would allow anyone to create a group and to make an RT role [35] owned by that role confer administrative rights on role members. The use of RT would conveniently support delegation of administrative rights, as well as separating various administrative rights by associating them with different RT roles. However, new administrative models may be desirable for certain information sharing scenarios. For instance, multiple organizations may form an isolated group and share their users and objects. New objects (representing intellectual property) may be created in the group during the course of collaboration. Admin models should be able to specify a policy such as newly created objects, if exported out of the group, should be provided to every participating organization as they are equal stake-holders. The main research activity in this task will be to explore design choices and select administrative models that control group membership.

P3: Defining ideal stateless specification—connected case Informed by the requirements of case studies, this task will design inter-group relationships. For example, as mentioned in the introduction, membership in one group might confer rights to access objects in other groups, or membership in one group may be contingent on membership in another. In the former case, we say that the latter groups are *subordinate* to the first group; in the latter, we say that membership in the first group is *conditional* on membership in the second. Relationships such as these are transitive in their effect. For instance if membership in group g_3 is conditional on membership in group g_2 and membership in g_2 is conditional on membership in g_1, then membership in g_3 is, in effect, conditional on membership in g_1. Note that if group g_4 is subordinate to group g_5, users in g_5 do not gain direct membership into g_4. Instead, g_5 users' "sessions" receive some form of access to objects in g_4, provided other conditions are satisfied. Variations of subordination include read-only and read-write, which determine the mode of object access enabled by this relationship.

For this task, the PI will collaborate extensively with the Center for Infrastructure Assurance and Security (CIAS) at UTSA (see the collaboration commitment from Dr. Greg White, Director of CIAS). The CIAS has gained a national reputation through its efforts to help states and communities establish cross-sector cyber security programs including information sharing. Recently, CIAS has partnered with the Retail Cyber Intelligence Sharing Center (R-CISC) to coordinate its information sharing programs. R-CISC is a nonprofit organization established to support the retail and commercial services industries as the resource for sharing cybersecurity information and intelligence. R-CISC members include retail merchandising industry, restaurant or food service industry, sports leagues, gaming organizations, etc.

The PI will consult with CIAS's information sharing activities for developing concrete use-cases that can inform the model design for connected g-SIS. The PI strongly believes that CIAS's activities in this arena can help shape the models such that it has a practical relevance. Much like the core properties for g-SIS$^{\text{i}}$, this task will investigate core properties for g-SIS$^{\text{c}}$. The transient nature of groups require a careful analysis of subtle information flow properties. For example,
$\Box(\mathrm{possibleFlow}(g1, g2) \leftrightarrow \exists g3.(\mathrm{directFlow}(g1, g3) \wedge (g3 = g2 \vee \Diamond \mathrm{possibleFlow}(g3, g2))))$ where,

$$\begin{aligned}\mathrm{directFlow}(g1, g2) = ((&\neg \mathrm{destroyG}(-, g1)\, \mathcal{S}\, \mathrm{createG}(-, g1)) \wedge (\neg \mathrm{destroyG}(-, g2)\, \mathcal{S}\, \mathrm{createG}(-, g2)) \wedge \\ &(\neg \mathrm{insubordinate}(-, g1, g2)\, \mathcal{S}\, (\mathrm{subordinateRO}(-, g1, g2) \vee \mathrm{subordinateRW}(-, g1, g2))))\end{aligned}$$

Formula directFlow states that a direct flow of information can occur from group $g1$ to $g2$ only if they simultaneously exist (that is, the groups have not been destroyed since they were created) and $g1$ is read-only or read-write subordinate to $g2$. However, information can flow indirectly from $g1$ to $g2$ over a period of time via another(other) group(s). For instance, information can flow from $g1$ to $g2$ via $g3$ even if $g1$ and $g2$ never existed simultaneously. Formula possibleFlow above states that information can possibly flow from g1 to g2 if a direct flow can occur to some simultaneously existing group $g3$, and in the future it can flow from $g3$ to $g2$. Note that this flow can occur over multiple groups ultimately flowing into $g2$ and hence the need for recursion. (($\Diamond$ p) means that p is true at current or some future state.) Note that this project does not address covert information flow issues which are known to be thorny issues in this discipline. The properties that will be investigated concern constraining overt information flow, enabled by administrative relationship establishment operations.

The goal of this task is to identify a wide-variety of inter-group relationships based on case studies and interactions with CIAS, develop stateless specifications of g-SIS$^{\text{c}}$ models and core properties, and conduct formal analysis of these properties such as verifying their consistency, and independence. The PI will build on his experiences in integrating g-SIS with relationship-based access control models [19], which contain both relationship- and history-based elements, while at the same time, the model is shown to be reducible to a purely relationship-based one.

P4: Defining and verifying ideal stateful specification—connected case Showing correctness in this case is much more challenging than it is in the isolated case. In particular, it will be much harder to show that results obtained by model checking with a small number of groups generalize to systems having an unbounded number of groups. The result will be particularly challenging to obtain if subordination and conditionality relationships are dynamic, as we would want them to be. Intuitively, the situation in this case is very different from that in the isolated case because a user's access to an object in a given group can depend on relationships between an unbounded number of other groups. The notion of correctness that we can obtain may have to be weaker than in the isolated case, if we allow inter-group relationships to be fully dynamic. For instance, it may be that we can show only a soundless result that each trace generated by the EFSM satisfies the stateless specification. This kind of compromise is commonplace when the model-checking technique of abstraction is applied. One approach to abstraction that is likely to be helpful in our situation would be based on three-valued logic [33].

P5: Administration model specification—connected case The connected case introduces additional group operations that do not arise in the isolated case. This indicates the need for new admin models. For instance, if one group is made subordinate to another, it may be that the administrators of both groups must take this action in a coordinated manner. Furthermore, administrative policies should be dynamic to adjust to changing inter-group relationships over a period of time. Following the example above, if additional groups are to be added to the subordination chain it may be that the administrators of every group in the chain must take this action in a coordinated manner. Alternatively, this could be just the administrator of the most dominating group in the chain. Similar scenarios arise in other forms of inter-group relationships. Evidently, usable admin models design becomes more complicated when different types of relationships interact over a period of time. (Imagine a scenario when a vertical subordination chain intersects with a horizontal conditional chain. Who is responsible for specifying membership and relationship constraints in such a scenario? What if mutual exclusion relationship is brought into the picture?) Again, the main activity in this task is to design appropriate admin models.

C.4.2 E-layer Research Tracks

E1: Defining enforcement policy—isolated case This task will investigate the E-layer properties that will allow one to safely approximate ideal stateful policies.

Following the stateful policy discussed in beginning of section C.4, the timeline in figure 5 illustrates how staleness arising due to the mismatch in attribute values as compared to that maintained at a distributed policy information points (PIP) can lead to critical access violations. RT refers to Refresh Time, which is the time at which a policy decision point (PDP) synchronizes attributes with the PIP. The local copy of attributes maintained by the PDP may not be up to date during the time between two RT's. Thus access control decisions made by the PDP during this period are based on potentially stale attributes. In the figure, after RT_3, it is possible that the user is removed from the group. In this context, allowing access to $o1$ is a less of a problem as compared to allowing access to $o2$. This is because the latter clearly violates the Authorization Provenance property, where $u1$ and $o2$ never had a simultaneous period of membership.

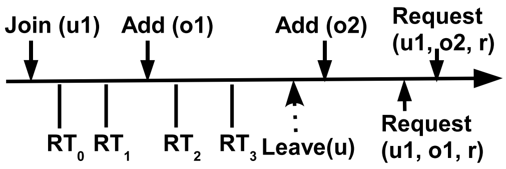


Figure 5: Staleness leading to access violation.

Formal Specification of Stale-Safety. The stale-safe property we specify requires that a requested action be performed only if a refresh of authorization information has shown that the action was authorized at that time. This refresh is permitted to have taken place either before or after the request was made. The last refresh must have indicated that the action was authorized, and all refreshes performed since the request, if any, must also have indicated the action was authorized. Let us introduce two formulas formalizing pieces of stale-safe security property. Intuitively, α_1 can be satisfied only if authorization was confirmed prior to the request, while α_2 can be satisfied only if authorization was confirmed after the request. Formally,

$$\alpha_1 = \ominus\, (\neg \mathrm{perform} \wedge (\neg \mathrm{RT} \vee (\mathrm{RT} \wedge \mathrm{Authz})))\, \mathcal{S}\, (\mathrm{request} \wedge (\neg \mathrm{RT}\, \mathcal{S}\, (\mathrm{RT} \wedge \mathrm{Authz})))$$

$$\alpha_2 = \ominus\, (\neg \mathrm{perform} \wedge \neg \mathrm{RT})\, \mathcal{S}\, (\mathrm{RT} \wedge \mathrm{Authz} \wedge ((\neg \mathrm{perform} \wedge (\neg \mathrm{RT} \vee (\mathrm{RT} \wedge \mathrm{Authz})))\, \mathcal{S}\, \mathrm{request}))$$

Note that α_2 can be satisfied without an authorizing refresh having occurred prior to the request, whereas α_1 cannot. Thus, though α_2 ensures fresher information is used to make access decisions, it does not logically entail α_1 as it is satisfied by traces that do not satisfy α_1.

Definition C.1 An enforcement model has *weak stale-safety* if: $\Box(\mathrm{perform} \rightarrow (\alpha_1 \vee \alpha_2))$.

Definition C.2 An enforcement model has the *strong stale-safety* if: $\Box(\mathrm{perform} \rightarrow \alpha_2)$.

The enforcement policies will address issues including out-of-sync authorization attributes distributed at the policy information point (PIP) and the policy decision/enforcement points (PDP/PEP) due to network delays, locally cached stale information, and off-line access. A novel aspect of this work on enforcement is the way we handle the fact that the above policy will not yield authorization decisions that correspond precisely with those of the ideal policy, rather it will manage this discrepancy to different degrees. We plan to use FOTL to specify the enforcement policies, enabling us to reason about them in the next task, E2.

E2: Defining and verifying enforcement models—isolated case In this task, EFSM-based enforcement models is defined, and model checking is used to formally verify that they adhere to the corresponding enforcement policies. The model is specified as a composition of EFSMs [37, 38] that represent the behaviors of PIPs, group admins, and PDPs, respectively. EFSMs in an enforcement model can execute concurrently, and synchronize and communicate with each other to ensure compliance with the FOTL enforcement policy. Determining satisfaction of an FOTL policy can have high complexity, leading to difficulty in the assessment of whether an EFSM's behavior is admissible. This research will combine automated proof methods (e.g., model checking) and manual proofs to verify that the EFSM-based model enforces the FOTL specification. Like earlier, manual proofs will utilize extending small model verification approach.

E3: Defining enforcement policy—distributed case This task will design enforcement policies for the connected groups developed in task P4. These policies will regulate authorization decisions at PEPs based on the views they receive of the current authorization states and relationships among groups in essentially the same way that decisions are made in the stateful model designed in P4. In the context of multiple connected groups, we will have many more forms/sources of information about the authorization states than in the isolated case. Information about group subordination, conditional relationships among groups, membership in enabling groups, mutual exclusion, etc. will need to be collected about various groups from various locations. Thus, stale-safety is a complex issue in the context of multiple PIPs and multiple groups. For instance, suppose, the user authorization view is held in PIPu while the object authorization view, in PIPo. Then the PEP needs to periodically synchronize with PIPu and PIPo. The authorization view from PIPu may change before synchronizing authorization view with PIPo (or vice-versa) leading to an instability of authorization views. One way to ensure stability is to require that at the time of synchronization with PIPo, the object authorization view has not changed since the user authorization view changed (which is evident from previous synchronization with PIPu). This requires time-stamping authorization view synchronization times, as well as authorization view-change times at PIPu and PIPo. Another interesting stale-safety issue arises due to changing inter-group relationships. Administration of such changes is likely to be highly distributed, which poses challenges in characterizing stale-safety. Synchronization of authorization views now involves keeping track of membership changes, subjects executing in different groups, etc. For instance, when a subordination or a conditional membership relation between two groups is removed, how do we efficiently yet safely enforce termination of subjects executing in other groups? When we consider systems in which there is no centralized authority that holds the view given by the ideal policy, the relationship between the view held by the PEP and the ideal policy is harder to make precise. Achieving this precision is a significant objective.

Note that different kinds of stale-safe properties can be designed for g-SIS^c depending on the semantics of inter-group relationships. For instance, if a user is known to be a member of a particular group at a particular time, and the membership in that group is conditional on that of another group, then authorization to access objects in the former group would also imply simultaneous authorization in the latter. A number of such interesting properties will be investigated in this task.

E4: Defining and verifying enforcement models—distributed case In this task we will construct enforcement models and demonstrate that they safely approximate corresponding enforcement policies. Each system will consist of components that manage attribute information associated with groups, users, subjects, and objects. The system will also include components that are PDPs, or PEPs. The model has the following characteristics: 1. The authorization states are frequently changed due to the dynamic nature of collaboration and sharing among groups; 2. The authorization information is controlled by distributed points; 3. The authorization decision is made upon the collaborative information from various points. Because the model as stated above can be quite large and complex, the naive application of model checking will not suffice to verify these requirements. In particular, reachability via group relations must be considered when handling both subordination and conditional groups. We plan to utilize a combination of methods, including model checking, automated abstraction techniques based on 3-valued logic [33], small model theorem [53], and manual approaches to reason about our enforcement model.

C.4.3 I-layer Research Track

This task concerns a complete implementation of the g-SIS models in a cloud-based platform as a proof-of-concept, and for the purpose of evaluation. This implementation will be carried out on a large-scale OpenStack-based cloud infrastructure that is available to the PI at UTSA (see Facilities and Equipment).

Investigate g-SIS implementation in OpenStack. This task will enhance the existing access control mechanism of OpenStack with g-SIS^c capability. There are compelling reasons for choosing OpenStack. (1) OpenStack is a robust open-source IaaS software for building public, private or hybrid clouds that is developed and maintained by a vibrant community with participation from more than 200 world-leading organizations with a release cycle of 6 months. (2) UTSA has strong collaboration with engineers from Rackspace, a leading IaaS provider headquartered in San Antonio (see the letter from the department chair). Rackspace is an original founder of OpenStack along with NASA and utilizes OpenStack for its commercial cloud service offerings. (3) The native constructs of cloud such as tenants are highly suitable to evaluate g-SIS. Since a tenant represents an information sharing organization, it is easy to simulate such organizations as tenants in OpenStack. The tenant construct also lends itself to create new groups dynamically.

As mentioned earlier, a unique aspect of this implementation is to build the STIX-TAXII-CybOX framework (see section C.1) for cyber threat information exchange between different organizations. The Trusted Automated eXchange of Indicator Information (TAXII) defines a standard set of services and message exchanges that enables participants to control what information is shared, and with which other participants that information is shared. The Structured Threat Information eXpression (STIX) defines a standard language for representations of cyber threat information. The Cyber Observable eXpression (CybOX) specifies a standard for cyber observables such as a change in the operating system registry key value. CybOX is used within STIX to describe cyber observables.

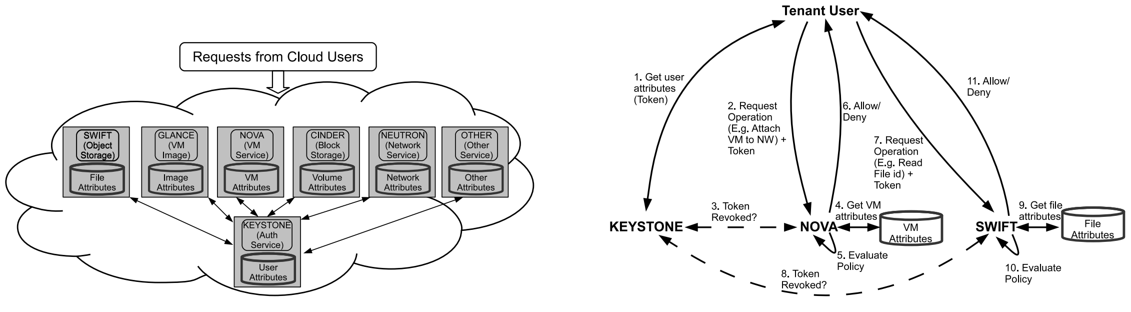
Different models of information sharing will be deployed and evaluated in this testbed. For example, TAXII can be deployed using architectures such as hub and spoke, source/subscriber and peer-to-peer. These architectures specify how information is shared between different participating organizations, and which organization act as a clearinghouse. The OpenStack-based platform will allow us to extensively test a wide variety of architectures and use cases for the information sharing specifications.

Figure 6(a) illustrates various service components of OpenStack. The entities in OpenStack include Swift (object storage), Glance (VM image storage), Nova (VM scheduler and overall compute manager), Cinder (block storage), Quantum (networking), etc. Each of these entities are responsible for managing the respectively indicated types of virtual resources. Finally, Keystone provides the identity and authorization information service for all the internal services. Figure 6(b) illustrates the simple enforcement model of OpenStack. Tenant users authenticate with Keystone which provides a token which is then forwarded by the user to the respective component from which a service is needed. The token contains all the attribute information that is necessary to make a decision. In OpenStack, each service component has its own PDP and PEP which use the token to make its decisions. Evidently, this implementation task will impact the Keystone and relevant service components (e.g. Swift) of OpenStack. Currently, OpenStack does not support API-based dynamic creation of tenants. Further, by default, tenants are strongly isolated from each other in OpenStack. The PI has conducted initial investigations in enabling cross-tenant sharing [51].

C.5 Research Evaluation Plan

This section discusses the complete evaluation of specifications developed in various tasks.

Design Adequacy: A critical success factor for this project is the design adequacy of the g-SIS^c models that are developed. This will be evaluated using real-world use-cases obtained in collaboration with CIAS. As mentioned earlier, CIAS conducts information sharing exercises across the nation and works with many organizations. This evaluation will also provide a critical feedback loop that will influence the design of the models themselves. Design adequacy at all levels (P/E-layers) of this project will be rigorously investigated



(a) OpenStack Service Components

(b) OpenStack enforcement.

Figure 6: OpenStack Components and Enforcement.

throughout the life of this project. Another aspect of adequacy is the feasibility of easy implementation of E-layer g-SIS^c by a typical programmer. This will be evaluated by hiring groups of undergraduate students and proving these specifications as a project in independent study courses. Following this, consistency between the EFSMs and the logic in the code that the students implemented will be evaluated.

Performance: This includes 2 aspects: (1) the ability of the model to efficiently satisfy the principles specified in section C.4.1, and (2) the performance of the g-SIS systems in the cloud platform under realistic loads. The former will be evaluated using simulated dataset where a large number of groups and combinations of inter-group relationships will be generated. All possible queries will be evaluated against the designed algorithms. The latter will be evaluated using a simulated workload in the cloud platforms. Multiple tenants can be created in a test platform, and information sharing operations can be simulated using the Swift service in OpenStack. Performance on such workload will be evaluated and fine-tuned.

Usability: This is a make or break metric for this project's success. For g-SIS^c to be usable, the interface for configuring groups with the intended effect should be intuitive. One approach planned is to offer the interface designed in the OpenStack platform to Amazon Mechanical Turk users, and ask them to perform specific configuration activities that will achieve the intended effect. Another lighter-weight approach is to conduct a similar experiment with a simulated tool, instead of the implemented software in cloud. The users will be provided concrete scenarios, and will be asked to perform specific operations to achieve a specific effect. For example, given a specific snapshot of inter-group relationships, the users can be asked to perform configuration operations so that a particular group may obtain access to a specific set of objects. The results of these experiments will be used to further improve the design interfaces and abstractions.

Practical Adoption: Given that STIX-TAXII is being actively explored and utilized by many public sector organizations, the PI will work with CIAS to explore the adoption of the g-SIS based, STIX-TAXII enhanced, OpenStack in real-world applications. For instance, the PI will actively explore the ties with R-CISC for adoption of the platform for testing current information sharing amongst retail organizations.

C.6 Integration of Research and Education

As confirmed by the Chair of the PI's department, the department seeks to excel in research and education in cybersecurity and cloud computing (letter of commitment attached). This aligns very well with the PI's motivation to develop a strong curriculum on theses topics. We discuss the educational goals and the planned educational activities supporting each respective goal below.

1. Launch undergraduate students onto entrepreneurial pathways. The PI strongly commits to mentoring undergraduate students in entrepreneurial activities. The Center for Innovation and Technology Entrepreneurship (CITE) at UTSA is an NSF-designated Innovation Corps (I-Corps) site. The primary goal of this NSF I-Corps site is to foster entrepreneurship that will lead to the commercialization of technology. CITE hosts biannual technology venture competitions for students, and supports promising student-faculty startups via the Roadrunner Incubator and New Venture Incubator on campus. Local business partnerships, including companies like Rackspace Hosting make this a vibrant ecosystem to nurture fledgling technology startups. CITE's primary partner for the NSF I-Corps program is the Texas Research & Technology Foundation

(TRTF), a nonprofit organization dedicated to building San Antonio's technology economy. Building on this project, the PI plans to (see collaboration commitment from the Assistant Director of CITE): (a) mentor undergraduate students in the biannual CITE competitions by providing the necessary training in the development of cutting edge cybersecurity technology during their capstone design projects that will better prepare them for their professional careers, (b) mentor graduate and undergraduate researchers in his laboratory and enable the generation of intellectual property in terms of patents, and successive licensure of such technology concerning SIS, and (c) work with student groups on topics relating to this project with the potential to be translated to the marketplace, and mentor such student teams through the I-Corps ecosystem, including national I-Corps workshops and competitions, which will enable the entrepreneur-scientists of tomorrow to receive hands-on training.

2. Strive to enhance participation of students from underrepresented groups. San Antonio is a predominantly Hispanic city and UTSA is a research-intensive, Hispanic Serving Institute with more than 60% of the students from traditionally underrepresented groups in science and engineering fields. Many of these are first-time college-goers in their family. The first educational activity toward this goal is to work with UTSA's Center for Excellence in Engineering Education (CE3). The center offers many workshops and mentors many students from underrepresented groups. In collaboration with the center director Dr. Mehdi Shadaram (letter of collaboration attached), the PI plans to offer many workshops on technical topics such as programming and gaming to excite initial interest in students. Subsequently, the PI will recruit many undergraduate students from the center for an independent study course on security and cloud related topics, where the PI will have the opportunity to mentor those students one-on-one.

Another goal is to educate high-school students on how computer security impacts many aspects of their day to day activities and why security matters. The PI plans to collaborate with the Interactive Technology Experience Center (iTEC) at UTSA for this purpose (see the letter from Dr. Heather Shipley, the Director of iTEC). The iTEC receives a huge number of students from underrepresented groups given the demography of San Antonio. The PI plans to conduct 1-day spring break camps for high-school students every year. Since most students use smartphones, the PI will provide seminars on security and privacy policies concerning smartphones, and the challenges involved in designing such policies. For instance, the PI will exemplify this problem by explaining the design challenges and risks involved when apps specify the permissions that they need at the time of install.

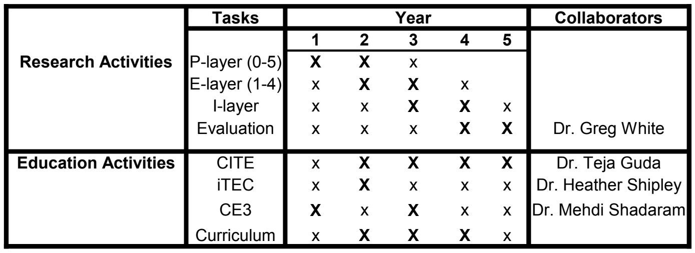
3. Develop new curriculum on security at UTSA. In the PI's department, there is currently one course relating to security that was developed by the PI: "EE 5453: Introduction to Computer and Network Security". Building on this project, the PI will introduce students to formal methods and its applications to security policy design, initially using this current course. Over the course of this project, the PI would like develop a dedicated course on topics of security policy design and enforcement, and cloud computing and security using OpenStack as an educational platform. The cloud courses will encompass access control issues, security configuration of the whole system, including cryptographic protections, firewall configurations, etc. The PI will build on his preliminary investigation on curriculum design for cloud computing [30].

Evaluation: The Pi will conduct both formative (ongoing) and summative (outcome oriented) evaluation [20] of the proposed educational activities so as to assess if they meet the stated goals. Goal (1) will be evaluated based on the impact the PI has had in the number of entrepreneurial pursuits over multiple years. Several metrics will used for this evaluation, including the number of teams that work with the PI, the competition success to failure ratio, number of successful start-up pursuits, etc. Goal (2) will be evaluated based on public data available from UTSA's CE3 and iTEC. For instance, out of the students who participate in the CE3 workshops, we can quantify the number of students who pursue their degree in computing before and after the PI conducts those workshops. The iTEC camps can also be evaluated by tracking participation rate year after year. Furthermore, simple verbal questions can be posed during the camps to evaluate whether current participants' "awareness" of security issues increases from year to year. Goal (3) is straight-forward to evaluate. The success criteria is the successful adoption of the curriculum by the UTSA community.

Formative evaluation is more critical for this goal so that the implementation and progress can be adjusted from year-to-year. Note that these tasks do not involve collecting individual data from human subjects.

C.7 Project Timeline

The detail project schedule is illustrated in figure 7. The tasks are split into research and educational activities. An ‘X’ represents main focus during that particular year, while ‘x’ represents secondary focus. The PI will be supported by 1 graduate student and 1 undergraduate student in this project.



	Tasks	Year					Collaborators
		1	2	3	4	5	
Research Activities	P-layer (0-5)	X	X	x			
	E-layer (1-4)	x	X	X	x		
	I-layer	x	x	X	X	x	
	Evaluation	x	x	x	X	X	Dr. Greg White
Education Activities	CITE	x	X	X	X	X	Dr. Teja Guda
	iTEC	x	X	x	x	x	Dr. Heather Shipley
	CE3	X	x	X	x	x	Dr. Mehdi Shadaram
	Curriculum	x	X	X	X	x	

Figure 7: Project Schedule.

C.8 Broader Impacts of the Proposed Work

Both the research and educational activities of this project will have a significant broader impact. With over 20 PhD programs, UTSA is one of the very few major institutions that is a *Hispanic Serving Institution*, a *Minority Institution*, and which is also a *research-intensive* institution. UTSA is Carnegie-recognized for Community Engagement and High Research Activity. As discussed in the education plan, the PI will seek all such avenues to recruit underrepresented students to participate in this project. The project will train students, particularly at the PhD level, in an area of national interest, enhance their careers, and contribute to their professional growth. The planned entrepreneurship activity will create great opportunities for such underrepresented students to pursue ground-breaking pathways in their careers.

Secure Information Sharing of cyber incident threat information between and among private and public sectors is being actively pursued in the U.S. This requirement was recently signed-in as an Executive Order by the current U.S. President. Thus this research project directly addresses a problem of critical importance to the nation and society and will have academic, community, and industrial impact. The techniques and tools that will be generated will benefit security applications in many areas, including national defense in cyber space, financial and banking systems, and community cyber security. The PI will disseminate the results by publishing them in high-quality security and formal methods conferences and journals at national and inter-national venues. The project will produce techniques and tools that support the design, analysis and maintenance of security policies and their enforcement, which will increase the capability of administrators and stake-holders to understand and improve the level of assurance they have of their systems’ and communities’ security. The software developed in this project will be made freely available to the public. The proposed implementation of g-SIS using the STIX-TAXII framework in the OpenStack cloud computing platform will be made open to researchers and students. Given the presence of the 24th Air Force (the cyber space combat forces of the United States Air Force) in San Antonio and the CIAS’s capacity to guide local communities in terms of cyber security preparedness, the PI strongly believes that this project will have major impact at the local, and potentially national level.

C.9 Results from Prior NSF Support

R Krishnan is the PI of CNS–1423481, “TWC: Small: Attribute Based Access Control for Cloud Infrastructure as a Service”, $500,000, 10/01/2014-09/30/2017. The goal of this project is to develop a theory of ABAC models in the context of resource administration and management across different tenants in an IaaS cloud. The project began in October 2014. Initial ABAC models have been developed and implementations in OpenStack cloud operating system is underway. One PhD student, Khalid Bijon, funded partially by this project, graduated in Spring 2015. Khalid currently works for MosaixSoft, a cloud startup company in Los Altos, CA. So far, this grant has fully/partially helped produce 5 conference papers [11, 12, 13, 23, 43], 2 workshop papers [42, 51], 1 poster [52], and 1 PhD dissertation [10].

D References Cited

References

[1] The center for infrastructure assurance and security. http://cias.utsa.edu/. [Online; accessed July, 2015].

[2] Department of homeland security. http://www.dhs.gov/. [Online; accessed July, 2015].

[3] Executive order – promoting private sector cybersecurity information sharing. http://go.wh.gov/SHrGEM. [Online; accessed July, 2015].

[4] Heroku cloud application platform. https://www.heroku.com/. Accessed: July 2015.

[5] Information sharing specifications for cybersecurity. https://www.us-cert.gov/Information-Sharing-Specifications-Cybersecurity. [Online; accessed July, 2015].

[6] The MITRE corporation. http://www.mitre.org/. [Online; accessed July, 2015].

[7] Openstack. https://www.openstack.org/. [Online; accessed July, 2015].

[8] Bowen Alpern and Fred B Schneider. Recognizing safety and liveness. *Distributed computing*, 2(3):117–126, 1987.

[9] V. Atluri and J. Warner. Automatic Enforcement of Access Control Policies Among Dynamic Coalitions. *International Conference on Distributed Computing and Internet Technology, Bhubaneswar, India, Dec*, 2004.

[10] Khalid Bijon. *Constraints for Attribute Based Access Control with Application in Cloud IaaS*. PhD thesis, University of Texas at San Antonio, 2015.

[11] Khalid Bijon, Ram Krishnan, and Ravi Sandhu. Mitigating multi-tenancy risks in iaas cloud through constraints-driven virtual resource scheduling. In *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies*, SACMAT '15, pages 63–74, New York, NY, USA, 2015. ACM.

[12] Khalid Bijon, Ram Krishnan, and Ravi Sandhu. Virtual resource orchestration constraints in cloud infrastructure as a service. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 183–194, New York, NY, USA, 2015. ACM.

[13] KhalidZaman Bijon, Ram Krishnan, and Ravi Sandhu. A formal model for isolation management in cloud infrastructure-as-a-service. In ManHo Au, Barbara Carminati, and C.-C.Jay Kuo, editors, *Network and System Security*, volume 8792 of *Lecture Notes in Computer Science*, pages 41–53. Springer International Publishing, 2014.

[14] A. Cimatti, E.M.Clarke, F.Giunchiglia, and M. Roveri. NuSMV: A new symbolic model checker. *International Journal on Software Tools for Technology Transfer*, 2(4):410–425, 2000.

[15] Edmund M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Language and Systems (TOPLAS)*, 8(2):244–263, 1986.

[16] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. The MIT Press, 1999.

[17] Eve Cohen, Roshan K. Thomas, William Winsborough, and Deborah Shands. Models for coalition-based access control (cbac). In *SACMAT '02: Proceedings of the seventh ACM symposium on Access control models and technologies*, pages 97–106, New York, NY, USA, 2002. ACM.

[18] Jason Crampton, Wing Leung, and Konstantin Beznosov. The secondary and approximate authorization model and its application to bell-lapadula policies. In *Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies*, SACMAT '06, pages 111–120, New York, NY, USA, 2006. ACM.

[19] Philip W.L. Fong, Pooya Mehregan, and Ram Krishnan. Relational abstraction in community-based secure collaboration. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer &#38; Communications Security*, CCS '13, pages 585–598, New York, NY, USA, 2013. ACM.

[20] Joy Frechtling. The 2002 user-friendly handbook for project evaluation. *ERIC/NSF*, 2002.

[21] Eric Freudenthal, Tracy Pesin, Lawrence Port, Edward Keenan, and Vijay Karamcheti. drbac: Distributed role-based access control for dynamic coalition environments. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 411, Washington, DC, USA, 2002. IEEE Computer Society.

[22] Xin Jin. *Attribute-Based Access Control Models And Implementation In Cloud Infrastructure as a Service*. PhD thesis, University of Texas at San Antonio, 2014.

[23] Xin Jin, R. Krishnan, and R. Sandhu. Role and attribute based collaborative administration of intra-tenant cloud iaas. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*, pages 261–274, Oct 2014.

[24] Xin Jin, Ram Krishnan, and Ravi Sandhu. A unified attribute-based access control model covering dac, mac and rbac. In *Data and Applications Security and Privacy XXVI*, pages 41–55. Springer, 2012.

[25] Xin Jin, Ram Krishnan, and Ravi Sandhu. Reachability analysis for role-based administration of attributes. In *Proceedings of the 2013 ACM Workshop on Digital Identity Management*, DIM '13, pages 73–84, New York, NY, USA, 2013. ACM.

[26] Cliff B. Jones. Specification and design of (parallel) programs. In *IFIP Congress*, pages 321–332, 1983.

[27] H. Khurana and V.D. Gligor. A Model for Access Negotiations in Dynamic Coalitions. *Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE '04)*, pages 205–210.

[28] Himanshu Khurana, Virgil Gligor, and John Linn. Reasoning about joint administration of access policies for coalition resources. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 429, Washington, DC, USA, 2002. IEEE Computer Society.

[29] R. Krishnan, J. Niu, R. Sandhu, and W.H. Winsborough. Stale-safe security properties for group-based secure information sharing. In *Proceedings of the 6th ACM workshop on Formal methods in security engineering*, pages 53–62. ACM New York, NY, USA, 2008.

[30] Ram Krishnan and Eugene John. Design of a curriculum on cloud computing. *International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'13)*, July 2013.

[31] Ram Krishnan, Jianwei Niu, Ravi Sandhu, and William Winsborough. Group-centric secure information sharing models for isolated groups. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–29, 2011.

[32] Adam J. Lee and Marianne Winslett. Enforcing safety and consistency constraints in policy-based authorization systems. *ACM Trans. Inf. Syst. Secur.*, 12(2):1–33, 2008.

[33] Tal Lev-Ami and Shmuel Sagiv. Tvla: A system for implementing static analyses. In *SAS '00: Proceedings of the 7th International Symposium on Static Analysis*, pages 280–301, London, UK, 2000. Springer-Verlag.

[34] R. Levin, E. Cohen, W. Corwin, F. Pollack, and W. Wulf. Policy/mechanism separation in Hydra. In *5th ACM Symposium on Operating Systems Principles*, pages 132–140, 1975.

[35] Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130. IEEE Computer Society Press, May 2002.

[36] Bertrand Meyer. *Object-Oriented Software Construction*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[37] Jianwei Niu. *Template Semantics: A Parameterized Approach to Semantics Based Model Compilation*. PhD thesis, University of Waterloo, School of Computer Science, 2005.

[38] Jianwei Niu, Joanne M. Atlee, and Nancy A. Day. Template semantics for model-based notations. *IEEE Transactions on Software Engineering*, 29(10):866–882, October 2003.

[39] C.E. Phillips Jr, TC Ting, and S.A. Demurjian. Information sharing and security in dynamic coalitions. *Proceedings of the 7th ACM Symposium on Access Control Models and Technologies*, pages 87–96, 2002.

[40] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, volume 526, pages 46–67, 1977.

[41] A. Pnueli. In transition from global to modular temporal reasoning about programs. In *Logics and models of concurrent systems*, pages 123–144, New York, NY, USA, 1985. Springer-Verlag.

[42] Navid Pustchi, Ram Krishnan, and Ravi Sandhu. Authorization federation in iaas multi cloud. In *Proceedings of the 3rd International Workshop on Security in Cloud Computing*, SCC '15, pages 63–71, New York, NY, USA, 2015. ACM.

[43] QasimMahmood Rajpoot, ChristianDamsgaard Jensen, and Ram Krishnan. Integrating attributes into role-based access control. In Pierangela Samarati, editor, *Data and Applications Security and Privacy XXIX*, volume 9149 of *Lecture Notes in Computer Science*, pages 242–249. Springer International Publishing, 2015.

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

[45] Deborah Shands, Richard Yee, Jay Jacobs, and E John Sebes. Secure virtual enclaves: Supporting coalition use of distributed application technologies. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, volume 1, pages 335–350. IEEE, 2000.

[46] Mukesh Singhal, Santosh Chandrasekhar, Tingjian Ge, Ravi S Sandhu, Ram Krishnan, Gail-Joon Ahn, and Elisa Bertino. Collaboration in multicloud computing environments: Framework and security issues. *IEEE Computer*, 46(2):76–84, 2013.

[47] A Prasad Sistla. Safety, liveness and fairness in temporal logic. *Formal Aspects of Computing*, 6(5):495–511, 1994.

[48] Mahesh V. Tripunitara and Ninghui Li. A theory for comparing the expressive power of access control models. *J. Comput. Secur.*, 15(2):231–272, April 2007.

[49] J. Warner, V. Atluri, R. Mukkamala, and J. Vaidya. Using semantics for automatic enforcement of access control policies among dynamic coalitions. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 235–244, New York, NY, USA, 2007. ACM.

[50] Qiang Wei, Jason Crampton, Konstantin Beznosov, and Matei Ripeanu. Authorization recycling in hierarchical rbac systems. *ACM Trans. Inf. Syst. Secur.*, 14(1):3:1–3:29, June 2011.

[51] Yun Zhang, Ram Krishnan, and Ravi Sandhu. Secure information and resource sharing in cloud infrastructure as a service. In *Proceedings of the 2014 ACM Workshop on Information Sharing &#38; Collaborative Security*, WISCS '14, pages 81–90, New York, NY, USA, 2014. ACM.

[52] Yun Zhang, Ram Krishnan, and Ravi Sandhu. Secure information and resource sharing in cloud. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 131–133, New York, NY, USA, 2015. ACM.

[53] Lenore D. Zuck and Amir Pnueli. Model checking and abstraction to the aid of parameterized systems (a survey). *Computer Languages, Systems and Structures*, 30(3-4):139–169, 2004.