CERIAS Tech Report 2011-09

A Framework for Composition and Enforcement of Privacy-aware and Context-driven Authorization Mechanism for Complex Systems

by A M Samuel, M I Sarfraz, H Haseeb and A Ghafoor
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

A Framework for Composition and Enforcement of Privacy-aware and Context-driven Authorization Mechanism for Complex Systems

A M Samuel, M I Sarfraz, H Haseeb and A Ghafoor

Purdue University, West Lafayette, IN 47907

***Abstract*—Security and privacy of complex systems is a concern due to proliferation of cyber based technologies. Several researchers have pointed out that for the proper enforcement of privacy rules in a complex system, the privacy requirements should be captured in access control systems. In this paper, we present a framework for composition and enforcement of context-aware rules for such systems. The focus of this paper is the design of a system to allow a user (not a system or security administrator) to compose conflict free access control policies for his or her on-line assets. An additional requirement in this case is that such a policy be context-aware. We also present a methodology for verifying the privacy rules to ensure correctness and logical consistency. The verification process is also used to ensure that sensitive security requirements are not violated when privacy rules are enforced.**

***Index Terms*—Access control, role based, privacy rules, context**

1. INTRODUCTION

A complex system is defined as "one made up of a large number of parts that interact in a nonsimple way. In such systems, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole" [Sim62]. The complexity here stems primarily from the often unknown nature of interactions between different parts or components of the system and the consequent implications. Complex systems operate in a dynamic environment where security and privacy characteristics of contexts are different from one another and uniform access to secure and privileged resources/data everywhere and any time poses daunting challenges. The problem of designing, managing, and coordinating the myriad activities that make up complex systems such as economic institutions, cyber-physical systems, online financial systems, healthcare systems, or e-government applications has been central to the concerns of security and privacy.

Today, an increasing number of users are turning to the Internet to manage their personal information regarding finances, credit, health-care, travel, investments, employment history, etc. This trend is further being fueled by an ever-growing number of companies and government agencies such as banks, hospitals and employers, managing users' personal information in some form of on-line applications and databases. The aim is to save time and money, by streamlining and facilitating access to and manipulation of information on-line using the Internet both in a static and mobile environment. The overriding concern for using any internet-based service dealing with user's personal information is ensuring security and privacy of their personal information.

*This research is in part supported by an NSF grant number: IIS - 0964639

The development of privacy policies reflecting the privacy preferences expressed by the users for managing their information is a daunting task. The main concern in specifying privacy policies is how and by whom the policy is defined [Ard08]. With regards to who defines the privacy policies, our solution empowers the user to directly control his or her private information in terms of sharing with other parties in a private, secure and confidential environment. This is in contrast to "all or nothing" approach where an entity defines its privacy policies and a user can only accept or reject them according to his or her privacy preferences. With respect to the second issue, (i.e. how a privacy policy is defined), several researchers have pointed out that for the proper enforcement of privacy rules within an enterprise, the privacy requirements should be captured in access control systems which is a hybrid approach. The major contribution of this paper is to propose a hybrid approach where traditional access control policies can be integrated by adding a component that should specify privacy requirements (e.g., intentions of data use and so on) regulating under which circumstances the personal information of a user can be disclosed. The stand-alone approach, where privacy policies are defined as independent rules, is not a feasible approach for complex healthcare systems due to the numerous challenges posed in the form of scalability, heterogeneity of devices and mobility as explained later in section 3.

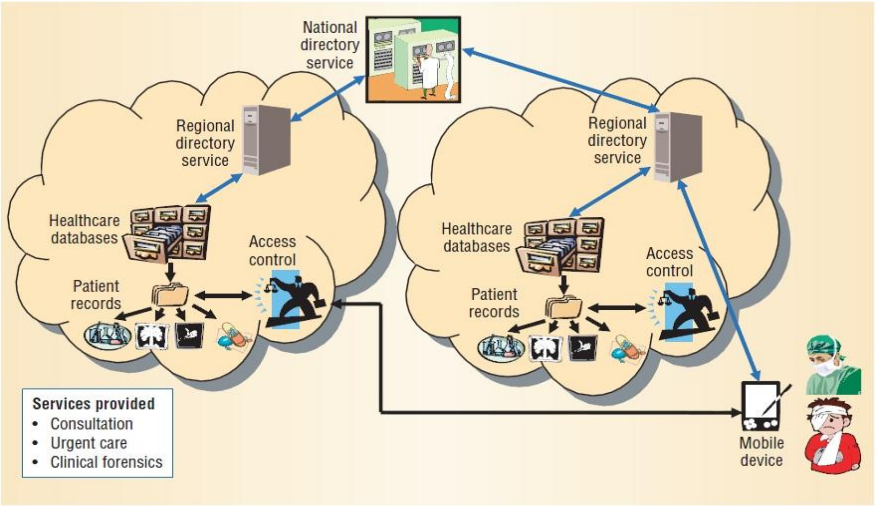

Fig. 1. A complex healthcare system

Figure 1 provides a high-level view of a complex system in healthcare domain. This distributed system, extensible to a higher level (e.g. from state-wide to nationwide), allows access to healthcare resources through appropriate datacenters. In this system, each datacenter consists of a regional directory service and a patient record database. When the regional datacenter receives a request, it queries its component regional directory service for the

patient's records. If the record exists within this datacenter, the regional directory service returns its location; otherwise the datacenter reroutes the request to the statewide directory service, which eventually returns the resource's location. When it identifies the resource location, the original datacenter submits an access request to the target datacenter housing the resource. The target datacenter follows its resource access-control and privacy policies with regard to releasing the requested information. In the healthcare domain such a system is vital to support the Personal Health Record (PHR) technology [Tan98, Mas02] which allows users the full-ownership of their Electronic Health Records (EHR) as enforced by HIPAA [Act96] in terms of access, management and sharing of their data across multiple healthcare providers (e.g. clinical practices, hospitals, pharmacies, etc). The key challenge behind such systems is to empower user to control his or her private information not only in terms of management and access but also allowing the sharing of their information with others whom they authorize, in a private, secure and confidential environment. The key tenet of such information sharing is that the decision to disclose personal information should entirely rest with the user. One of the barriers to wider use of such systems is the inability of the user to define context-aware disclosure and sharing rules for a collection of information from dispersed sources in a user friendly and consistent manner. Context is defined as "any information that can be used to characterize the situation of an entity" [Abo99]. For example time of day of a certain activity is a context parameter for that activity. Similarly, location of activity is also one of the context parameters. A high level view of supporting a privacy preserving PHR system is a *Secure User Data Repository System (SUDRS)*, the design of which is the theme of this paper. SUDRS, outlined in dashed line in Figure 2, provides a hybrid solution that allows the composition and enforcement of disclosure rules for personal data. These rules are specified by the contributors, consumers and owner of data.

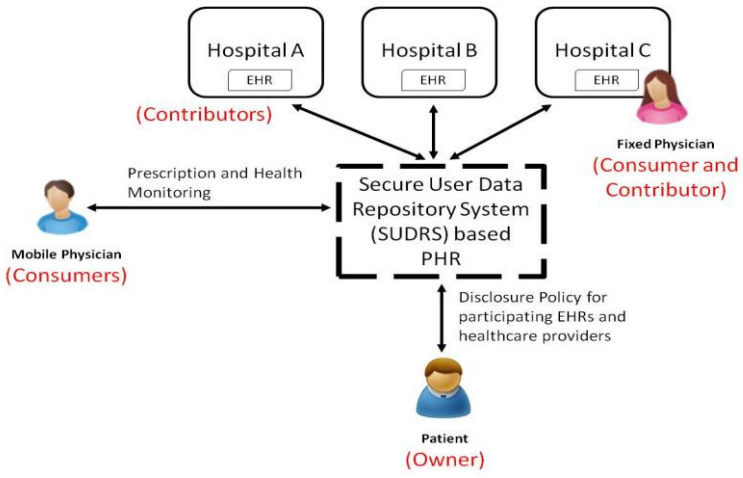

Fig. 2. An example of a privacy preserving Personal Health Record system

The broader scope of this paper viz-a-viz privacy preserving via the hybrid solution is the business-to-business (B2B) interactions in complex business systems and cloud computing. The evolving nature of B2B interactions that are no longer bounded by a traditional network perimeter or limited to traditional methods of data exchange and communication require fine-grained security and privacy controls. The complexity here arises from the fact that businesses today have global operations and numerous trusted partners with different business rules that require constant access to corporate resources. The need for security and privacy measures is further exacerbated because partners can now access corporate information and resources via methods, such as mobile and smart devices, and Web 2.0 technologies. It's not uncommon today to have business transactions and interactions "on the go" with the use of mobile devices and interactive media using Web 2.0 applications.

Cloud computing is a compelling case that businesses can't ignore for extending resources to B2B clients because of the economics, scale, and flexibility. The swarm of cloud-based services raises concerns about privacy and control due to the large scale, dynamic, and heterogeneous nature of clouds as organizations want more visibility into these cloud services.

The remainder of this paper is organized as follows. In Section 2, we present the related work in each of the areas we have addressed in our research. Section 3 illustrates the requirements that need to be met in order to ensure security and privacy management followed by the design of a generic architecture of the proposed SUDRS. Section 4 presents the design of a secure, private healthcare system. Section 5 concludes the paper.

2. RELATED WORK

The previous works most directly related to the areas we have addressed in complex systems are policy composition, requirement modeling and privacy-aware languages and models [Bas09, Hal08, Mcd09, Med08, Swa08]. Policy composition and requirement modeling by end-users implies that behavior of complex systems in response to both events and status be effectively modeled. The requirements of software systems are also known to be bifurcated as what is within the confines of the software system (operative requirements), and what is outside the confines of the software system (indicative requirements) [Zav97]. While time has a special meaning in terms of real time systems, where timeliness is an important system property, response to timed stimuli is also important to all other kinds of systems. In addition, with advances in mobile and ubiquitous computing, location has also become an important environmental parameter for policy composition. In order to understand the behavior of such systems from the

user's perspective, effective modeling of location contextual parameter, in addition to time, is required. While challenges of generic software requirement modeling are effectively addressed by many researchers, using state transition diagrams, context diagrams, visual representations and formal languages, these techniques do not address the issues related to development of context-aware privacy controlled policies by end-users.

Some of the notable work in a modeling environment for policies has been reported in [Vet02, Jur02] who propose extension of UML by defining stereotypes to evaluate diagrams and to indicate possible vulnerabilities. In [Eps99], researchers have proposed using UML to support role engineering. In [Shi00] use class diagram is used for static view of roles in policies, use-case diagram for the functional view and collaboration diagram for the dynamic model representing complex systems. In [Ray04] UML template class diagram is used to capture the structure of policies. An insightful comparison between policies and requirements can be found in [Ant01], along with guiding principles for composing policies with focus on software requirements. While the above mentioned work on composition of policies as software artifacts provides valuable insight into the issues surrounding policy engineering, they lack a clear methodology for a high level context-based policy requirement model which can be composed by an end-user with minimal abstraction of requirements.

Enforcement of privacy policies have been studied by several researchers. The Platform for Enterprise Privacy Practices (E-P3P) [Ash02] defines a fine-grained privacy policy model to formalize the desired enterprise internal handling of collected data. A particular data user is then allowed to use certain collected data for a given purpose if and only if the E-P3P authorization engine allows this request based on the applicable E-P3P policy. E-P3P is similar to APPEL [Con02] in the sense that both languages define the syntax and semantics for describing privacy control rules. However, they are designed for different purposes. APPEL is used by end users to describe their individual preferences. On the other hand E-P3P is used by enterprises to describe their internal privacy policies [Ash02]. Karjoth et al. [Kar02] extend Jajodia's Authorization Specification Language (ASL) [Jaj01], to include obligations and user consent. They also discuss a solution to automatically translate inner-enterprise privacy policy stated using E-P3P [Ash02] to publishable P3P policies for customers [Kar03]. The language has been formalized and refined to form IBM Enterprise Privacy Authorization Language (EPAL) [Ibm03]. Ni et al. propose P-RBAC [Niq10], a privacy-aware role-based access control model, which incorporates privacy policies into RBAC. They encapsulate data, action, purpose, condition, and obligation as privacy data permission. A permission assignment in P-RBAC is an assignment of privacy data permission to a role. A Purpose-Aware Role-Based Access Control model

(PuRBAC) is proposed by [Mas08] which extends RBAC by capturing privacy requirements of an enterprise by treating purpose as a central entity. The model assigns permissions to roles based on purpose related to privacy policies. Pearson et al. [Pea11] propose an approach that uses cryptographic mechanisms to strongly associate sticky policies with data for privacy management. In this approach, policies are defined and maintained for each data instance. Ardagna et al. [Ard08] present a privacy-aware framework in the context of the PRIME EU project that integrates access control policies together with data handling policies. The data handling policies primarily dictate the secondary use of personal data for the purpose of access control enforcement. Our solution empowers the user to control his or her private information not only in terms of management and access but also allowing the sharing of their information with others whom they authorize, in a private, secure and confidential environment. We also provide the user with the ability to define context-aware disclosure and sharing rules from dispersed sources in a user friendly and consistent manner. Moreover, we present a verification methodology to ensure that sensitive security requirements are not violated when privacy rules are enforced.

3. SYSTEM REQUIREMENTS AND DESIGN FOR PRIVACY AND SECURITY MANAGEMENT

In this section, we define the requirements related to privacy and security of complex systems and present an underlying architecture fulfilling the listed requirements.

3.1 DESIGN REQUIREMENTS

Complex systems face multifaceted security challenges such as heterogeneity, scalability, mobility and overall verifiability of underlying security framework. Heterogeneity implies specialized functionality of devices while, scalability deals with provisioning of security for large number of users and devices interacting across the whole system. In addition, complex systems should provide support for privacy requirements such as *Control, Collection, Intention Specification* and *Recording*. *Control* implies an organization which collects personal data should control access and use of information while *Collection* refers to collecting information in a disciplined fashion in conformance with the privacy policies. Personal data maintained by an enterprise should not be more than fulfilling the needs of the system. *Intention Specification* requires that data access requests must specify the intention or intentions for how and by whom the data is going to be used. Collected data must therefore be used only for specified intentions. *Recording* requires those who collect and release data should record each release or use to facilitate audits, inquiries and requests for access and revision.

3.2 PROPOSED SYSTEM DESIGN

The aforementioned security challenges are addressed by presenting a scalable generic architecture using a hybrid approach for the proposed SUDRS depicted in Fig 3. The architecture consists of four components and several databases. To deal with the context requirements and to address heterogeneity challenge related to privacy and security of complex systems, we use a Generalized Spatio-temporal Role Based Access Control (GST-RBAC) model [Sam07]. GST-RBAC uses the basic RBAC model by taking into account the environmental contexts, such as location and time to provide a comprehensive and generalized approach to security and privacy management. In other words, GST-RBAC is a spatio-temporal extension of RBAC, a de-facto model for specifying security requirements of large organizations. RBAC model consists of four basic components [Jos05] including a set of users/devices, a set of roles, a set of permissions, and a set of sessions. A user can be a human being, an autonomous agent, a task, a physical device or a subsystem [Woo10]. A role is a collection of permissions needed to perform a certain job function within an organization. Permission is an access mode that can be exercised on objects in the system, and a session relates a user to possibly many roles. RBAC model differentiates itself from traditional access control models in that the permissions in RBAC are not directly associated with users, but with roles. Roles are created by security administrators to reflect the various functional categories of users within an enterprise. Users are then assigned membership to roles, and these roles are in turn assigned permissions. Grouping of users to assigned roles in RBAC is termed as user-role assignment and grouping of permission to roles is termed as permission role assignment. Such grouping provides a scalable mechanism, which is major advantage of RBAC over other authorization approaches [Jos05]. Also, RBAC is distinguished by its inherent support for principle of least privilege which requires a user to be given no more privileges than necessary to perform a task. Role enabling constraint in GST-RBAC defines the relationship between locations and time for which the role can be enabled. A role is enabled at a certain location and time, while it is not enabled at other locations or other times. Spatial constraints in GST-RBAC take location and temporal constraints take time as a context parameter validating an access request. The GST-RBAC model allows specification of spatial and temporal constraints on role enabling, user-role assignment, temporal constraints on role enabling, user-role assignment, and role-permission assignments, activation, runtime events, constraint enabling expressions and triggers as mentioned in detail in [Jos05]. Further details of GST-RBAC model along with temporal and spatial constraints can be found in [Sam07].

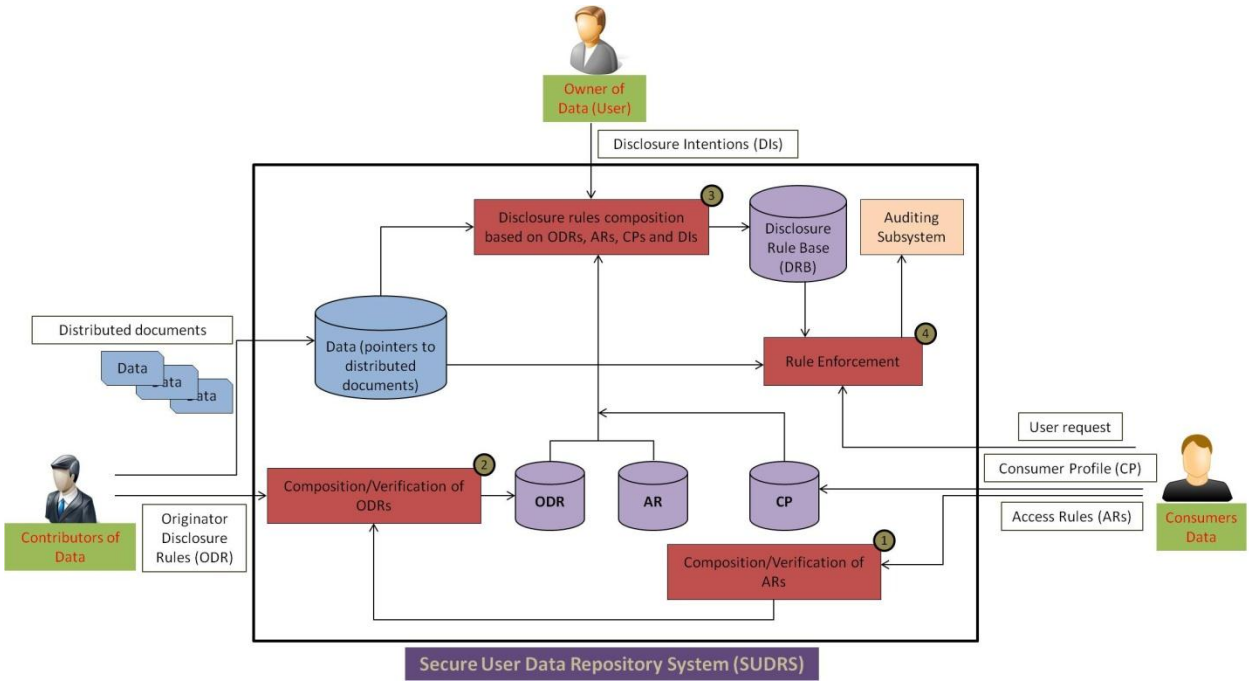

Fig. 3. An architectural view of the proposed system

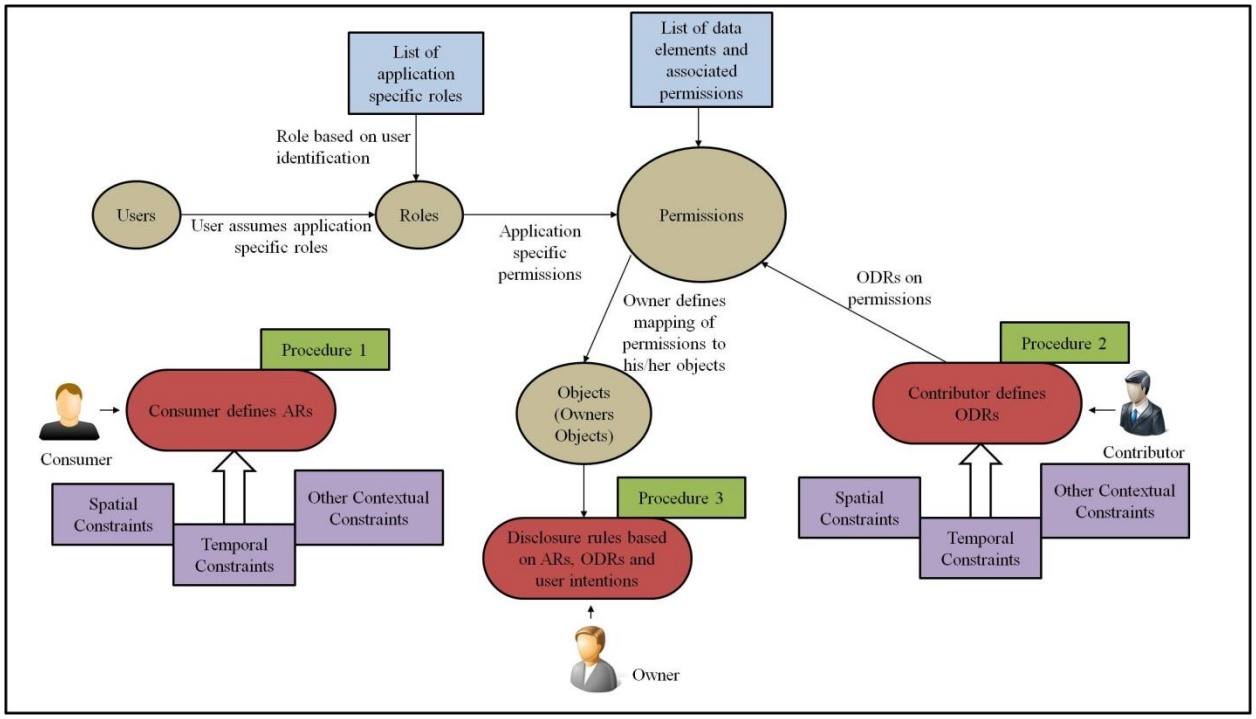

Fig. 4. Functional view of the proposed system

We now describe the high-level functions of Fig. 3 and then discuss the functionality of the components in later sections. Functional overview of components of Fig. 3 is depicted in Fig. 4. The proposed SUDRS allows the composition and enforcement of disclosure rules for personal data maintained by it. These rules are specified by the contributors, consumers and owner of data. The contributors of data are the originators of data generated as a consequence of the user's interaction with contributors. An example in this case is a health-care provider (doctors, nurses etc.) who generate health related information concerning a patient (owner) and contribute this information to be stored in the SUDRS. In this case, healthcare provider is a contributor and patient is the owner of data. In addition to contributing data, the contributor also generates Originator Disclosure Rules (ODRs) for each element of data.

The consumers of data in Fig. 3 are entities interested in accessing data. Consumers can also be contributors of data. For example, physician who is a consumer is adding records to patient profile thus acting also as a contributor. Consumers provide Access Rules (ARs) to the SUDRS defining the access times, locations, and other context parameters that govern a particular access for generic data types. An example in this regard is a physician defining the times of day as well as the location of access (e.g. from his clinic) for viewing pathology reports of a particular patient. Alternatively, this access by the physician can be from a remote location using a PDA or an Internet enabled mobile devices. The consumers in Fig. 3 also provide their profile information (Consumer Profile (CP)) such as name, credentials and affiliation.

The owner of data in Fig. 3 composes disclosure rules based on his or her Disclosure Intentions (DI), ODRs, ARs and CPs. For example, a patient can specify that the physician can access patient's pathology report only from hospital and only at certain times. These spatio-temporal access rules are then stored in the Disclosure Rule Base (DRB) thus constituting a privacy driven GST-RBAC policy. Any access to the user's personal data by the consumers is thus evaluated by the GST-RBAC model against the stored rules and access is granted accordingly. The evaluation process is described later in this section. The required data is extracted from the data storage system and sent to the consumer. Each request, either satisfied or denied, is also logged in the auditing subsystem. This logging allows the owners to track accesses to their personal records as well as evaluate accesses not satisfied by the disclosure rules in the system. Additionally, the audit subsystem keeps track of all disclosure rules that allowed access to a certain part of data with a view of facilitating the owner to adapt disclosure rules. The following are the four critical components depicted in Fig. 3 that enable SUDRS to maintain privacy and security of data.

1. Component 1(Composition and verification of Access Rules (ARs)): This component allows consumers of data to compose and verify access rules held in the SUDRS.

2. Component 2 (Composition and verification of Originator Disclosure Rules (ODRs)): This component facilitates the contributor to compose and verify the *ODRs related* to data contributed to the SUDRS.

3. Component 3 (Composition and verification of disclosure rules): Component 3 enables the composition and verification of disclosure rules by owners based on ODRs, ARs, and User Intentions (UIs).

4. Component 4 (Enforcement of disclosure rules): Component 4 evaluates request of data stored in UDR based on context of request.

3.2.1 Component 1 (Composition and Verification of Access Rules (ARs)):

Component 1 in Fig 3 enables consumers to define their access rules. Such specification involves defining and verifying rules consistency. The process of Component 1 consists of a series of inter-related steps depicted in Figure 5 and abstracted in the form of *ComposeVerifyAccessRule* algorithm in Table 1.

Algorithm ComposeVerifyAccessRules
Input :*Consumer.Credentials, Consumer.Input*
Output : SUDRS.*AccessRules, SUDRS.ConsumerProfile*

```
1. IF (authenticate_consumer(SUDRS,Consumer.Credentials)=TRUE) THEN
2. ConsumerProfile.Input=Consumer.Input
3. SUDRS.ConsumerProfile=consumerprofile(ConsumerProfile.Input) THEN
4. AccessRules.LocationProfile =location_profile(Consumer)
5. AccessRules.ContextualParameters=contextual_parameters(Consumer)
6. AccessRules.TemporalInformation=temporal_information(Consumer)
7.     IF (verification_ar(AccessRules)=TRUE) THEN
8.     SUDRS.AccessRules=save_ar(AccessRules)
9.     RETURN SUDRS.AccessRules
9.     ELSE
10.    correct_violations_ar(AccessRules)
11.    END
12. END
```

```
Functions used are defined as follows:
1) consumerprofile(cp)   returns consumer profile
   generated  by input cp
2) location_profile(user) returns location profile
   generated by input user
3)  temporal_information(user)  returns  temporal
   profile created by input user
4)  verification_ar(rules) returns Boolean TRUE or
   FALSE based on verification of input rules
5) save_ar(rules) saves input rules and returns
   memory address to saved input rules in AR
   database
6) correct_violations_ar(rules) allows the user to
   correct input rules failing verification
```

Table 1: Algorithm for Composition and Verification of Access Rules

According to this algorithm, the consumer enters personal information such as credentials for authentication. The function *authenticate_consumer* in Line 1 of *ComposeVerifyAccesRules* algorithm (Step 1 of Fig 5) uses this information (*Consumer.Credentials*) to identify the consumer in the SUDRS. On successful authentication, the

consumer is allowed composition of access rules. Subsequently, the consumer selects roles from the GST-RBAC system that are predefined in the SUDRS. Examples of pre-defined roles in a PHR implementation can be Primary Physician, Radiologist, etc. A consumer's assignment to specific role implies access privileges defined by the owners in the disclosure rules (Component 3 of Fig. 3). The consumer enters: (a) temporal constraints such as, times and days of the week or month, for which the consumer is available to access data from the SUDRS, and/or (b) location information such as GPS coordinates or IP address, from where the consumer can access the data. These locations, together with the temporal constraint define the spatio-temporal context of access preferences by the consumer. Component 1 also allows the consumer to enter any other contextual parameters such as system properties or environmental conditions under which data is accessed from SUDRS. In this case, context parameter can indicate the type of device the consumer intends to use to access data. The entered information from the consumer in the access rules represented by data structure *AccessRules* in the algorithm is verified for consistency in Line 7 of algorithm and step 6 of Fig. 5. The properties to be verified are listed in Table 2. Verification is done using standard model checking techniques as discussed in [Hol97]. Verification involves the evaluation of safety and liveness properties listed in Table 2. In case of any violations, the user is directed to rectify the errors in the access rules. On successful verification, ARs are saved in the database. The resulting rules can be saved in any one of the following standards: Text, XML, relational database, etc.

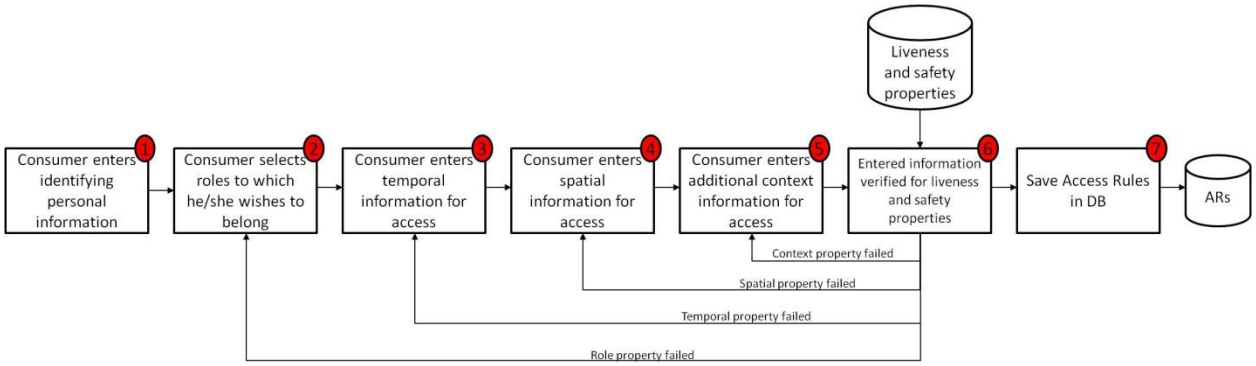

Fig. 5. Steps for composition and verification of access rules by the consumer

In addition to storing and verifying ARs from the consumer, the process in Component 1 allows entry of profile information by the consumer. This information is called Consumer Profile (CP). CP includes individually identifying information such as name, position in an organization, credentials such as qualifications of a physician

etc. CP is generated by the function *consumer_profile* in the algorithm *ComposeVerifyAccessRules* in Table 1and is used in Component 3 to compose individualized disclosure rules.

No	Property	Explanation	Type
1	Reachability of Role	No user defined for a role	Safety
2	Reachability of Role	User selects a role to be part of	Liveness
3	Reachability of Permissions	Permission not granted to a role	Safety
4	Reachability of Permissions	Permission granted for a role	Liveness
5	Conflicting Constraints	Define two similar (e.g. temporal or spatial) contextual constraints opposing each other	Safety
6	Conflicting Constraints	Define two dissimilar contextual constraints opposing each other	Safety
7	Access Leakage	Same permission (access to same object) associated with roles of different users (owner)	Safety
8	Access leakage	Permissions for different users granted to the same role	Safety

Table 2. Sample Liveness and Safety Properties for Verification of ARs

3.2.2 Component 2 (Composition and Verification of Originator Disclosure Rules (ODRs)):

Data can be contributed to the SUDRS electronically as well as manually. Process of Component 2 allows the composition of ODRs related to data being uploaded to the SUDRS in electronic format. The underlying process utilized by Component 2 is depicted in Figure 6 and corresponding algorithm, *ComposeVerifyOriginatorDisclosureRules* is shown in Table 3.

Algorithm ComposeVerifyOriginatorDisclosureRules

```
Input    : Contributor.Data, SUDRS
Output   : SUDRS.OriginatorDisclosureRules
1.     SUDRS.Data=upload(Contributor.Data)
2.     SUDRS.Field= extract_fields(SUDRS.Data)
3.     ODR=compose_odr(SUDRS.AccessRules,SUDRS.ConsumerProfile,SUDRS.Field)
4.           IF (verification_odr(ODR)=TRUE) THEN
5.           SUDRS.OriginatorDisclosureRules=save_odr(ODR)
6.           RETURN SUDRS.OriginatorDisclosureRules
6.           ELSE
7.           correct_violations_odr(ODR)
8.           END
```

```
Functions used are defined as follows:
1) upload(data) returns address to uploaded input data
2) extract_fields(data) returns list of label fields extracted from input data
3) compose_odr(ar,cp,field) returns ODR,Originator Disclosure Rules composed by access rule
    input ar, consumer profile input cp and data fields input field.
4) verification_odr(rules) returns Boolean TRUE or FALSE based on verification of input
    rules
4) save_odr(rules) saves input rules and returns memory address to saved input rules
5) correct violations odr(rules) allows the user to correct input rules failing verification
```

Table 3: Algorithm for Composition and Verification of Disclosure Rules

The process of Component 2, highlighted in Fig. 6 and algorithm *ComposeVerifyOriginatorDisclosureRules* in Table 3, facilitates the contributor to upload data to the SUDRS. The interface provided by Component 2 allows upload of data as in number of formats such as simple text (with data labels), excel sheets, word documents (with labels conforming to predefined document elements), XML, multimedia content (video, images, etc). Fields from the uploaded document are extracted by the function *extract_fields* in *ComposeVerifyDisclosureRules* algorithm in Table 3. This extraction is based on pre-defined labels in the files as show in Step 2 of Fig. 6. The ARs and CPs collected by the SUDRS (Component 1) are used to formulate the ODRs composed by the contributor. The contributor is presented with a user interface which lists all the extracted labels and a selectable list of roles, users, time and location. All these parameters are retrieved from the AR and CP base. The contributor forms a list by selecting corresponding roles, users, time and location. The information is verified for consistency by function *verification_odr* in Line 4 of algorithm in Table 3. The generic liveness and safety properties are defined before hand and are stored in a database. The properties verified are depicted in Table 2. The verified ODRs are stored in the ODR base to be used by Component 3 for composition of the disclosure rules.

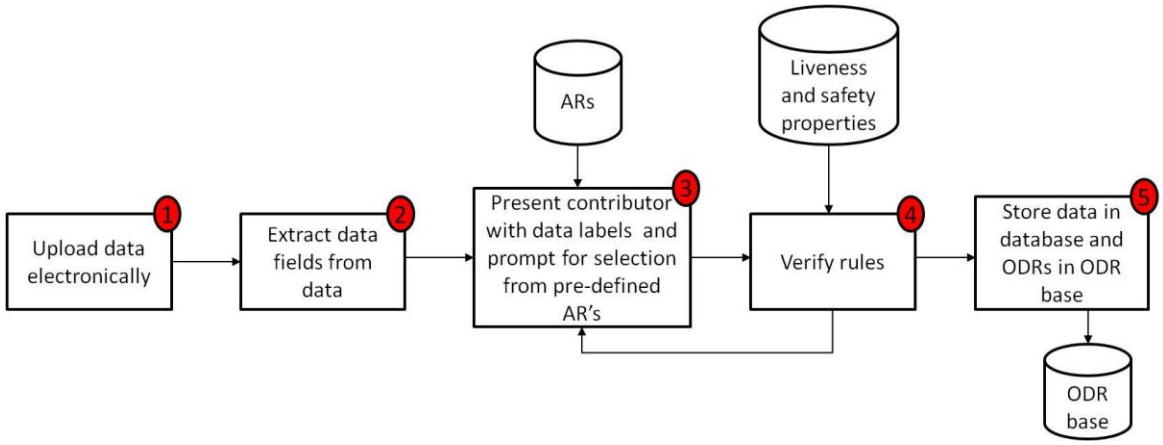

Fig. 6. Process for composition and verification of ODRs

3.2.3 Component 3 (Composition and Verification of Disclosure Rules):

This component allows disclosure rule composition and verification. The owner selects disclosure intentions at each step of the composition process, prompting the component to verify the new intentions/purposes and guiding the owner through series of step. Process in Component 3 is implemented by *ComposeVerifyDisclosureRules* algorithm depicted in Table 4 and depicted in Fig. 8.

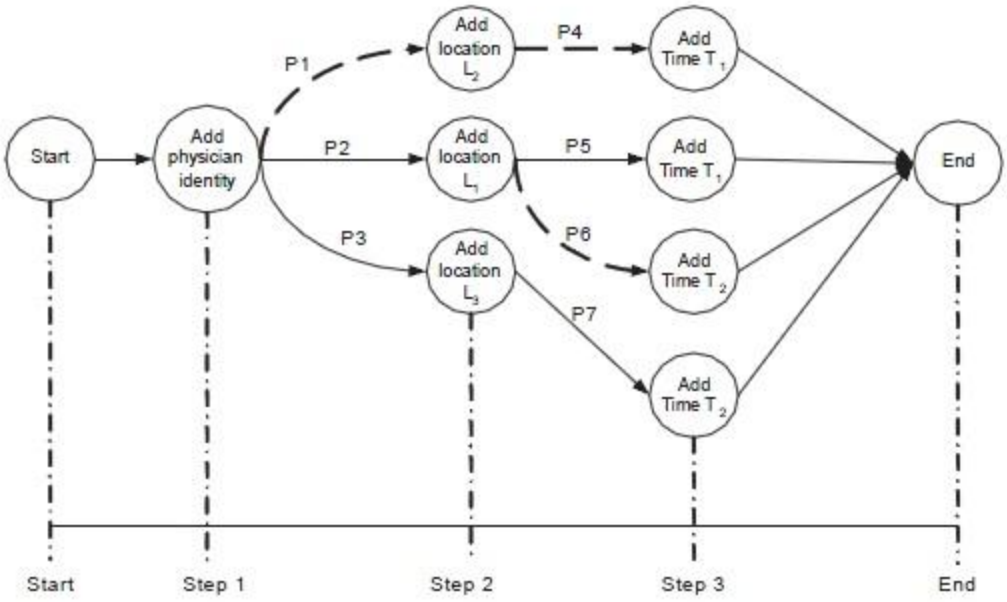

Figure 7. An example of disclosure policy verification algorithm

The composed disclosure rules are verified by the function *save_disclosure_rulebase* in Line 5 of *ComposeVerifyDisclosureRules* algorithm. The function *save_disclosure_rulebase* utilizes the Disclosure Policy Verification Algorithm (DPVA) for a step-wise verification of the disclosure policy before saving it in the policy base. The various steps of DPVA are depicted in Figure 7. Starting at Step 1, a contributor/consumer is added to the disclosure policy from a list consisting of contributors and consumers interacting with the system. The liveness property from Step 1 to Step 2 indicates that the selected consumer or contributor is available at locations L1 and L3. Similarly, the safety property from Step 1 to Step 2 indicates that the selected contributor/consumer is not available at location L2. DPVA suggests viable paths to the owner by displaying the locations at which the selected contributor/consumer can access the desired data. The owner can select one of the viable options. From Step 2 to Step 3, the liveness properties imply that time T1 is associated with location L1 and time T2 is associated with location L3. The safety properties in this step are to check association of T1 with location L2 and T2 with location L1. The owner selects one of the viable paths and DPVA inhibits the owner from pursuing non-viable paths. Finally, the owner is given the choice of selecting path P2-P5 or P3-P7, both being viable. This step-by-step verification at the composition time allows the owner to go back to the previous step and express alternate intentions. In essence, DPVA assists the owner to intelligently compose disclosure policies based on the contextual constraints stored by

SUDRS. The overall process involved in Component 3 is depicted in Figure 8. The details of the process are provided below.

Algorithm `ComposeVerifyDisclosureRules`
Input : *SUDRS*
Output : *DRB*

```
1.      DRB.Data=select_drb_data(SUDRS.Data)
2.      DRB.ConsumerList= select_consumer(SUDRS.OriginatorDisclosureRules)
3.      DRB.LocationProfile=select_locationprofile(SUDRS.AccessRules)
4.      DRB.TemporalInformation =select_temporal_information(SUDRS.AccessRules)
5.      save_disclosure_rulebase(DRB)
6.      RETURN DRB
```

```
Functions used are defined as follows:
1) select_drb_data(data) returns address to input data
2) select_consumer(odr) returns consumer list of input odr
3) select_locationprofile(ar) returns location profile of input ar
4) select_temporal_information (ar) returns temporal information of input ar
5) correct_violations_odr(rules) allows the user to correct input rules failing verification
```

Table 4: Algorithm for Composition and Verification of Disclosure Rules

Through Component 3, the owner initially selects data elements to share with consumers as in step 1 in Fig. 8 and Line 1 of Table 4. The selection of data elements can be done at a general level (e.g. all radiology data) or at a specific level (by selecting a specific X-ray image). This list of data elements is based on data related to the owner uploaded by the relevant contributors. This component allows the owner to select potential consumers to whom thr selected data can be released. The list of consumers in this case is based on the ODRs defined by the contributor. In case the ODRs do not limit disclosure of data to any consumer, a complete list of consumers (in ARs) is presented. The owner selects the location of access for the selected consumer. The list of locations is also entered by the consumer at the time of composition of ARs. The owner can select the time of access as well. Again, the listed times are provided by the consumer while composing ARs. Note, the location- time pair is pre-verified as part of the process followed by Component 1. In case no change is required by the owner to disclosure rules, the rules are saved in the DRB.

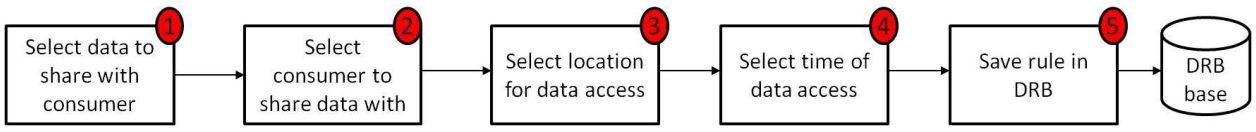

Fig. 7. Process for capturing user intentions to compose disclosure rules

3.2.4 Component 4: Enforcement of Disclosure Rules

Component 4 comprises the enforcement of disclosure rules on the data requested by the user (consumer, contributor or owner). The enforcement decision depends on the context parameters extracted from the request as well as from context sensors (such as system clock). The process involved in implementing Component 4 is depicted in Fig. 9 and the implementing algorithm *EnforceDisclosureRules* in Table 5.

Algorithm EnforceDisclosureRules
Input :*Consumer.Credentials, Consumer.Input*
Output :Consumer.Data

```
1.      IF (authenticate_consumer(SUDRS,Consumer.Credentials)=TRUE)
2.        IF (AR=access_request(Consumer.Input,Consumer.Credentials,Consumer.ContextualParameters)
          THEN
3.        Consumer.Data=show_data(AR)
4.        RETURN Consumer.Data
4.        ELSE
5.        deny_request(Consumer)
6.        END
7.      END
```

Functions used are defined as follows:
1) authenticate_consumer(SUDRS,user)authenticates by returning Boolean TRUE or FALSE based on input user and SUDRS.
2) access_request(userinput,usercred,usercontextparam) evaluates access request based on input userinput, usercred and usercontextparam and returns Boolean TRUE or FALSE.
3) show_data(ar) returns data based on input ar
4) deny_request(consumer) denies request to resource to user represented by input consumer

Table 5: Enforcement of Disclosure Rules

The request submitted by the consumer contains context parameters such as IP address or GPS to provide location, information, user credentials etc. These parameters and additional context parameters are extracted from this request. The disclosure rules held in the DRB are retrieved and are compared with the context parameters. The decision to whether or not grant the request by the consumer is also taken in Line 2 of Algorithm *EnforceDisclosureRules* in Table 5. On successful evaluation, relevant data is retrieved from the data stored in the SUDRS. As a result, the data conforms to the privacy intentions defined in the disclosure rules by the owner under the current context information. The relevant data is sent back to the consumer depicted in step 5 of Fig. 9. In case, of denial of the access request, a relevant error message is delivered.

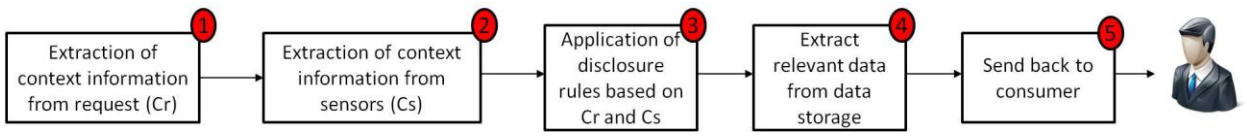

Fig. 9. Steps for enforcement of disclosure rules based on context of request

4 A N EXAMPLE OF A CONTEXT-AWARE, SECURE AND PRIVATE HELATHCARE SYSTEM

In this section we present the design of Intelligent Privacy Manager (iPM), a Context-aware, GST-RBAC based SUDRS for sharing and collaboration of Electronic Health Records (EHR) in a secure and private manner. (A prototype of iPM SUDRS can be accessed at http://cobweb.ecn.purdue.edu/~dmultlab/lab). iPM is based on the SUDRS design proposed in Section 3 for secure and private information sharing in healthcare domain.

4.1 Salient Features of iPM

Access to iPM SUDRS by an owner/contributor is provided based on user id and password. For a consumer to access the system, consumer can present his or her credentials in the form of a SAML assertion [Oas03]. Based on IP address of consumer request, the system identifies the location of the requester by a database lookup using IP2Location [Ipl01]. Logging-in allows the:

1) owner to view all EHR held by the SURDS along with sharing options provided by the *Asset Manager*. All EHR are divided into categories and types of files. Additional records can be uploaded by selecting "Open file manager" on the top.
2) owner/contributor to manage EHR held in the form of files by the *File Manager*. File Manager allows an owner/contributor to manipulate EHR in terms of uploading new content (files), deleting content and files.
3) consumer to define his or her sharing options with owner/contributor using *Profile Manager*. The consumer defines sharing options separately for each owner. The logged in owner's/contributor's sharing conditions and consumer's conditions may not match, as they may indicate different time, location or day a particular access can take place.
4) owner to manage his or her preferences with consumers using the *Sharing Manager*. Sharing Manager facilitates the owner, who in conjunction with the system, decides about the individual/devices accessing the related data under various context. The owner defines groups of consumers and associates data to be accessed by each group. The proposed system at the time of composing disclosure rules assists the owner in deciding the most suitable sharing times, locations, and days that can allow all members of a group to get access data. The owner can use these suggestions to segregate consumers (e.g. two consumers can view a lab report only at two different locations) as well as data. At all times, the proposed system prompts the owner with the best options for sharing, which the owner may accept or reject.

5) owner to view access log for auditing purposes provided by *View Access Log*. View Access Log allows the owner/contributor to review the list of users/devices accessing his or her data along with time, location and day of access.

4.2 Design Requirements of iPM

Below we present a set of use cases along with related design requirements which are the foundations of iPM SUDRS development for healthcare domain.

TABLE 6
USE CASES AND RELATED DESIGN REQUIREMENTS

Use Case	Description	Design Requirements	Description
1	A physician (consumer) defines his or her sharing profile. This sharing profile consists of location from where he or she will access a particular patient's (owner) on-line records.	Context-aware sharing relationship	R1: A physician specifies his location from where he or she can access a patient's EHR.
			R2: A physician specifies his time of day (e.g. from 9 AM to 5 PM) when he or she can access a patient's EHR.
			R3: A physician specifies his day of week (e.g. Tuesday, and Friday) when he or she can access a patient's EHR.
2	A patient uploads content to iPM SUDRS and defines its category. All items in the collection of EHR are partitioned according to this category. All items are held in collections (or files).	Partitioned on-line records	R4: On-line records are partitioned based on their semantics (e.g. health records and billing statements are separate).
			R5: All on-line records are held in files.
3	A patient defines new groups to contain his or her physicians.	User roles	R6: A patient can define roles to which physicians can be part of.
4	A patient populates each group with physicians and designates conditions (location, time and day of week) when a particular physician can be part of the group. These conditions will be a subset of a patient's sharing profile with a physician. The selected conditions will be evaluated against physician's sharing profile to arrive at the best possible fit.	User roles assignment	R7: A patient defines conditions under which a physician can be assigned to a (group) role. These conditions are that of location, time and day of week.
5	A patient defines a collection of filess for each group of physicians to access.	Permission role assignment	R8: A patient defines permissions to a file which a member of a role can access.
6	A physician accesses patient's EHR by first becoming part of a group and then gaining access to an associated data. Access to group and data is subject to location, time and day of access.	User assumes a role	R9: A physician becomes part of a group (role) based on matching the conditions of access defined in R7.

		User accesses a permission	R10: A physician accesses a file and its content based on matching the conditions of access defined in R8.
7	All access to a patient's EHR are audited and presented for further analysis.	Auditing	R11: All access to patients data and groups are logged.

4.3 User Policy Representation in iPM

In this section we present an example of an XML-based Generalized Spatio Temporal Role Based Access Control (X-GSTRABC) policy. [Bha05, Sam07]. X-GSTRBAC, uses XML-based policy specification language, and is an extension of the RBAC model suitable for addressing the access management challenges in federated systems. X-GSTRBAC language specification is captured through a context free grammar called X-Grammar, which follows the same notion of terminals and non-terminals as in Backus-Naur Form (BNF), but supports the tagging notation of XML that also allows expressing attributes within element tags. The use of attributes helps maintain compatibility with XML schema syntax, subsequently serving as the type definition model for X-GSTRBAC language.

Pertinent portions of the example policy generated based on the aforementioned Use Cases are discussed next. An example of the user sheet depicting user's credential (termed as XUS in [Bha05]) is given in Table 7 (Part A). Note, that the credentials associated with user John include information about location (in this case any Operation Room), time of day, qualification, and membership. These credentials are used to satisfy the sharing requirements outlined in Use Case 1. Role sheet [termed as XRS in Bha05] is depicted next (Table 7 - Part B) with role id as well as role cardinality. A user can define as many roles as required. The Physician role could only be enabled if the temporal (week one of first quarter of the year) and spatial constraints (area represented by code 786) specified required for access are satisfied. A user is assigned to a role using a user-to-role assignment sheet, termed as XURAS [Bha05], and is shown in Table 7. The assignment constraints in this example (Part C) are location, time and credentials. A user request coming in from the specified location at specified time with certain credentials is allowed to access resources allocated to this role. On-line records are associated with roles in the permission-to-role assignment sheet (XPRAS in Table 7) along with conditions that must be satisfied to effect this association. In this sheet the role Physician is assigned permission P1. The permission P1 is defined elsewhere in the policy.

TABLE 7
EXAMPLE POLICY SHEETS

A. XML User Sheet (XUS)

```
<XUS xus_id="HL7_XUS">
<User user_id="U1">
<UserName>John</UserName>
<CredType cred_type_id="c1" type_name="physician_cred">
<CredExpr><Attribute name="employee_id">HL7123</Attribute>
<Attribute name="membership">AMA</Attribute>
<Attribute name="qualification">PHD</Attribute>
<Attribute name="location">Operation room</Attribute>
<Attribute name="time">13:00</Attribute>
</CredExpr></CredType>
</User>
</XUS>
```

B. XML Role Sheet (XRS)

```
<XRS xrs_id="HL7_XRS">
<Role role_id="Phy" role_name= "Physician"/>
<Cardinality>2</Cardinality>
<EnabConstraint>
<EnabCondition pt_expr_id="PTFirstQuarterWeekOne"/>
<EnabCondition spatial_constraint_id="47906"/>
</EnabConstraint>
</Role>
<Role role_id="Nur" role_name="Nurse">
<Cardinality>5</Cardinality>
<EnabConstraint><EnabCondition pt_expr_id="PTQuarterAllWeekAll"/>
</EnabConstraint>
</Role>
</XRS>
```

C. XML User to Role Assignment Sheet (XURAS)

```
<XURAS xuras_id="HL7_XURAS">
<URA ura_id="uraPhy" role_name="Physician">
<AssignUsers><AssignUser user_id="U1">
<AssignConstraint>
<Attribute name="location">John Hopkins Hospital</Attribute>
<Attribute name="time">8-15</Attribute>
<Attribute name="cred_type">c1</Attribute>
</AssignConstraint>
</AssignUser></AssignUsers>
</URA>
</XURAS>
```

D. XML Permission to Role Assignment (XPRAS)

```
<XPRAS xpras_id="HL7_XPRAS">
<PRA pra_id="praPhy" role_name="Physician"
<AssignPermissions>
<AssignPermission prem_id="P1">
<AssignConstraint>
<AssignCondition Cred_type="c1">
<LogicalExpr op="AND"><Predicate>
</Predicate></LogicalExpr>
</AssignCondition></AssignConstraint>
</AssignPermission>
</AssignPermissions>
</PRA>
</XPRAS>
```

5 CONCLUSION

In this paper, we have presented design of a system for composing and enforcing context-aware disclosure rules for preserving privacy and security. The proposed system allows an on-line user to compose disclosure rules which are consistent and have been verified for a set of verification properties. We also present design of iPM (the Intelligent Privacy Manager) prototype developed to implement the above mentioned design. Its various components and resulting policy is also presented. Pertinent research challenges which have surfaced as a result of this research are as follows:

- An intuitive and user friendly GUI is an implementation challenge that needs to be addressed.
- Though the use of XML and grammar based composition of user disclosure rules allows maximum portability of the resulting policy, the issues related to portability of accompanying user data from one information sharing site to another is a major privacy concern for the user, a challenge that needs to be researched.

REFERENCES

[Abo99] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *HUC*, 1999, pp. 304-307.

[Act96] A. Act, "Health insurance portability and accountability act of 1996," *Public Law*, vol. 104. 191.

[Ant01] A. I. Anton, J. B. Earp, C. Potts, and T. A. Alspaugh, "The role of policy and stakeholder privacy values in requirements engineering," in *5th IEEE International Symposium on Requirements Engineering*, Toronto, Canada, 2001, pp. 138-145.

[Ard08] C. A. Ardagna, M. Cremonini, S. De Capitani Di Vimercati, and P. Samarati, "A privacy-aware access control system," *Journal of Computer Security*, vol. 16. 369-397, 2008.

[Ash02] P. Ashley, S. Hada, G. Karjoth, and M. Schunter, "E-P3P privacy policies and privacy authorization," presented at the Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, Washington, DC, 2002.

[Bas09] D. Basina, M. Clavelb, J. Dosera, M. Egeac, "Automated analysis of security-design models," *Information and Software Technology*, pp. 815-831, 2009.

[Bha05] R. Bhatti, A. Ghafoor, E. Bertino, and J. Joshi, "X-GTRBAC: an XML-based policy specification framework and architecture for enterprise-wide access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8. 187-227, 2005.

[Con02] W. W. W. Consortium, "*A P3P Preference Exchange Language 1.0 (APPEL1.0)"*. Available: http://www.w3.org/TR/P3P-preferences/ (accessed 26 Sept, 2011).

[Eps99] P. Epstein and R. Sandhu, "Towards a UML based approach to Role Engineering," in *Proceedings of RBAC'99: 4th ACM Workshop on Role-Based Access Control, 28-29 Oct. 1999*, New York, NY, USA, 1999, pp. 135-43.

[Hal08] C. Haley, R. Laney, J. Moffett, B. Nuseibeh, "Security Requirements Engineering: A Framework for Representation and Analysis," *IEEE Transcations on Software Engineering*, pp. 133-153, 2008.

[Hol97] G. J. Holzmann, "The model checker SPIN," *IEEE Transactions on Software Engineering*, vol. 23. 279-95, 1997.

[Ibm03] IBM, "*The enterprise privacy authorization language"*. Available: http://www.zurich.ibm.com/security/enterprise-privacy/epal/ (accessed 26 September 2011).

[Ipl01] IP2Location, "*IP2Location"*. Available: http://www.ip2location.com/ (accessed 26 September, 2011).

[Jaj01] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian, "Flexible support for multiple access control policies," *ACM Transactions on Database Systems*, vol. 26. 214-60, 2001.

[Jos05] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17. 4-23, 2005.

[Jur02] J. Jurjens, "Using UMLsec and goal trees for secure systems development," in *Applied Computing 2002: Proceeedings of the 2002 ACM Symposium on Applied Computing, March 11, 2002 - March 14, 2002*, Madrid, Spain, 2002, pp. 1026-1030.

[Kar02] G. Karjoth and M. Schunter, "A privacy policy model for enterprises," in *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*, 2002, pp. 271-281.

[Kar03] Karjoth G., M. Schunter, and M. Waidner, "Platform for enterprise privacy practices: privacy-enabled management of customer data," presented at the Proceedings of the 2nd international conference on Privacy enhancing technologies, San Francisco, CA, USA, 2003.

[Mas08] A. Masoumzadeh and J. Joshi, "PuRBAC: Purpose-aware role-based access control," *On the Move to Meaningful Internet Systems: OTM 2008*. 1104-1121, 2008.

[Mas02] D. Masys, D. Baker, A. Butros, and K. E. Cowles, "Giving patients access to their medical records via the Internet," *Journal of the American Medical Informatics Association*, vol. 9. 181, 2002.

[Mcd09]P. McDaniel, S. McLaughlin, "Security and Privacy Challenges in the Smart Grid," *IEEE Security and Privacy*, pp. 75-77, 2009.

[Mel08] D. Melladoa, E. Fernández-Medinab, M. Piattinib,"Towards security requirements management for software product lines: A security domain requirements engineering process," *Computer Standards & Interfaces*, pp. 361-371, 2008.

[Niq10] Q. Ni, E. Bertino, J. Lobo, C. Brodie, C. M. Karat, J. Karat, and A. Trombetta, "Privacy-Aware Role-Based Access Control," *ACM Transactions on Information and Systems Security*, vol. 13. 24 (31 pp.), 2010.

[Oas03] OASIS, "*Security Assertion Markup Language (SAML)"*. Available: http://xml.coverpages.org/saml.html (accessed 26 September, 2011).

[Pea11] S. Pearson and M. Casassa-Mont, "Sticky Policies: An Approach for Managing Privacy across Multiple Parties," *Computer*, vol. 44. 60-68, 2011.

[Pow02] C. S. Powers, P. Ashley, and M. Schunter, "Privacy promises, access control, and privacy management. Enforcing privacy throughout an enterprise by extending access control," in *Electronic Commerce, 2002. Proceedings. Third International Symposium on*, 2002, pp. 13-21.

[Ray04] I. Ray, N. Li, R. France, and D.-K. Kim, "Using UML to visualize role-based access control constraints," in *Proceedings on the 9th ACM Symposium on Access Control Models and Technologies, SACMAT 2004*, Yorktown Heights, NY, United states, 2004, pp. 115-124.

[Sam07]A. Samuel, A. Ghafoor, and E. Bertino, "A Framework for Specification and Verification of Generalized Spatio-Temporal Role Based Access Control Model," *CERIAS Technical Report*, Purdue University, 2007.

[San98] R. Sandhu and Q. Munawer, "How to do discretionary access control using roles," *Proceedings of the ACM Workshop on Role-Based Access Control*. 47-54, 1998.

[Shi00] M. E. Shin and A. Gail-Joon, "UML-based representation of role-based access control," in *Proceedings of WET ICE 2000. 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 14-16 June 2000*, Los Alamitos, CA, USA, 2000, pp. 195-200.

[Sim62] H. A. Simon, "The architecture of complexity," *Proceedings of the American Philosophical Society*, vol. 106. 467-482, 1962.

[Swa08] N. Swamy, B. J. Corcoran, M. Hicks, "Fable: A Language for Enforcing User-defined Security Policies," *IEEE Symposium on Security and Privacy*, pp. 369-383, 2008.

[Tan98] P. C. Tang and C. Newcomb, "Informing Patients," *Journal of the American Medical Informatics Association*, vol. 5. 563, 1998.

[Vet02] M. Vetterling, G. Wimmel, and A. Wisspeintner, "Secure systems development based on the common criteria: The PaIME project," in *Proceedings of the 10th ACM SIGSOFT Symposium on the Foundations of Software Engineering*, Charleston, SC, United states, 2002, pp. 129-138.

[Woo10]W. J. Wook, H. M. Jin, L. C. Gyeong, and Y. H. Yong, "Dynamic Role-Based Access Control with Trust-Satisfaction and Reputation for Multi-agent System," in *IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, Los Alamitos, CA, USA, 2010, pp. 1121-6.

[Zav97] P. Zave and M. Jackson, "Four dark corners of requirements engineering," *ACM Transactions on Software Engineering and Methodology*, vol. 6. 1-30, 1997.