Towards Improved Privacy Policy Coverage in Healthcare Using Policy Refinement

Rafae Bhatti and Tyrone Grandison

IBM Almaden Research Center,
650 Harry Road, San Jose, California 95120, USA
{rbhatti,tyroneg}@us.ibm.com

Abstract. It is now mandatory for healthcare organizations to specify and publish their privacy policies. This has made privacy management initiatives in the healthcare sector increasingly important. However, several recent reports in the public media and the research community about healthcare privacy [1, 2] indicate that the use of privacy policies is not necessarily a strong indication of adequate privacy protection for the patient. These observations highlight the fact that the current state of privacy management in healthcare organizations needs improvement. In this paper, we present PRIMA, a PRIvacy Management Architecture, as a first step in addressing this concern. The fundamental idea behind PRIMA is to exploit *policy refinement* techniques to gradually and seamlessly embed privacy controls into the clinical workflow based on the actual practices of the organization in order to improve the *coverage* of the privacy policy. PRIMA effectively enables the transition from the current state of *perceived to be privacy-preserving* systems to *actually privacy-preserving* systems.

Key words: Privacy Management, Healthcare, HIPAA, Compliance, Refinement

1 Introduction

Privacy management is one of the main inhibitors of the deployment, adoption and use of electronic records systems in the healthcare industry. There are several privacy laws and regulations that have emerged around the world in the past few years [3], such as the Personal Data Protection Law [4] in Japan, the Health Insurance Portability and Accountability Act (HIPAA) in the United States [5] and the Personal Information Protection and Electronic Documents Act [6] in Canada. For American healthcare, HIPAA is normally assumed to provide the baseline for privacy compliance for healthcare entities.

While HIPAA and other healthcare-related privacy laws and regulations make it mandatory for organizations to specify and publish privacy policies regarding the use and disclosure of personal health information, recent media and academic reports about healthcare privacy [1, 2] indicate that there is not necessarily a strong correlation between the use of privacy policies and adequate patient privacy protection. In [2], the authors examined the actual access patterns for a Norwegian healthcare organization. Their study indicates that in spite of possessing strict policy and regulation, the security mechanisms of the

IT system were under-utilized and often bypassed in order to deliver care. This phenomenon creates an over-reliance on exception-based access for any situation that does not seem to be explicitly covered by the policy, and even for some that are. In the healthcare environment, disallowing access during service delivery is not an option because it may lead grim consequences for the patient.

These observations, which have also been echoed in the United States [1, 7], intimate the existence of a general state of affairs in healthcare organizations, where circumventing data security and privacy controls is the rule and not the exception. This trend is alarming because it negates the existence and efficacy of policy. In this state, the policy does not precisely represent or embody the actual level of data protection afforded to the patient, i.e. the policy is no longer a genuine reflection of the organization's privacy practices. Additionally, it undermines the notion of empowering the patient, as his consent may no longer be valid because the policy is no longer valid. While these observations may appear to reflect negatively on the healthcare organizations, these scenarios are in fact a direct result of applying prior technology without considering the nuances of the clinical workflow. In light of the recent push to electronic health records [8], this conundrum will multiply in effect.

It is our belief that it is possible to leverage artifacts from the actual clinical workflow to inform and construct appropriate privacy protection mechanisms for patients. We purport that *policy refinement*, which we will define as the process of improving the rules that define the level of protection, can be employed to gradually and seamlessly embed meaningful privacy controls into the clinical workflow based on the actual practices of the organization. This concept is the base construct for the PRIMA system, which also leverages data mining [9] and Hippocratic Database technology [10]. In particular, the architecture builds upon the *Active Enforcement* [11] and *Compliance Auditing* [12] components of the Hippocratic Database technology, and leverages standard data analysis techniques. PRIMA's *policy refinement* helps mitigate the above stated conundrum by (i) improving the design of the policies, which should elevate the level of privacy protection afforded to the patient, and (ii) better aligning the system policies with the actual privacy practices of the organization to improve the *coverage* of the privacy policy. To the best of our knowledge, no prior work on policy refinement for healthcare systems has been undertaken.

The rest of the paper is organized as follows. In section 2, we provide some background from a regulatory standpoint regarding use and disclosure of personal health information. Then we analyze the rationale for stated privacy policies not being actual representations of patient privacy protection. In section 4, we describe the PRIMA architecture and technical details. In section 5, we illustrate the use of PRIMA in a healthcare scenario. We conclude in section 6.

2 Background

Privacy legislation around the world are based on the notions captured in the OECD Data Protection Principles [3]. For the purposes of exemplification, and without loss of generality, we ground our discussion on privacy protection in the healthcare sector by examining the **Limited Use and Disclosure** provision of the HIPAA Privacy Rule. The motivation and arguments for this provision can

be extrapolated to the other similar legislation, regulations and laws around the world.

With respect to the HIPAA Privacy Rule, *covered entities* refer to health plans, healthcare providers and healthcare clearinghouses, and *Protected Health Information (PHI)* refers to all individually identifiable health information held or transmitted by a covered entity or its business associate, electronically, on paper, or orally. The limited use and disclosure provision requires that covered entities must use or disclose the *minimum necessary* PHI for a specific purpose and ensure the development and implementation of policies and procedures governing access and use.

In accordance with the purpose specification provision in privacy regulations, a privacy policy statement normally contains specific purposes for which data can be used or disclosed. However, the defined purposes tend to be very broad in scope [2]. For example, many real-world policies mention collecting information for the purpose of "administering healthcare". This granularity is coarse enough to subsume many information uses and disclosures. We recognize that this practice may not be performed with mal-intent, but may be a function of reducing the complexity of policy specification, which reduces the size of the rule base.

It was also observed [2] that organizations had difficulty defining specific employee categories (i.e. useful roles), which define the authorizations for viewing specific patient data categories [13]. Typically, the collected information is available to all "members of medical staff", which effectively results in an umbrella authorization. Again, we recognize that the transition from the *generalist school of medicine* to the *specialist school of medicine* over the last few decades has meant that the number of healthcare professionals involved in the delivery of care, to a single patient, has increased significantly and that categorizations may be hard because roles are so fluid and cannot be assumed to be mutually exclusive. Additionally, the primary purveyors of healthcare tends to be the nursing staff and it is understandable that authorization difficulties may exist. However, there are still clearly defined lines, at least legally, on who should be able to view and use particular aspects of patient data. Thus, role delineation and categorization is necessary and critical. For a few years now, the broader community has realized and advocated the need for fine-grained access control. This view is shared by both academic researchers [14–16] and medical professionals [17, 13].

From the previous discussions, it is clear that, despite the underlying reasons, the limited use and disclosure provision in the HIPAA Privacy Rule has not been interpreted and implemented very well in existing healthcare informatics systems. Our view is that healthcare is an industry that will always require customized mechanisms to balance privacy and operational considerations.

In order to not be disruptive and to automate this process of customization, PRIMA attempts to gradually embed policy controls in the system by analyzing the information already existing in the system and informing the new state that the system needs to evolve to. Thus, the overall goal is to bridge the disparity between intended and achieved levels of privacy protection.

3 Formal Model

For an arbitrary healthcare organization, *HO*, the policy that they define for their IT systems embodies the regulations, legislation, laws and organizational

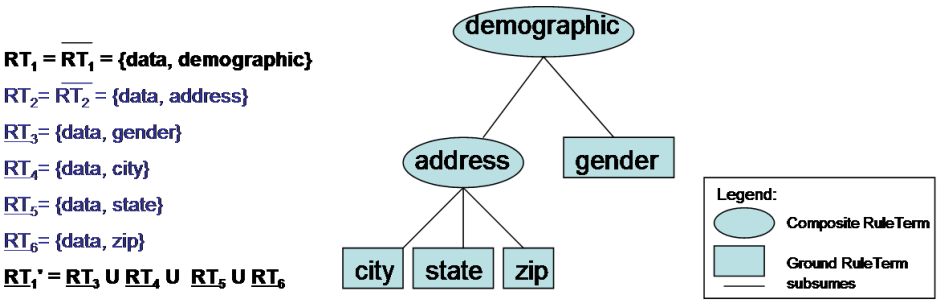


Fig. 1. A Sample Privacy Policy Vocabulary.

mandates that they must follow. This represents what they would ideally like to happen, i.e. their ideal workflow W_{Ideal}.

The studies presented earlier state that after a period of operation, the audit trails of system accesses, which represents *HO's* real workflow W_{Real}, is primarily filled with exception-based access statements.

3.1 Core Constructs

Let's formalize the underlying notions and the goal of the PRIMA system, which is the reduction of the gap between real and ideal workflows. We assume that the HO has chosen a privacy specification notation and has a mapping from the terms used in this notation to the artifacts that the IT system will manipulate. Hereafter, we refer to these artifacts as the privacy policy vocabulary (or vocabulary, for short). The formal representation of the policies relies on the key concepts that make up a policy, namely *RuleTerm* and *Rule*.

Definition 1. *(RuleTerm): A RuleTerm (RT) is a tuple with two literal-valued elements, attr and value. It is written as* $RT = (attr, value)$*. The two elements of RT are accessed as RT.attr and RT.value.*□

A *RuleTerm* models the assignment of an attribute in a policy rule. For example, demographic data is represented as (data, demographic) and telemarketing purposes as (purpose, telemarketing). *RuleTerm* is the fundamental construct for our formalism in order to ensure that the model is applicable to any arbitrary specification notation.

Definition 2. *(RuleTerm Types): A RuleTerm, RT, is considered* ***ground*** *(written as* $\underline{RT}$*) iff its attribute value (RT.value) is an atomic-valued literal, with respect to the privacy policy vocabulary used. Otherwise, it is* ***composite*** *(written as* $\overline{RT}$*).*□

Let's define RT_1 = (data, demographic), RT_2 = (data, address), and RT_3 = (data, gender). The particular policy vocabulary used here is depicted in Figure 1[1]. In this example, RT_3 can be considered a ground *RuleTerm* since

[1] Only the *RT.value* element of each RT is shown in the figures for conciseness.

it contains the attribute value "gender", which cannot be further divided into multiple *RuleTerm*s according to the chosen vocabulary. On the other hand, RT_1 is unequivocally a composite *RuleTerm* since demographic information could be further divided into information about address and gender. In fact, both RT_2 and RT_3 are subsumed by RT_1.

For each composite RuleTerm $\overline{RT}$, we assume the existence of a special set, written as $\underline{RT'}$, that contains all the ground rule terms $\underline{RT_1}, \ldots, \underline{RT_n}$ that can be derived from $\overline{RT}$ using the chosen privacy policy vocabulary. In the example shown in Figure 1, the set $\underline{RT'_1}$ for $\overline{RT_1}$ is shown to comprise of four ground RuleTerms.

Definition 3. *(Existence of Ground RuleTerm): Given a composite RuleTerm $\overline{RT}$ and a privacy policy vocabulary, it can always be transformed to a corresponding ground RuleTerm $\underline{RT}$. Formally,* $(\forall x : RT(x \in \overline{RT})) \rightarrow (\exists y : RT(y \in \underline{x'})$.[2]

An important notable notion is that of the equivalence of *RuleTerm*s. Given a privacy policy vocabulary or set of vocabularies, the equivalence notion allows for the comparison of *RuleTerm*s.

Definition 4. *(Equivalence of RuleTerms): Two RuleTerms, RT_i and RT_j, are considered equivalent, written $RT_i \approx RT_j$, iff* $\exists x, y : RT(x \in \underline{RT_i}) \wedge (y \in \underline{RT_j}) \wedge (x.attr = y.attr) \wedge (x.value = y.value))$.□

In the example in Definition 1, both RT_2 and RT_3 are equivalent to RT_1 because there exists ground *RuleTerms* $\underline{RT_2}$ and $\underline{RT_3}$ belonging to the set $\underline{RT'_1}$.

Definition 5. *(Rule): A Rule, R_i, is a conjunction of RuleTerms. It is written as $R_i = \{RT_1 \wedge \ldots \wedge RT_n\}, n \geq 1$. The number of RuleTerms of a Rule, n, is referred to as the cardinality of the Rule, written as $\#R$.*□

A *Rule* models a specific combination of attribute assignments, which represents individual statements in a policy. For example, "nurses are authorized to see insurance information for billing purposes"may be represented as $\{(data, insurance) \wedge (purpose, billing) \wedge (authorized, nurse)\}$.

A *Rule*, R_i, is said to be a **ground** rule (written as $\underline{R_i}$) if all *RuleTerm*s in R_i are ground. R_i is a **composite** rule (written as $\overline{R_i}$) if there exists at least one *RuleTerm* that is a composite *RuleTerm*.

Corollary 1. *(Existence of Ground Rule): From Definition 3, it follows that for any Rule R_i, there always exists a corresponding Rule $\underline{R_i}$.*

Definition 6. *(Equivalence of Rules): Two Rules, R_1 and R_2, are said to be equivalent, written as $R_1 \approx R_2$, iff* $(\#\underline{R_1} = \#\underline{R_2}) \wedge (\forall x : RT\ (x \in \underline{R_1}) \rightarrow (\exists\ y : RT\ (y \in \underline{R_2}) \wedge (x \approx y))$.□

Essentially, rules are equivalent when they have the same number of terms and every term in one rule is equivalent to another in the other rule.

[2] Note that $\underline{x'}$ is a set

Definition 7. *(Policy): A policy, P_x, is a collection of rules that is symbolically tied to a data store x, where x can be either the policy store, PS, or the audit logs, AL. A policy is written as $P_x = R_x^1, \ldots, R_x^m, m \geq 1$. The number of Rules in the P_x, m, is referred to as the cardinality of P_x, which is written as $\#P_x$.*□

For our purposes, we equate W_{Ideal} to P_{PS} and W_{Real} to P_{AL}. This is a simplification that holds true because the artifacts under investigation are the workflows relating to healthcare data disclosure and use.

Given that *RuleTerm*s and *Rule*s can be either ground or composite, a *Policy* can be too. A *Policy*, P_x, is a **ground** policy (written $\underline{P_x}$) if all *Rules* are ground *Rule*s. If there is at least one composite *Rule* in $\overline{P_x}$, then it is a composite policy (written $\overline{P_x}$).

For each composite policy $\overline{P_x}$, we assume the existence of a special set, written as $\underline{P_x}'$, that contains all the ground rules that can be derived from the composite rules in P_x using the chosen privacy policy vocabulary. The existence of this set follows from Definitions 3, 5, and 7.

Corollary 2. *(Existence of Ground Policy): From Corollary 1, it follows that for any Policy P_x, there always exists a corresponding Policy $\underline{P_x}$.*

3.2 Policy Coverage

The concept of policy coverage builds upon the idea of comparing the real state of the system P_{AL}, as represented by the audit logs, with the ideal state of the system P_{PS}, as represented by the policy store that contains the rules specified by some system administrator, privacy officer, etc. We recognize that the P_{PS} will normally be specified at a high level of abstraction (and later mapped to low-level control statements), and that P_{AL} will be low-level information gathered by the system in its normal operation. Thus, in order to perform a meaningful comparison, we must transform both to the lowest common denomination, i.e ground policies, and then do our evaluation.

Definition 8. *(Range): Given a policy P_x, its range, $Range_{P_x}$, is the set containing all the rules in $\underline{P_x}'$.*□*.*

The cardinality of the *Range* is the number of elements in the set $Range_{P_x}$, written $\#Range_{P_x}$. Given this definition, we can now define policy coverage.

Definition 9. *(Coverage): Given two policies P_x and P_y, and a privacy policy vocabulary V, the coverage of P_x in relation to P_y, written $Coverage_{P_y}^{P_x}$, is given by $\#(Range_{P_x} \cap Range_{P_y}) \div \#Range_{P_y}$.*□

Here, the intersection is computed using the equivalence of rules as defined in Definition 6. Informally, the policy coverage in a given system is defined as the amount of overlap between the real and ideal representations of the system state, namely P_{PS} and P_{AL}. The coverage of P_x with respect to P_y is computed as a ratio using the algorithm ComputeCoverage given below.

The overall goal of the PRIMA system is to move towards a state of complete coverage, which is defined below. It is acknowledged that complete coverage may not be attainable given the human component, but higher levels of coverage should be a realistic goal. The process of improving the policy coverage is visually shown in Figure 2.

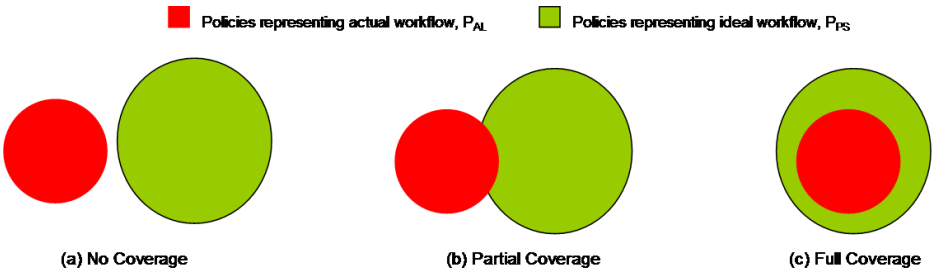


Fig. 2. Simplified Visual Representation of Policy Coverage.

Definition 10. *(Complete Coverage): Given two policies P_x and P_y, and a privacy policy vocabulary V, P_x completely covers P_y iff $Range_{P_x} \cap Range_{P_y} = Range_{P_y}$.*□

Algorithm 1 $ComputeCoverage(P_x, P_y, V)$

Require: $\exists getCardinality(S)$ (returns the cardinality of a set S)
Require: $\exists getRange(P, V)$ (returns the range of the policy P according to the policy vocabulary V)

1: $coverage \leftarrow 0$
2: $range_x[] \leftarrow getRange(P_x, V)$
3: $range_y[] \leftarrow getRange(P_y, V)$
4: $m_y \leftarrow getCardinality(range_y)$
5: $overlap[] \leftarrow range_x \cap range_y$
6: $m_o \leftarrow getCardinality(overlap)$
7: $coverage = m_o \div m_y$
8: **return** *coverage*

3.3 Illustrative Example

Let's look at a simple example that demonstrates coverage calculation. Consider the policy store shown in Figure 3(a). Let the policy tied to this policy store be denoted as P_{PS}. The top table shows the abstract-level composite policy, $\overline{P_{PS}}$, which comprises of three rules. The bottom table shows a portion of the ground policy, $\underline{P'_{PS}}$.

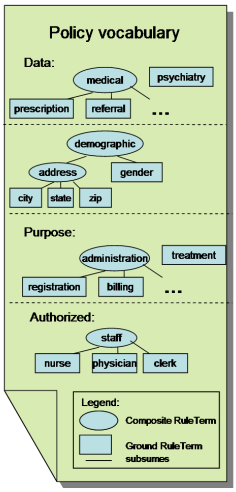
Now consider the audit logs shown in Figure 3(b). Let the policy tied to the audit logs be denoted as P_{AL}. By default, this policy is a ground policy, $\underline{P_{AL}}$, and it comprises of six rules. We observe that rules 1, 2, and 5 in $\underline{P_{AL}}$ are matched by rules 1a, 1b, and 3a, respectively, in $\underline{P'_{PS}}$, but rules 3, 4, and 6 in $\underline{P_{AL}}$ are not matched by any rules in $\underline{P'_{PS}}$. This indicates that exception-based accesses were utilized to access the data in a situation which not was allowed by the policy. These exception scenarios are pointed out in the figure.

$\overline{P_{PS}}$ (The abstract level composite policy tied to policy store)

Rule 1	Attr: data Value: medical	Attr: purpose Value: treatment	Attr: authorized Value: nurse
Rule 2	Attr: data Value: psychiatry	Attr: purpose Value: treatment	Attr: data Value: physician
Rule 3	Attr: data Value: demographic	Attribute: purpose Value: administration	Attribute: data Value: staff

P'_{PS} (The ground policy tied to policy store)

Rule 1a	Attr: data Value: prescription	Attr: purpose Value: treatment	Attr: authorized Value: nurse
Rule 1b	Attr: data Value: referral	Attr: purpose Value: treatment	Attr: authorized Value: nurse
Rule 2	Attr: data Value: psychiatry	Attr: purpose Value: treatment	Attr: authorized Value: physician
Rule 3a	Attr: data Value: address	Attr: purpose Value: billing	Attr: authorized Value: clerk
Rule 3b	Attr: data Value: state	Attr: purpose Value: billing	Attr: authorized Value: clerk
Rule 3c	Attr: data Value: zip	Attr: purpose Value: billing	Attr: authorized Value: clerk
Rule 3d	Attr: data Value: city	Attr: purpose Value: billing	Attr: authorized Value: clerk
Rule 3e	Attr: data Value: address	Attr: purpose Value: registration	Attr: authorized Value: nurse
...	...	...	...



(a)Policy tied to policy store

P_{AL} (The ground policy tied to audit logs)

Rule 1	Attr: data Value: prescription	Attr: purpose Value: treatment	Attr: authorized Valu: nurse
Rule 2	Attr: data Value: referral	Attr: purpose Value: treatment	Attr: authorized Value: nurse
Rule 3	Attr: data Value: referral	Attr: purpose Value: **registration**	Attr: authorized Value: nurse
Rule 4	Attr: data Value: psychiatry	Attr: purpose Value: treatment	Attr: authorized Value: **nurse**
Rule 5	Attr: data Value: address	Attr: purpose Value: billing	Attr: authorized Value: clerk
Rule 6	Attr: data Value: **prescription**	Attr: purpose Value: billing	Attr: authorized Value: clerk
...	...	...	...

Exception Scenarios:

"nurse" may need to access **"referral"** data for **"registration"** purposes too

"nurse" too may need to access **"psychiatry"** data for **"treatment"** purposes

"clerk" may need to access **"prescription"** data too for **"billing"** purposes

(b)Policy tied to audit logs

Fig. 3. Example scenario illustrating coverage computation.

To elaborate, the reason for rule 3 not being matched is that a *nurse* needed to access *referral* data for *registration* purpose, but the policy allows the use of such data only for *treatment* purpose. The reason for rule 4 not being matched is that a *nurse* needed to access *psychiatry* data for *treatment* purpose, but the policy allows such data to be accessed only by a *physician*. Lastly, the reason for rule 6 not being matched is that a *clerk* needed to access *prescription* data for *billing* purpose, but the policy allows the use of only *demographic* data for this purpose. These scenarios indicate the customary practices during the clinical workflow which should be incorporated in the privacy policy of the system.

Invoking *Compute Coverage*(P_{PS},P_{AL},V) on this system, the policy coverage of P_{PS} with respect to P_{AL} in this system is found to be 50 %, i.e. $\#(Range_{P_{PS}} \cap Range_{P_{AL}}) \div \#Range_{P_{AL}}$ is 3/6.

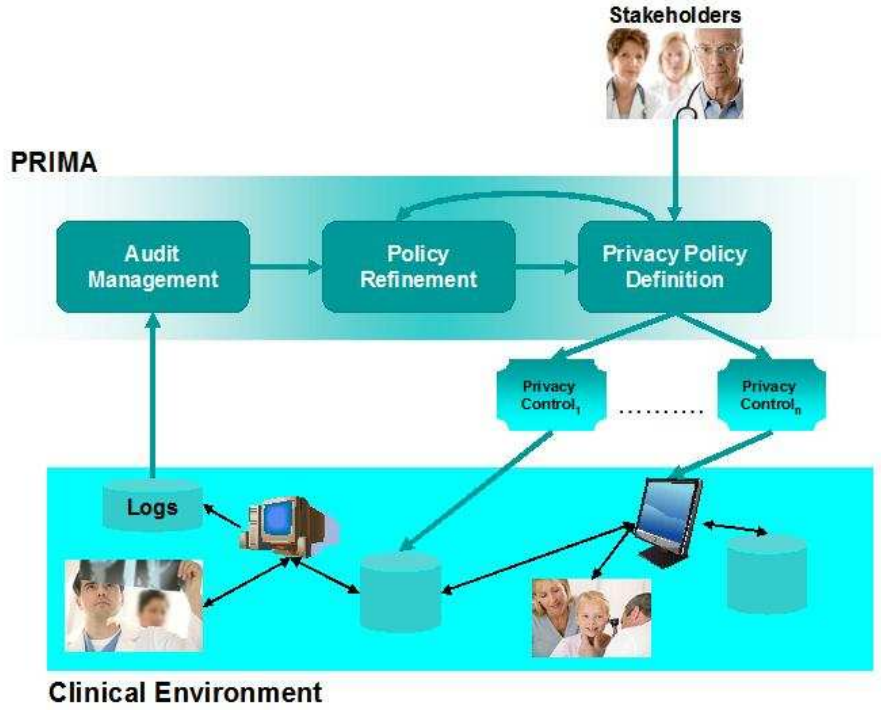


Fig. 4. The PRIvacy Management Architecture (PRIMA)

4 PRIMA: The System

The discussion on policy coverage is useful in formally understanding the goal of PRIMA. However, a significant consideration, from the clinical standpoint, is the design of the PRIMA system in a way that aligns with, and not impedes, the clinical workflow. PRIMA attempts to improve policy coverage by gradually embedding new policy statements, which were discovered through the process of policy refinement, into the clinical system.

Figure 4 provides a high-level view of PRIMA. *Stakeholders* define the privacy policies for the *HO*, which is embedded in *privacy controls* that are integrated into the clinical environment. One of these privacy controls is an auditing function that automatically generates entries for the system's audit logs. These logs are either periodically replicated or PRIMA-enabled, by the construction of a consistent consolidated view of them. In the simplest case, there is just one log. We will discuss desired features of audit controls in section 4.2. At regular intervals or at the request of the stakeholders, the *Policy Refinement* component extracts input from the *Audit Management* component and the *Privacy Policy Definition* component and outputs a list of definitions, if any exist, that should be included in the policy definitions. Let's discuss each of these components in more depth.

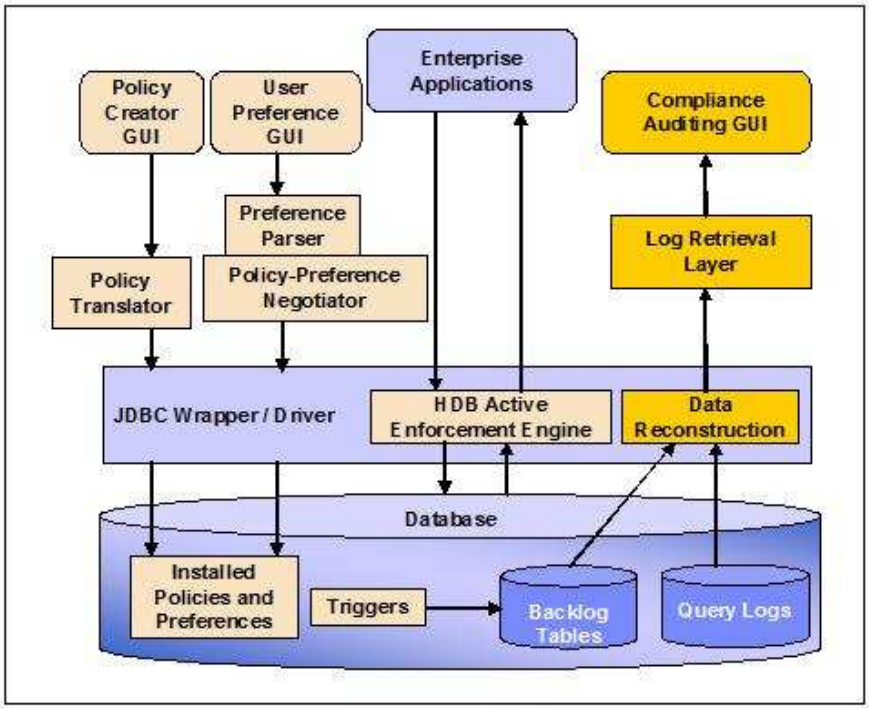


Fig. 5. Combined Architecture of HDB Active Enforcement (AE) and Compliance Auditing (CA).

4.1 Privacy Policy Definition

In this context, we assume that input is gathered by all the *stakeholders*, i.e. patients, medical practitioners, payers etc., and a representative uses this information to specify the *HO*'s policy. At an abstract level, PRIMA may leverage any arbitrary privacy policy definition tool that has the facility to create privacy controls that can be embedded into the clinical workflow. As a proof of concept, the initial instantiation utilizes the HDB Active Enforcement [11] and HDB Compliance Auditing [12] components (Figure 5), which produces augmented database interfaces that both enforce fine-grained policy and patient consent and create minimal impact, storage and performance efficient logs. Our user would use the HDB Control Center to enter fine-grained rules, patient consent information and specify what needs to be auditable.

The HDB components (Figure 5) operate at the middleware layer between the clinical database and the end user query interface. When the AE component receives user queries, it rewrites the queries so that only data consistent with policy and patient preferences is returned. The rewritten request gets sent to the database for execution and is also stored along with the query issuer, purpose, time and date in the audit log.

4.2 Audit Management

Retroactive controls, such as audit trails, and the threat of inevitable violation detection and prosecution are prevalent in healthcare information systems. Unfortunately, there are a series of concerns that may stem this approach. The first concern is the impact on the existing infrastructure, i.e. the degradation in system performance and the increased storage demand. The second is the nature of the technology's use, i.e. the logs tend to be used only when someone raises a red flag about an improper data disclosure, not as a part of a continuous, proactive process. Finally, not all the necessary contextual information may have been logged with the request. The first and third concerns translate into requirements for auditing systems within the clinical environment.

Use of HDB Compliance Auditing in the clinical workflow allows us to meet these two requirements. The schema for an audit entry is $\{$*(time,*t_j*), (op,*X_j*), (user,*u_j*), (data,*d_j*), (purpose,*p_j*), (authorized,*a_j*), (status,*s_j*)*$\}$, where t_j is the entry's timestamp, X_j is either 0 (disallow) or 1 (allow), u_j is the entity that requested access, d_j is the data to be accessed, p_j is the purpose for which the data is accessed, a_j is the authorization category (e.g. role) of the entity that requested access, and s_j is either 0 (exception-based access) or 1 (regular access). The status s_j of access would in practice be recorded at the time the user either chooses or manually enters the purpose of access, where former corresponds to a regular access and latter to an exception-based access. We realize that this model could be augmented with the inclusion of conditions. However, the techniques that will be used on the core elements presented are also applicable to augmentations of the model

The PRIMA *Audit Management* component acts as a consolidation for the audit systems in the clinical environment. In the first instantiation, we use DB2 Information Integrator as the federation technology in the PRIMA *Audit Management* component to create a virtual view of all the audit trails. Alternative methods may be used that can consolidate all audit data in one place for subsequent analysis.

Irrespective of the mechanism used to populate the P_{AL} used by PRIMA, we must be cognizant that the audit logs may contain different kinds of information. There may be data on attempts to break into the system, i.e. possible violations or data breaches, or information that represents undocumented, informal clinical practice. We need to differentiate between violations and informal practice entries in the refinement process.

4.3 Policy Refinement

Refinement is based on the premise that a feedback loop is required between real and ideal policy; in order to create policy that (i) more accurately represents the covered entity's intent and behavior, and (ii) more adequately represents the level of privacy protection afforded to the patient.

The pseudocode for the refinement process is given in Algorithm 2. The function is provided (i) the policy store, P_{PS}, (ii) the policies in the logs, P_{AL}, and (iii) the privacy vocabulary, V, being used by this particular covered entity. P_{AL} is filtered to remove prohibitions, analysis is performed on the resulting set to create a set of patterns (if any exist), which are then pruned based on the coverage of P_{PS} with respect to P_{AL}.

Algorithm 2 $Refinement(P_{PS}, P_{AL}, V)$

Require: $\exists Filter(P)$ (returns the non-prohibitions in policy P)
Require: $\exists extractPatterns(P)$ (returns the rules that may be undocumented patterns)
Require: $\exists Prune(Patterns, P_{PS}, V)$ (returns the patterns to be incorporated into the system's current policy)
1: $Practice[] \leftarrow$ **Filter**(P_{AL}) (see Algorithm 4.3)
2: $Patterns[] \leftarrow$ **extractPatterns**$(Practice, V)$ (see Algorithm 4)
3: $usefulPatterns[] \leftarrow$ **Prune**$(Patterns, P_{PS}, V)$ (see Algorithm 6)
4: **return** $usefulPatterns$

Algorithm 3 $Filter(P)$

Require: $\exists getCardinality(P)$ (returns the cardinality of P)
Require: $\exists getRule(P, i)$ (returns the ith rule of policy P)
Require: $\exists getStatus(R)$ (returns the value for the status attribute in rule R)
Require: $\exists appends(R, R_{set})$ (appends Rule R to the set of rules R_{set})
1: $Practice \leftarrow []$
2: $n \leftarrow getCardinality(P)$
3: **for** $i = 1$ to n **do**
4: $R_i \leftarrow getRule(P, i)$
5: **if** $getStatus(R_i) == 0$ **then**
6: $append(R_i, Practice)$
7: **end if**
8: **end for**
9: **return** $Practice$

Algorithm 4 $extractPatterns(P, V)$

Require: $\exists dataAnalysis(P, A, f, c)$ (Given a policy P, an Audit Schema (or a subset thereof) A, a frequency f and a condition c, perform data analysis)
1: $A \leftarrow$ *get attributes from Audit Schema* (may also be sent to any subset of Audit Schema)
2: $f \leftarrow$ *system-defined threshold frequency* (by default set to 5)
3: $c \leftarrow$ *system defined condition* (by default set to $COUNT(DISTINCT(User) > 1)$
4: $Patterns \leftarrow []$
5: $Patterns[] \leftarrow$ **dataAnalysis**(P, A, f, c)(see Algorithm 5)
6: **return** $Patterns$

Algorithm 5 $dataAnalysis(P, A, f, c)$

Require: $\exists executeQuery(SQL)$ (executes SQL statement and returns results)
1: *Split A into* $(Attr_1, .., Attr_n)$
2: $statement \leftarrow$ (**SELECT** $Attr_1, .., Attr_n$ **FROM** P's table **GROUPBY** $Attr_1, .., Attr_n$ **HAVING** $COUNT(*) > f$ **AND** c)
3: $results[] \leftarrow executeQuery(statement)$
4: **return** results

Even though refinement is an ongoing process, we assume that there is a *training period*, where a reasonable amount of information is collected in the

Algorithm 6 $Prune(Patterns, P_{PS}, V)$

Require: $\exists getCardinality(S)$ (returns the cardinality of a set S)
Require: $\exists getRange(P, V)$ (returns the range of the policy P according to the policy vocabulary V)
Require: $\exists getComplement(S_x, S_y)$ (returns the 'set complement'of S_x and S_y)
1: $range_x[] \leftarrow getRange(P_{PS}, V)$
2: $range_y[] \leftarrow getRange(Patterns, V)$
3: $usefulPatterns[] = getComplement(range_x, range_y)$
4: **return** $usefulPatterns$

audit log. This training period is totally dependent on the particular healthcare entity deploying the system.

Filter Algorithm 4.3 outlines the filter process. Given the schema in subsection 4.2 and the policy under examination, this process removes all rules that are not exception-based access entries. Given a more restrictive or totally different schema, the problem of separating violations from useful exceptions in an audit trail may require more sophisticated algorithms and even further research.

Extract Patterns In this step, the exceptions provided by the *Filter* phase, referred to as *Practice* in Algorithm 4.3, are analysed using a standard *data analytics* technique. The process is outlined in Algorithm 4.

To do the data analytics, a simple routine is called that takes a set of attributes, A, which is (a subset of) our audit schema, a minimum frequency, f, and a simple condition, c, translates it into a SQL statement and executes it on *Practice* to retrieve a list of entries that have occurred at least f and satisfy condition c (Algorithm 5). The technique finds the exact rules that have occurred more than f times. The data analysis routine has a well-defined interface that allows the *extractPatterns* algorithm to evolve and be easily customizable.

Prune Not all the patterns produced from the extraction phase may be good candidates for inclusion into P_{PS}. As a first step in determining these useful patterns, we implement a prune mechanism, Algorithm 6, that removes the patterns that are already present in P_{PS}. This is where our implementation of prune ends, because we recognize that some patterns may represent behavior that needs to be stopped. This implies that human input is prudent at this stage to determine which patterns are actually good practice and which should be investigated or terminated.

5 Use Case Scenario

We will now illustrate the use of PRIMA in a realistic healthcare use case scenario. We will refer to the system for which the policies tied to the policy store and audit logs have already been described in Section 3.3.

We have already defined the basic audit trail schema as {*(time,*t_j*), (op,*X_j*), (user,*u_j*), (data,*d_j*), (purpose,*p_j*), (authorized,*a_j*), (status,*s_j*)*}. Building on the

policy store P_{PS}, audit logs P_{AL} and policy vocabulary V used in Section 3.3, the audit trail generated by the system is shown in Table 1. Here we assume that the audit logs have been maintained for a period sufficient to be considered as the training period for this system and none of the exceptions reported in the logs are violations.

Time	**Op** (1:allow)	**User**	**Data** (Category)	**Purpose**	**Authorized** (Role)	**Status** (0:Exception)
t1	1	John	Prescription	Treatment	Nurse	1
t2	1	Tim	Referral	Treatment	Nurse	1
t3	1	Mark	Referral	Registration	Nurse	0
t4	1	Sarah	Psychiatry	Treatment	Doctor	0
t5	1	Bill	Address	Billing	Clerk	1
t6	1	Jason	Prescription	Billing	Clerk	0
t7	1	Mark	Referral	Registration	Nurse	0
t8	1	Tim	Referral	Registration	Nurse	0
t9	1	Bob	Referral	Registration	Nurse	0
t10	1	Mark	Referral	Registration	Nurse	0

Table 1. Audit trail, P_{AL}, for the system described in Figure 3

Invoking *ComputeCoverage*(P_{PS}, P_{AL}, V) on this snapshot of the audit logs reveals that the coverage has actually dropped to 30%. This is because the ratio of matching rules to total rules between P_{PS}, as per Figure 3, and P_{AL}, as per this snapshot, is now 3/10. In order to improve the coverage, we will run the *Refinement* algorithm. At line 1 of this algorithm, the *Filter*(P_{AL}) function filters out the log entries which are marked as non-exceptions, and therefore the *Practice* array now contains only the entries recorded at $t3$, $t4$ and $t6 - t10$.

The next step is to run data analytics to get the patterns that are candidate for inclusion in the policy. This is done at line 2 of Algorithm 2, when *extractPatterns*(P_{AL}, V) algorithm is called. As first steps in this algorithm, the relevant variables are set to enable data analysis ($A = \{data, purpose, authorized\}$, $f = 5$, $c =$"$COUNT(DISTINCT(User)) > 1$"). The output of the *dataAnalysis* (P_{AL}, A, f, c) routine returns those $(data, purpose, authorized)$ tuples in P_{AL} that occur at least 5 times. In this instance, the pattern is $Referral : Registration : Nurse$, i.e. tuples t3 and t7-t10.

As the last step, in line 3 of the *Refinement* algorithm, *Prune*$(Patterns, P_{PS}, V)$ is called to obtain the useful patterns from the ones in $Patterns$. The prune algorithm works by taking the ranges of both P_{PS} and $Patterns$ and then getting the 'set complement'of their intersection. This resulting set effectively contains those patterns that are not covered by existing rules in the policy store.

Thus, at the end of the *Refinement* algorithm, $Patterns$ contains $Referral : Registration : Nurse$ which is recorded in entries at t3 and t7-t10. This reveals that a Nurse accesses the Referral data for a patient too frequently for Registration reasons using the exception mechanism. Assuming that this is not a negative trend, then it suggests that a rule should be included in the policy stating that Nurses may be allowed to access patient Referral data for Registration purposes.

We are cognizant that the criterion used for pattern extraction, such as the threshold frequency of rules and numbers of users involved, is clearly subjective

and this scenario only serves to illustrate our approach and is not meant to be a definitive solution. The PRIMA systems will need to be configured and tuned as per the requirement specifications of the target environment. Secondly, simple data analytics techniques may not be sufficient in all cases. In order to enable a bit more sophisticated inference, we propose to leverage the frequent pattern mining algorithm [18] in our future work to detect correlations between attribute pairs that are not discovered by simple SQL queries.

6 Conclusion

In this paper, we formally introduced the problem of *policy coverage* in healthcare systems, which emerges from the over-reliance on the bypassing of security controls to access sensitive medical information, a phenomenon which is referred to in the medical community as *Break The Glass*. Our formalization is supported by PRIMA, a PRIvacy Management Architecture for healthcare systems, which addresses this problem of the circumvention of policy. PRIMA utilizes the actual practices of the organizations (embodied in the audit logs) to perform policy refinement. The system's advantages are that (i) it fits to the clinical workflow and does not require the workflow to fit to it, i.e. it does not impede the clinical workflow, (ii) it enables precise (or rather more realistic) definitions of purposes, criteria for exception-based accesses and categories of authorized users, and (iii) it enables improved privacy protection for the patient.

While emerging healthcare organizations leverage relational database systems, legacy systems employ hierarchical, XML-like structures. Thus, the natural evolution for PRIMA is to adapt the core concepts and technology to the tree-based structures.

References

1. Robert Pear. Warnings over privacy of us health network. New York Times, February 18, 2007.
2. L. Rostad and O. Edsburg. A study of access control requirements for healthcare systems based on audit trails from access logs. In *Proc. of the 2006 Annual Computer Security Applications Conference*, Miami Beach, FL, USA, December 2006.
3. Rebecca Wong. An overview of data protection laws around the world. *http://pages.britishlibrary.net/rwong/dpa.html*.
4. Ministry of Internal Affairs, Communications Information, and Communications Policy. Personal data protection law. *http://www.kantei.go.jp/jp/it/privacy/houseika/hourituan/index.html*.
5. Health insurance portability and accountability act, u.s. department of health and human services. *http://www.hhs.gov/ocr/hipaa/*.
6. Office of the Privacy Commissioner of Canada. Personal information protection and electronic documents act. *http://www.privcom.gc.ca/legislation/02_06_01_01_e.asp*.
7. Break-glass an approach to granting emergency access to healthcare systems. *http://www.nema.org/prod/med/security/upload/Break-Glass-Emergency_Access_to_Healthcare_Systems.pdf*.
8. United states presidential directive. *http://www.himss.org/CPRIToolkit/html/4.11.html*.

9. D. J. Hand, H. Mannila, and P. Smyth. Principles of data mining, aug 2001.
10. R. Agrawal, J. Kiernan, R. Shrikant, and Y. Xu. Hippocratic databases. In *Proc. of the 2002 Very Large Data Bases*, Hong Kong, China, June 2002.
11. IBM. Ibm hippocratic database active enforcement (version 1.0) : User's guide. http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/HDBEnforcementUserGuide.pdf.
12. IBM. Ibm hippocratic database compliance auditing (version 1.0) : User's guide. http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/HDBAuditingUserGuide.pdf.
13. B. Blobel. Authorisation and access control for electronic health record systems. *International Journal of Medical Informatics*, 73(3), 2004.
14. R. Anderson. A security policy model for clinical information systems. In *Proc. of the 1996 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 1996.
15. R. Bhatti, K. Moidu, and A. Ghafoor. Policy-based security management for federated healthcare databases (or rhios). In *Proc. of the 2006 International Workshop on Healthcare Information and Knowledge Management*, Arlington, VA, USA, November 2006.
16. A. C. Weaver, S. J. Dwyer III, and A. M. Snyder. Federated, secure trust networks for distributed healthcare it services. In *Proc. of the 2003 IEEE International Conference on Industrial Informatics*, Alberta, Canada, August 2003.
17. Ihe patient care coordination technical framework: Basic patient privacy consents, supplement 2005-2006, August 2006.
18. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 1994 Very Large Data Bases*, Santiago, Chile, sep 1994.