

A Distributed Requirements Management Framework for Compliance and Accountability

Travis D. Breaux and Annie I. Antón

North Carolina State University
Raleigh, North Carolina, USA
{tdbreaux, aianton}@ncsu.edu

Eugene H. Spafford

Purdue University, CERIAS
West Lafayette, Indiana, USA
spaf@purdue.edu

***Abstract:** Increasingly, regulations are requiring organizations to comply with the law and account for their actions. Individuals responsible for ensuring compliance and accountability currently lack guidance and support to manage their legal obligations within relevant information systems. Software controls provide assurances that business processes adhere to specific requirements, such as those derived from government regulations. We propose a requirements management framework that enables executives, business managers, software developers and auditors to distribute legal obligations across business units and/or personnel with different roles and technical capabilities. This framework improves accountability by integrating traceability throughout the policy and requirements lifecycle. We illustrate the framework within the context of a concrete healthcare scenario in which obligations incurred from the Health Insurance Portability and Accountability Act (HIPAA) are delegated and refined into software requirements. Additionally, we show how auditing mechanisms can be integrated into the framework and how auditors can certify that specific chains of delegation and refinement decisions comply with the intent of government regulations.*

1. Introduction

National and international standards, regulations and laws impose restrictions on business practices to achieve societal goals, such as improving corporate accountability in financial markets or ensuring the privacy of medical records in the healthcare industry. Mature standards and regulations describe specific personnel responsibilities that cut across several business units and require comprehensive documentation to demonstrate how personnel decisions implement standards and regulations. Furthermore, accreditation boards and government auditors impose penalties on organizations to motivate corrective action in the event of non-compliance. While organizations are often held accountable, recent legislation such as Sarbanes-Oxley¹ (SOX) and the Health Insurance

Portability and Accountability Act² (HIPAA) in the U.S. shift liability towards personnel, imposing fines and prison sentences on individuals for their actions that contribute to non-compliance.

Government regulations are consistently requiring a set of artifacts that demonstrate and account for personnel actions taken to comply with the law – the matter of *accountability*. Because business practices often include a significant human factor (e.g., people implementing policies in a potentially ad-hoc fashion), compliance with standards and regulations is complicated by pressures on human performance (e.g., increasing profits, decreasing costs). In large organizations, software systems that support these processes can provide increased compliance assurance by implementing software controls that restrict what actions personnel can perform with oversight under the law. In effect, software provides the toolset enabling business practices while limiting the improper use of resources that would otherwise violate the law.

Organizations are in need of mechanisms to help manage resources that provide assurances that operational practices comply with standards and regulations. According to a 2005 Ernst and Young survey of over 1,300 international organizations, compliance with regulations and policy surpassed worms and viruses as the primary driver of information security policy in 2005. To this end, we propose a distributed requirements management framework that provides a transparent and accountable method to ensure obligations in standards and regulations are implemented by functional software requirements and software controls. In our framework, personnel satisfy obligations by refining these obligations into functional requirements or by creating new obligations that are delegated to others. Through delegation and refinement, personnel will contextualize their obligations incurred from standards and regulations using their own business knowledge and goals. Recording the personnel decisions to delegate and refine obligations improves accountability, because each decision can be evaluated and compared against

¹ U.S. Public Law 107-204, 116 Stat. (2002)

² U.S. Public Law 104-191, 110 Stat. (1996)

best practices. Auditors can certify these decision chains to demonstrate that, at least at a specific point in time, organizations comply with the intent of the law. Furthermore, as policies change, organizations can re-evaluate their decisions to delegate and refine their legal obligations in a framework that dynamically dispatches these changes to personnel responsible for accommodating these changes.

In our previous work, we analyzed privacy policies in healthcare and finance [BA05a, BA05b], HIPAA regulations [BA05c, BVA06] and organizational security policies [BAK06] to identify policy elements necessary to align systems with standards and regulations. In each of these studies, we identified a need to manage obligations in a single, distributed framework. Our proposed framework builds upon this need by managing obligations through delegation and refinement with special focus on the needs of auditors.

This paper is organized as follows: in Section 2, we consider a simple scenario to motivate our framework; in Section 3, we present the framework formalism and definitions; in Section 4, we instantiate our framework by elaborating on an application in the healthcare domain; in Section 5 we discuss related work; in Section 6 we discuss requirements for a tool supporting our framework with our conclusion in Section 7.

2. Requirements Scenario

Consider a scenario in which a Chief Security Officer (CSO) has been assigned the high-level security goal (a non-functional requirement) NFR_1 = “to ensure that information is secure.” The CSO implements NFR_1 by assigning several new non-functional requirements including NFR_2 “ensure internal communications are confidential” to his IT security manager in charge of network security. The IT security manager responds by identifying all modes of “internal communications” relevant to satisfying her new obligation. As a result, the manager identifies internal web and e-mail servers among others that use TCP/IP network connections to share information between internal systems. The manager, with both authority over who administers these servers and knowledge of available security mechanisms in these systems, implements her obligation by assigning new functional requirements including FR_1 “ensure web servers use SSL for internal connections” and FR_2 “ensure mail servers use TLS for internal connections” to relevant system administrators across different departments. The system administrator responsible for administering a mail server running Linux receives FR_2 and implements the requirement with a series of configuration directives that he applies to the system: FR_1 = “install latest OpenSSL libraries,” FR_2 = “compile and configure Sendmail with TLS support”, FR_3 = “generate X.509 certificates for Sendmail,” etc.

At each level in the delegation hierarchy, a manager knows *what* goal their staff member must achieve but the manager may not have the technical knowledge to know *how* their staff will achieve this goal. Each obligation is owned by someone who is ultimately accountable for that obligation and the decisions to refine an obligation are also recorded. Tracing permissions and obligations through ownership, delegation and refinement are necessary to quickly and effectively identify how and why vulnerabilities are addressed to reduce risk of non-compliance.

3. Management Framework

Our proposed distributed requirements management framework provides traceability from the regulations that govern organizations to the decisions of actors that assign obligations to other actors and finally to the software requirements assigned to systems that refine those personnel obligations. Figure 1 shows the associations maintained in our framework between actors, systems and obligations. In assignment, also called delegation, each assigned obligation has exactly one owner (an actor) who is ultimately responsible for satisfying the obligation. The owner may assign the obligation to other actors and the owner may further permit those actors to re-delegate the obligation to others. With regard to refinement, an actor who has been assigned an obligation may choose to refine the obligation into other obligations, called *refinements*; satisfying these refinements contributes to satisfying the refined obligation.

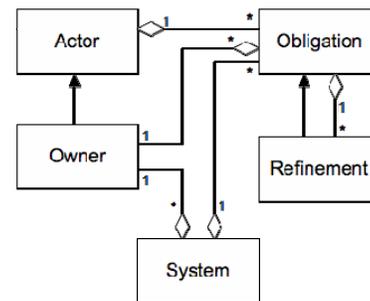


Figure 1: Framework Conceptual Model

Because the management framework seeks to coordinate between personnel and information systems, we distinguish between two types of obligations: *responsibilities* that require a person to perform some action and *requirements* that require a system to have some property or perform some function. For all systems, there is exactly one *administrative obligation* that requires exactly one actor to be ultimately responsible for implementing the requirements assigned to those systems. In some situations this actor may be permitted to re-delegate their administrative obligation for a specific system to other actors. In our

framework, we relegate the important issue of authorizing delegation and refinement decisions to related work in Section 5.

We now define key terms to elucidate the primary elements of our management model.

Definition 1: Management Model

Let the set A consist of actors, the set S consist of systems and the set O consist of obligations.

The *assignment set* $AS \subseteq (A \cup S) \times O$ is a many-to-many relation mapping actors and systems to their assigned obligations; each actor or system may have multiple obligations and each obligation may be assigned to multiple actors or systems.

The *delegation set* $DS \subseteq A \times AS$ is a one-to-many relation mapping actors (the delegator) to the obligations that they assign to actors and systems (the delegatee). To be consistent, we assume $\forall \langle a_i, a_j, o \rangle \in DS, \exists \langle a_j, o \rangle \in AS$. We also assume a permission framework is in place to ensure the delegator is authorized to delegate obligations to the delegatee.

The *ownership set* $N \subseteq A \times (O \cup S)$ is a many-to-one relation mapping actors to the obligations and systems they own; every obligation and system has exactly one owner. For each system $s \in S$ with owner a in $\langle a, s \rangle \in N$, there is one administrative obligation o in $\langle a, o \rangle \in AS$ that requires the owner a to satisfy all the system requirements r in $\langle s, r \rangle \in AS$.

The *refinement set* $RS \subseteq AS \times O$ is a many-to-one relation mapping actors or systems and their assigned obligations to refinements (other obligations). These refinements are declared by the actor assigned the obligation or by the system administrator. Multiple actors can refine a single obligation using different obligations depending on their view of the obligation in the context of their daily operations. For an actor or system $a \in (A \cup S)$ and an obligation $o \in O$, the set $\{r \mid \langle a, o, r \rangle \in RS\}$ is called a *refinement strategy*.

The *decision sequence* $\{d_1, d_2, \dots, d_n\} \subseteq (DS \cup RS)$ is derived by tracing an obligation through delegation and refinement decisions. A valid decision sequence is comprised of a series of delegation and refinement sequences: a delegation sequence $\{\langle a_1, a_2, o \rangle, \langle a_2, a_3, o \rangle, \dots, \langle a_n, a_{n+1}, o \rangle\} \subseteq DS$ traces the obligation o from the delegator a_1 to the delegatee a_{i+1} , and so on for $1 \leq i \leq n$; and a refinement sequence $\{\langle a, o_1, o_2 \rangle, \langle a, o_2, o_3 \rangle, \dots, \langle a, o_n, o_{n+1} \rangle\} \subseteq RS$ traces the obligation o_i assigned to the actor a to the refinement o_{i+1} , and so on for $1 \leq i \leq n$. A valid decision sequence begins with either a delegation or refinement decision $d_1 \in (DS \cup RS)$ and alternates between delegation and refinement sequences: a refinement sequence follows a delegation sequence by $\{\dots, \langle a_i, a_j, o_k \rangle, \langle a_j, o_k, o_l \rangle, \dots\}$ where the delegatee a_j refines obligation o_k into obligation o_l ; and a delegation sequence follows a refinement sequence

by $\{\dots, \langle a_i, o_k, o_l \rangle, \langle a_i, a_j, o_l \rangle, \dots\}$ where a delegator a_i refines an obligation o_k into an obligation o_l and delegates o_l to the actor a_j . We presently assume that no cycles exist in all sequences of delegations and refinements derivable from $(DS \cup RS)$. Refinement is complete for an obligation if it is accountable, which we now discuss.

An obligation is *accountable* if a mechanism exists to verify that the obligation has been satisfied [BAK06]. The mechanism may either be: an executable program that returns true, if and only if, the obligation is achieved or maintained; or the evaluation of a logical expression comprised of the conjunction of predicates. Each predicate denotes the satisfaction of obligations in a refinement strategy, each obligation of which must itself be accountable.

Consider, for example, the accountability of system requirements. Because *functional requirements* are testable, a program exists called a *test case* to verify whether a system satisfies the requirement – hence, functional requirements are always accountable. On the other hand, *non-functional requirements* are not testable, but they are accountable if they are refined into functional requirements. Therefore, testing non-functional requirements is tantamount to testing their refinements, assuming each refinement is either itself a functional requirement or another non-functional requirement that is refined into one or more other accountable requirements.

Personnel responsibilities may also be accountable within this framework, if they are supported by software systems that retain sufficient information to evaluate the responsibility using an executable program. For example, the responsibility “to only use a password that contains at least eight characters” is accountable by executing a program to check the length of a user’s personal password. However, the responsibility “to logout from a system when the system is no longer in use” is not accountable, as it is difficult to define the behavior of “in use” for all users, systems and applications.

Definition 2: Accountability

The *verification set* $VS \subseteq \text{Boolean} \times AS$ is a one-to-one mapping of Boolean predicates to assignments such that each predicate is true, if and only if the assigned obligation is satisfied by the actor or system in the assignment. For a verification $\langle v, a, o \rangle \in VS$, the predicate v evaluates to either: (1) true if an executable program called an *oracle* or *test case* that accepts some input data decides the obligation is satisfied or false otherwise; or (2) the logical conjunction of predicates assigned to some number of refinements: for the verification $\langle v, a, o \rangle \in VS$, let $v = v_0 \wedge v_1 \wedge \dots \wedge v_n$ such that $\langle v_i, a, o_i \rangle \in VS$ for all refinements $\langle a, o_i, o \rangle \in RS$. The expression must contain exactly those predicates

for refinements o_i that are necessary and sufficient to satisfy the obligation o .

4. Applying the Framework

We apply our framework from Section 3 to an example in which obligations (OB) from the HIPAA Security and Privacy Rules are delegated from upper management to their staff and later refined into software requirements (FR). At each delegation stage, an employee with specialized responsibility and technical expertise interprets their assigned obligations and refines and/or re-delegates these obligations, as needed. We illustrate this application by narrating the sequence of delegation and refinement decisions. At each state, we list the obligations followed by the expressions in our model that record these decisions.

The Chief Security Officer (CSO) for a healthcare provider (a covered entity) is assigned the following obligation from the HIPAA Security Rule (SR) §164.308(a)(2):

OB₁: Identify the security official who is responsible for the development and implementation of the policies and procedures required by the HIPAA section 164 subpart C for the covered entity.

$$\langle SR, CSO, OB_1 \rangle \in DS$$

$$\langle CSO, OB_1 \rangle \in N$$

The CSO identifies the SO which entails the delegation of obligations from the Security Rule §164.302–§164.318 to the SO, including:

OB₂: From §164.312(a)(1): Allow access only to those persons or software programs that have been granted access rights as specified in §164.308(a)(4);

OB₃: From §164.308(a)(4): Authorize access to electronic PHI (Protected Health Information) that are consistent with the applicable requirements of subpart E of this part (e.g., the Privacy Rule); and

OB₄: §164.308(a)(4)(ii)(B): Grant access to electronic PHI, for example, through access to a workstation, transaction, program, process, or other mechanism.

$$\{\langle SR, SO, OB_2 \rangle, \langle SR, SO, OB_3 \rangle\} \subseteq DS$$

$$\{\langle SR, SO, OB_4 \rangle\} \subseteq DS$$

$$\{\langle SO, OB_2 \rangle, \langle SO, OB_3 \rangle, \langle SO, OB_4 \rangle\} \subseteq N$$

Furthermore, for those authorizations in the Privacy Rule that participate in a transaction and utilize an “electronic communications network,” the SO must also implement technical measures to:

OB₅: From §164.312(e)(1): Guard against unauthorized access to electronic PHI that is

transmitted over an electronic communications network; and

OB₆: From §164.312(e)(1)(ii): Encrypt electronic protected health information whenever deemed appropriate.”

$$\{\langle SR, SO, OB_5 \rangle, \langle SR, SO, OB_6 \rangle\} \subseteq DS$$

$$\{\langle SO, OB_5 \rangle, \langle SO, OB_6 \rangle\} \subseteq N$$

Among the several authorizations in the HIPAA Privacy Rule (PR), we contrast the follow two authorizations to illustrate exceptions between rules; for example, authorization OB_8 explicitly excludes reports of child abuse, whereas, OB_7 permits disclosing such reports:

OB₇: From §164.512(b)(1)(ii): Disclose PHI to a government authority authorized by law to receive reports of child abuse or neglect.

OB₈: From §164.512(c)(1)(ii): Except for reports of child abuse or neglect permitted by paragraph (b)(1)(ii), disclose PHI about an individual whom the covered entity reasonably believes to be a victim of abuse, neglect, or domestic violence to a government authority authorized by law to receive reports of such abuse, neglect, or domestic violence... to the extent the disclosure is expressly authorized by statute or regulation and either: (A) The covered entity, in the exercise of professional judgment, believes the disclosure is necessary to prevent serious harm to the individual or other potential victims; or (B) If the individual is unable to agree because of incapacity, a public official authorized to receive the report represents that the protected health information contained in the disclosure is not intended to be used against the individual and that an immediate law enforcement activity that depends upon the disclosure would be materially and adversely affected by waiting until the individual is able to agree to the disclosure.

$$\{\langle PR, SO, OB_7 \rangle, \langle PR, SO, OB_8 \rangle\} \subseteq DS$$

$$\{\langle SO, OB_7 \rangle, \langle SO, OB_8 \rangle\} \subseteq N$$

$$\{\langle SO, OB_3, OB_7 \rangle, \langle SO, OB_3, OB_8 \rangle\} \subseteq RS$$

The SO delegates these obligations to an Information System Architect (ISA) who is responsible for external disclosures of PHI to business and government associates, third parties, etc.

$$\{\langle SO, ISA, OB_2 \rangle, \langle SO, ISA, OB_4 \rangle\} \subseteq DS$$

$$\{\langle SO, ISA, OB_5 \rangle, \langle SO, ISA, OB_6 \rangle\} \subseteq DS$$

$$\{\langle SO, ISA, OB_7 \rangle, \langle SO, ISA, OB_8 \rangle\} \subseteq DS$$

The ISA refines these obligations into the following functional requirements:

- FR₁**: The system shall identify users by role: one role per law that (1) authorizes a user to receive reports of child abuse or neglect; (2) authorizes a government authority to receive reports of other abuse, neglect or domestic violence.
- FR₂**: The system shall identify data by subsets: one subset per law designating which PHI may be disclosed to users authorized to receive reports of abuse, neglect or domestic violence.
- FR₃**: The system shall record individually identifiable testimony from : (1) the user, an employee of the covered entity, stating that they believe disclosing the PHI is necessary to prevent serious harm to the individual or other potential victims; or (2) the user receiving the PHI stating that the protected health information is not intended to be used against the individual and that an immediate law enforcement activity that depends on the PHI would be materially and adversely affected by waiting until the individual agrees to the disclosure.
- FR₄**: The system shall provide encrypted access to PHI identified in subsets (via FR2) only to users identified by roles (via FR1) only after receiving proper testimony (via FR4)

$$\begin{aligned} &\{\langle ISA, FR_1 \rangle, \langle ISA, FR_2 \rangle, \langle ISA, FR_3 \rangle\} \subseteq N \\ &\{\langle ISA, FR_4 \rangle\} \subseteq N \\ &\{\langle ISA, OB_2, FR_1 \rangle, \langle ISA, OB_4, FR_1 \rangle\} \subseteq RS \\ &\{\langle ISA, OB_6, FR_1 \rangle, \langle ISA, OB_5, FR_1 \rangle\} \subseteq RS \\ &\{\langle ISA, OB_7, FR_1 \rangle, \langle ISA, OB_7, FR_2 \rangle\} \subseteq RS \\ &\{\langle ISA, OB_8, FR_1 \rangle, \langle ISA, OB_8, FR_2 \rangle\} \subseteq RS \\ &\{\langle ISA, FR_4, FR_1 \rangle, \langle ISA, FR_4, FR_2 \rangle\} \subseteq RS \\ &\{\langle ISA, FR_4, FR_3 \rangle\} \subseteq RS \end{aligned}$$

The ISA surveys existing systems within the covered entity and assigns the four requirements FR_1 - FR_4 to relevant systems. If existing systems are unable to satisfy any requirements, those requirements are assigned to a Software Engineer who will design, develop and deliver a new system or configuration to meet these requirements.

4.1 Compliance and Accountability

Heterogeneity between business practices in different organizations makes it difficult to develop a single, de-facto implementation of standards and regulations to achieve compliance. Consequently, auditors and external reviewers must certify that a set of business practices comply with a set of standards or regulations at a specific point in time: a process called *certification*. In addition, auditors must acquire real

world evidence demonstrating that business practices continue to comply, either through random or continuous sampling: this acquisition is called an *audit*.

Using our framework, organizations can exhibit decision sequences that trace regulations to functional requirements. Each requirement is verified using one or more test cases to verify software systems. Auditors and external reviewers must certify that these sequences comply with law using a digital signature. The function $sign : K \times M \rightarrow S$ maps secret keys K and messages M to a unique signature in S . The signature can be used to verify that a message, in this case the decision sequence, has not changed. In follow-up reviews, decision sequences are re-certified using the history of verification predicates obtained during the policy and runtime requirements lifecycle.

Returning to the application of our framework in Section 4, the auditor identifies the following decision sequence $D = \{d_1, d_2, d_3, d_4\}$:

$$\begin{aligned} &d_1 = \langle SR, SO, OB_3 \rangle, d_3 = \langle SO, ISA, OB_7 \rangle \subseteq DS \\ &d_2 = \langle SO, OB_2, OB_7 \rangle, d_4 = \langle ISA, OB_7, FR_1 \rangle \subseteq RS \end{aligned}$$

The ISA exhibits the verification subset $V \subseteq VS$ consisting of verifications for all systems s in $\langle s, FR_1 \rangle \in AS$ and $\langle v, s, FR_1 \rangle \in V$. The auditors, deciding that the sequence D and verification set V are all necessary to comply with the regulations $\{OB_2, OB_3, OB_7\}$ at time t , uses their secret key k to provide the certificate $C = \langle sign(k, D + V + t), D, V, t \rangle$. In subsequent reviews at time $t' > t$, an auditor re-certifies these systems by verifying D and V against the previous certificate and, if the auditor accepts the result, the auditor will issue a new certificate updated for time t' . A series of these certificates for a single regulation at different time intervals is the compliance history for that regulation.

5. Related Work

Research in system security has shifted focus to include broader issues in organizational management. We see evidence for this shift in both access control frameworks limited to permissions [BS00, BDF02, OS02, PCN04] and more progressive approaches that also include obligations and delegation [DLS01, MS91, MU00, PS04]. Permissions describe what people and systems are permitted to do while obligations describe what people and systems are required to do given certain specific restrictions or constraints. In delegation, a person delegates their authority or responsibility to another person; the latter person acts on behalf of the former.

The proper authority to delegate obligations is important to ensure the standards and regulations can be properly refined into functional requirements. Barka and Sandhu consider modes of delegation including permanent and temporary delegation [BS00]. Revoking

delegation authority would first require evaluating the impact on obligations that were refined and/or re-delegated. Bandmann et al. propose constrained delegation as means to moderate delegated authority in a distributed system [BDF02]. Constrained delegation provides a means to control to whom a delegator can assign new obligations; this is important to prevent a delegator from obligating actors or systems that are beyond the delegator's scope of authority.

With regards to permissions, Oh and Sandhu use business units and organizational hierarchy to administer roles and permissions for users [OS02]. Park et al. attempt to align organizational structure with system structure to improve role-based access control implementations [PCN04]. These approaches highlight the organizational need to conceptually align existing authorization frameworks with organizational structure.

Moffett and Sloman introduced the concept of policies and system objects [MS91] which they later realized in the Ponder language [DDL01] to express authorizations and obligations for managing large networks. A deployment model for distributed network management was proposed using Ponder [DLS01]. Minsky and Ungureanu introduce Law-governed Interactions (LGI) in which actors suffer penalties if they violate their obligations [MU00]. In LGI, actors subscribe to a shared controller which audits their behavior to detect non-compliance. Park and Sandhu propose U_{CON}_{ABC} to manage authorizations and obligations using conditions for digital rights management [PS04]. Each of these approaches shares common elements relevant to regulatory compliance, including the ability to express permissions, obligations and delegations and the means to audit compliance through obligations. Moffett identified the need for requirements in policy models [M99]. For regulatory compliance, the refinement decisions of actors are also needed to completely trace from regulations to the requirements of systems that satisfy those regulations.

In requirements engineering, related work has focused on goal refinement and delegation [Ant97, AMP94, BLM04, DLF93, DDM97, GMM05]. Goals describe desired states or actions performed by actors without specific consideration for normative positions (e.g., permissions, recommendations and obligations.) Similar to obligations, goals can be decomposed into sub-goals intended to achieve the original goal [AMP94]. Darimont et al. describe the GRAIL tool [DDM97] that implements the KAOS framework [DLF93] for modeling goal refinement hierarchies using logical and/or relations and temporal logic. Regulatory compliance exacerbates the distributed and collaborative environment in which obligations are refined by personnel throughout an organization. While the GRAIL tool goes far to address the rich semantics of goal refinement, it does not address the broader

traceability issue where individual personnel decide how and when to refine obligations. Antón et al. show how non-functional requirements can be implemented through functional requirements [Ant97]; a notion we capture in our definition of refinement. Mylopolous et al. propose the Secure Tropos framework for modeling ownership and delegation that defines obligations as “trust in execution” [GMM05]. Similar to GRAIL, Secure Tropos provides a single-user perspective on goal refinement, whereas our compliance framework incorporates multiple viewpoints through distributed refinement. Bandara et al. propose applying goal refinement to policies using Event Calculus for temporal reasoning [BLM04]. While goal refinement is not new, tracing refinement and delegation of obligations, together, in a distributed environment that supports audits and external reviews provides new opportunities to explore compliance-related issues.

Rees et al. describe an information security framework named PFIREs that combines policy assessment and review to mitigate risk in organizational security [RBS03]. The framework does not address policies as system objects, per se. However, they propose improving security by passing messages between personnel to communicate obligations and monitor compliance; these messages are necessary to implement a compliance framework such as ours.

Finally, several publications on policy, including the NIST Security Handbook [80012], define policies as comprised of standards, guidelines and procedures. *Standards* are implementable obligations assigned to personnel and systems whereas *guidelines* are recommendations that may coincide with best practices for specific contexts. *Procedures* implement standards, often with a step-by-step description that is sufficient for other actors to reproduce the desired results. In our framework, these notions are complementary and can be easily supported. Guidelines require distinguishing the modality of recommendations (should) from obligations (must, shall) and “a procedure” requires conditionally sequencing a set of refinements. Despite such extensions, it is important to emphasize that standards, guidelines and procedures typically come from upper management, whereas our framework specifically supports middle and lower managers in their effort to contextualize organization-wide goals as obligations and requirements in a business unit.

6. Requirements for Tool Support

From our previous work analyzing policies and regulations, we propose requirements for a tool to support personnel from CSOs and business managers to software developers and system administrators. The work of CSOs is to organize and delegate the high-level goals that are relatively few in number and

change infrequently, whereas, system administrators must respond to numerous system requirements that adapt systems to emerging business needs and security vulnerabilities. Furthermore, compliance officers and auditors need to evaluate the delegation and refinement decisions made by such personnel.

The following is a minimal set of software requirements (SR) for a tool that implements our framework. Following the requirements, we briefly describe how these requirements would interact with users as well as future extensions to the framework that would extend the capabilities of the tool.

1. EXPLORATION:

SR₁: The tool shall allow each user to view the “context” of an obligation. The context includes:

- (1) The delegator who assigned the obligation to another actor;
- (2) The obligations, if any, that the assigned obligation refines (e.g., the overall goal the obligation is intended to achieve.)
- (3) The refinements, if any, to the assigned obligation proposed by the delegatee.
- (4) For a sequence of delegations and refinements, any certifications verified by digital signatures and the identity of signatory actor.

SR₂: The tool shall allow each user to separately view their obligations that are assigned to them by other actors.

SR₃: The tool shall allow each user with the administrative obligation to separately view the requirements assigned to the concerned systems.

SR₄: The tool shall allow each user to organize other users into groups such as business units or projects. Views may be restricted to only those users and obligations in a particular group.

2. ASSIGNMENT/ REFINEMENT:

SR₅: The tool shall allow users to assign obligations by user or group.

SR₆: The tool shall allow users to create rules that define the pre-conditions under which users are assigned obligations; the rules are automatically applied to users who will receive special indication, when viewing the obligation, that the obligation was assigned and by which rule.

SR₇: The tool shall allow users to refine their assigned obligations by creating new obligations and assigning those obligations to themselves or other actors.

3. CERTIFICATION:

SR₈: The tool shall allow users to certify a decision sequence using their secret key. The certificate (with digital signature) is used to indicate to other users the decision sequence is approved by the user and has not changed since the certification date. The certification is verified using the user’s registered public key.

A tool that meets these requirements will allow users to view their assigned obligations, refine these obligations into new obligations and delegate these obligations to others. In addition, compliance officers and auditors can expand obligations into two hierarchies that show delegations and refinements. These hierarchies allow compliance officers and auditors to investigate the decision chains introduced in Section 3 and discussed in Section 4.1. Furthermore, auditors can certify these chains and users can verify these certifications.

In addition, we believe users should be able to identify conflicts between obligations. These conflicts will require the owner or delegator of two or more obligations to assess the broader situation and prioritize these obligations. Another form of conflict or under-specification is an assigned obligation without prerequisite permissions. Our framework can be extended to support user requests for permissions from those with the authority to delegate them. Using the tool, these delegators can verify that the necessary obligations justifying the need for these permissions are assigned to the user and, if those obligations are ever revoked, these permissions can be automatically revoked, ensuring overall consistency between policies and systems.

Furthermore, we foresee organizing standards and regulations in a custom set of plug-ins that contain pre-defined actors, obligations and refinements sufficient to align with an organization’s business practices. These plug-ins should be developed by standards bodies and regulators and provided to organizations as part of the traditional compliance package. An organization would align the actors provided by the plug-in with their own personnel, who would then refine their new obligations and align them with existing software systems. Alternatively, the users may identify the need for new systems that meet these new obligations.

Finally, software products could be distributed with their own requirements plug-ins. System administrators that bring a new product online would receive a list of requirements certified by the product developer, based on actual developer test cases. Whereas organizations that develop their own software assume their own liability for resulting software failures, these third-party certifications would establish third-party liability against explicit product requirements within the

distributed management framework. Given a non-compliance event associated with a specific decision chain involving a third-party product, both the organization and third-party can determine the role the product played in the fault: the product may have failed to satisfy its requirements, or the requirements may have been inappropriately aligned with personnel obligations they could not reasonably satisfy.

7. Conclusion

In summary, we present a framework that combines delegation and refinement in a distributed system to capture the decisions that executives, managers, system administrators and developers make to achieve compliance with standards and regulations. The contributions in this work include a formal definition of decision chains that auditors can certify and review when determining compliance for an organization. Furthermore, we instantiate our framework using an example from the HIPAA Security and Privacy Rules and propose requirements for a tool to implement the framework with discussion of its use.

Acknowledgements

This work was funded, in part, by the Center for Education and Research in Information Assurance and Security (CERIAS) and the NSF Cyber Trust grant #0430166 titled: *Collaborative Research: Comprehensive Policy-Driven Framework for Online Privacy Protection: Integrating IT, Human, Legal and Economic Perspectives.*

References

- [AEB04] A.I. Antón, J.B. Earp, D. Bolchini, Q. He, C. Jensen and W. Stufflebeam. "The Lack of Clarity in Financial Privacy Policies and the Need for Standardization," □ *IEEE Security & Privacy*, 2(2), pp. 36-45, 2004.
- [AEV06] A.I. Antón, J.B. Earp, M.W. Vail, N. Jain, C. Gheen and J.M. Frink. "An Analysis of Web Site Privacy Policy Evolution in the Presence of HIPAA," □ To Appear: *IEEE Security & Privacy*, NCSU Technical Report #TR-2004-21, 2006.
- [Ant97] A.I. Antón, "Goal Identification and Refinement in the Specification of Software-Based Information Systems." Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 1997.
- [AMP94] A.I. Antón, W.M. McCracken, C. Potts. "Goal Decomposition and Scenario Analysis in Business Process Engineering." *Advanced Information Systems Engineering, 6th International Conference Proceedings (CAiSE '94)*, Utrecht, Netherlands, pp. 94-104, 6-10, 1994.
- [BLM04] A.K. Bandara, E.C. Lupu, J. Moffett, A. Russo, "A Goal-based Approach to Policy Refinement." In Proc. Policies for Dis. Sys. and Nets, Yorktown Heights, NY, USA, pp. 229-239, 2004.
- [BDF02] O. Bandmann, M. Dam, B.S. Firozabadi, "Constrained Delegation." In Proc. *IEEE Symp. on Security Privacy*, pp. 131-140, 2002.
- [BS00] E. Barka, R. Sandhu, "Framework for Role-based Delegation Models." In Proc. 16th Annual Conf. on Computer Security Applications, pp. 168-176, 2000.
- [BA05a] T.D. Breaux, A.I. Antón, "Deriving Semantic Models from Privacy Policies." In Proc. *IEEE 6th Workshop on Policies for Distributed Systems and Networks*, Stockholm, Sweden, pp. 67-76, 2005.
- [BA05b] T.D. Breaux, A.I. Antón, "Analyzing Goal Semantics for Rights, Permissions, and Obligations." In Proc. *IEEE 13th Requirements Engineering Conference*, Paris, France, pp. 177-186, 2005.
- [BA05c] T.D. Breaux, A.I. Antón, "Mining Rule Semantics to Understand Legislative Compliance." In Proc. *ACM Workshop on Privacy in Electronic Society*, Alexandria, Virginia, USA, pp. 51-54, 2005.
- [BAK06] T.D. Breaux, A.I. Antón, C-M. Karat, J. Karat, "Enforceability vs. Accountability in Electronic Policies." In Proc. *IEEE 7th Workshop on Policies for Distributed Systems and Networks*, London, Ontario, pp. 227-230, 2005.
- [DDL01] N. Damianou, N. Dulay, E. Lupu, M. Sloman, "The Ponder Policy Language." In Proc. Work. Policies for Dist. Sys. and Nets., Bristol, UK, pp. 29-31, 2001.
- [DDM97] R. Darimont, E. Delor, P. Massonet, A. van Lamsweerde, "GRAIL/KAOS: An Environment for Goal-driven Requirements Engineering." *IEEE 19th Int'l Conf. Soft. Engr.*, Boston, MA, pp. 612-613, 2005.
- [DLF93] A. Dardenne, A. van Lamsweerde, S. Fickas, "Goal-directed Requirements Acquisition." *Sci. Comp. Programming*, v. 20, pp. 3-50.
- [DLS01] N. Dulay, E. Lupu, M. Sloman, N. Damianou, "A Policy Deployment Model for the Ponder Language" In Proc. *IEEE/IFIP Int'l Sym. on Intenerated Net. Mgmt.* Seattle, WA, USA, pp. 529-543, 2001.
- [GMM05] P. Giorgini, F. Massacci, J. Mylopoulos, N. Zannone, "Modeling Security Requirements through Ownership, Permission and Delegation." In Proc. 13th IEEE Int'l Reqs. Engr. Conf., Paris, France, pp. 167-176, 2005.
- [[MU00] N.H. Minsky, V. Ungureanu, "Law-Governed Interaction: a Coordination and Control Mechanism for Heterogeneous Distributed Systems." *ACM Trans. on Soft. Engr. and Meth.*, 9(3), pp. 273-305, 2000.
- [M99] J.D. Moffett, "Requirements and Policies." In Proc. *Policy Workshop*, Bristol, U.K., 1999.
- [MS91] J.D. Moffett, M.S. Sloman, "The Representation of Policies as System Objects." In Proc. *Conf. on Organizational Computing Systems*, Atlanta, Georgia, USA, pp.171-184, 1991.
- [OS02] S. Oh, R. Sandhu. "Role Administration: A Model for Role Administration Using Organization Structure." In Proc. 7th ACM Sym. on Access Control Models and Tech., Monterey, CA, USA, pp. 155-162, 2002.
- [PCN04] J.S. Park, K.P. Costello, T.M. Neven, J.A. Diosomito, "A Composite RBAC Approach for Large, Complex Organizations." In Proc. 9th ACM Sym. On Access Control Models and Technologies, Yorktown Heights, NY, USA, pp. 163-172, 2004.

- [PS04] J.S. Park, R. Sandhu, “The UCON_{ABC} Usage Control Model” *ACM Trans. on Information and System Security*, 7(1), pp. 128–174, 2004.
- [RBS03] J. Rees, S. Bandyopadhyay, E.H. Spafford, “PFIREs: A Policy Framework for Information Security.” *Communications of the ACM*, 46(7), pp. 101-106, 2003.
- [HPR] “Standards for Privacy of Individually Identifiable Health Information.” 45 CFR Part 160, Part 164 Subpart E. In *Federal Register*, vol. 68, no. 34, February 20, 2003, pp. 8334 – 8381.
- [HPS] “Standards for the Protection of Electronic Protected Health Information” 45 CFR Part 164, Subpart C. In *Federal Register*, 68(34), February 20, 2003, pp. 8334 – 8381.
- [EY05] Ernst & Young, *Global Information Security Survey 2005: Report on the Widening Gap*, 2005.
- [800012] *An Introduction to Computer Security: the NIST Security Handbook*, NIST SP-800-12, Gaithersburg, MD, USA, 1995.