Protecting Privacy of Health Information through Privacy Broker

Jaijit Bhattacharya
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
jaijit@iitd.cse.ernet.in

S.K.Gupta
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
skg@iitd.cse.ernet.in

Bhurvi Agrawal
Oracle HP E-Governance Centre of Excellence
Oracle, India
bhurvi@iitd.cse.ernet.in

Abstract

Concerns about the protection of personally identifiable information are not unique to the health care industry; however, consumers view their medical records as more "private" than other information because involuntary disclosure can affect jobs or health insurance status. EPAL is a convenient way of capturing privacy policies. This paper presents an EPAL based privacy middleware architecture called Privacy Broker which attempts to reduce privacy violation risk and enforce committed privacy policy in health information system. This paper also provides a translation of SQL queries into Authorization Request in EPAL.

1. Introduction

Health information and the medical record reveal some of the most intimate aspects of an individual's life. In addition to diagnostic and testing information, the medical record includes the details of a person's family history, genetic testing, history of diseases and treatments, history of drug use, sexual orientation and practices, and testing for sexually transmitted diseases. Subjective remarks about a patient's demeanor, character, and mental state are sometimes a part of the record.

Over time, health information has come into use by many organizations and individuals who are not subject to medical ethics codes, including employers, insurers, government program administrators, attorneys and others. As uses of medical information multiply, there is a need to have regulatory protections for this highly sensitive and deeply personal information.

Computerization can reduce some concerns about privacy in patient data and worsen others, but it also raises new problems. Computerization increases the quantity and availability of data and enhances the ability to link the data, raising concerns about new demands for information beyond those for which it was originally collected. The potential for abuse of privacy by trusted insiders to a system is of particular concern.

At the same time, accurate and comprehensive health care information is critical to the quality of health care delivery, and to the physician-patient relationship. Many believe that the efficacy of the health care relationship depends on the patient's understanding that the information recorded by a physician will not be disclosed. Without these assurances, many patients might refuse to provide physicians with certain types of information needed to render appropriate care.

In order to protect the concerns of patients about privacy of data, Health Insurance Portability and Accountability Act (HIPAA) [11] was passed by Congress in 1996 to set a national standard for electronic transfers of health data. Such legislations implies that Medical Information System must implement reasonable privacy policies and procedures that limit how much protected health information is used, disclosed, and requested for certain purposes. It must also limit who within the entity has access to protected health information, and under what conditions.

The Privacy Broker proposed by Bhattacharya and Gupta [7] provided automated support for enforcing privacy policies related to purpose, limited retention, safety and compliance. However, this Privacy Broker is based on a modification of P3P [20]. P3P is typically characterized as supporting machine-readable policies; it does not provide an enforcement mechanism for organizations to use in monitoring their information handling practices. Therefore, this

0-7695-2507-5/06/$20.00 (C) 2006 IEEE

paper uses EPAL [4], a formal language, to develop a privacy middleware architecture based on Privacy Broker [7]. IBM's Enterprise Privacy Authorization Language (EPAL) addressed the need for machine-enforceable policies. EPAL is an XML-based language, designed for organizations to specify to internal privacy policies. These EPAL Policies can be used internally and amongst the organizations and its business partners to ensure compliance with underlying policies of each partner.

The Privacy Broker is particularly important in Medical Information System. It maintains the privacy of medical records, so that patients trust their health care provider.

This paper also provides a translation of SQL queries into EPAL authorization-request [4]. EPAL authorization-request consists of user-category, data-category, action and purpose. This information is passed on to the EPAL Rule Engine that determines which rule evaluates against the request. If the rule exists, then access to data is allowed. Translation has been done for the ease of the users of Medical Information System, so that they can query data in SQL, which is a popular language to retrieve data from database.

2. Current Approaches to Privacy

The issue of privacy has been addressed from several directions. Primary among them has been to (a) statistical databases, (b) define a privacy specification language, and (c) ensuring that the database itself ensures privacy [2].

2.1. Statistical Databases

Research in statistical databases was primarily focused on hiding individual data while enabling statistical information to be extracted. Such information was in the form of sum, average, count, maximum, minimum, pth percentile, etc. ([1][17]).

A large amount of work has also been done in the area of access control and security ([10] [14][21]). Whenever sensitive information is exchanged, it must be transmitted over a secure channel and stored securely. Other relevant work includes efforts to create tools and standards that provide platform for implementing a system such as what this research plan proposed ([20] [16]).

However, Statistical Databases does not help organizations in adhering to the legal requirements like purpose specification, limited retention etc. For example, a statistical database will not automatically remove the credit card data of a customer in an organization, after the payment has been received (principle of limited retention).

2.2. Privacy Specification Language

The Platform for Privacy Preferences Project (P3P) provides a standard to websites to communicate their data practices [20]. It provides the syntax and semantics of privacy policies and the mechanisms for associating policies with Web resources. It includes machine-readable privacy policy syntax that web browsers and other agent tools can use to fetch P3P privacy policy automatically.

The specification includes

–A standard vocabulary to describe a web site's data practices

–A set of base data elements that web sites can refer to on their P3P policy

–A protocol for requesting and transmitting web site privacy policy

However, the privacy specification language does not support implementation of the stated privacy policies. Therefore, even if an organization uses P3P to specify its privacy policies, it has no mechanism of enforcing these policies within the organization. Moreover, it does not allow personalization of privacy policies and merely helps in specifying the generic privacy policy in a machine understandable form. P3P also does not provide either enforcement or non-repudiation of the privacy policy. For example, a *data subject* may consent to provide data under one privacy policy. However, that policy may be changed without notice later and the data provided by the data subject may get compromised.

P3P also does not provide a means to establish a negotiated contract beyond APPEL (A P3P Preference Exchange Language). There is no mechanism for non-repudiation of agreements between data subject and websites. For example, if the data was provided by the data subject under a given privacy policy, there is no mechanism to uphold that in a court of law when the organization changes its privacy policy and violates data privacy under the previous privacy policy. Therefore, P3P is not able to help organizations adhere to the legal requirements of privacy.

2.3. Privacy Protection within Databases

This approach attempts to enable database systems to enforce privacy. Attempts at such solutions are basically based on Statistical databases [1] [17] and Secure databases [8].

Oracle 9i database has implemented privacy [10] using a combination of techniques that allow a higher

granularity of control at tuple level as well as at column level. The key mechanisms are as follows:

- Strong authentication and single sign-on: Strong authentication is generated by PKI infrastructure that uses industry standard X.509 digital certificates for strong authentication
- Granular Access control through views: A view is a subset of one or more tables. However, views have issues of scalability and complication in administration of security and privacy.
- Virtual Private Database (row level control): VPD enables, within a single database, per user or per group data access with the assurance of data separation. By dynamically appending SQL statements with a predicate (a "where" clause), VPD limits access to data at the row level and ties security policy to the table itself.
- Label-Based Access Control: The label security mediates access to data by comparing a sensitivity label on a piece of data with label authorizations assigned to an application user.
- Secure Application Role: It ensures that the appropriate conditions are met before the user can excursive privileges granted to the role in the database. This limits the bypassing of the application to directly access the database.
- Encryption in the database: Oracle supports DES (56 bit) and triple DES (112 and 168 bits) encryption of the records.

However, Oracle 9i's solution is not a tool dedicated for privacy but it is a tool that facilitates privacy-enabled implementations. It suffers from drawbacks such as the solution being database dependent thus making it difficult to migrate data to another database. Also it neither has a Capability migration mechanism nor can it handle complex privacy policies.

Another approach is that of Hippocratic databases [2] that uses components of Secure database and introduces privacy control within the database itself.

The Hippocratic database uses Privacy Metadata, which is defined as:

- External recipients
- Retention period
- Authorized users

This metadata is used to manage the privacy of the data.

This information is split into two conceptual tables (a) Privacy-policy and (b) Privacy-authorization.

However, such an approach makes the solution wedded to the database and hence requires fundamental changes in the Kernel of the database. This makes it difficult to be deployed on existing databases. Moreover, it does not allow individuals to authorize specific individuals to access their data (for e.g. Individuals might need to give access to their hospital health records to their family physician or employer). Also, it does not support any mechanism to establish a negotiated contract beyond APPEL (A P3P Preference Exchange Language). It also has no mechanism for non-repudiation of agreements between visitors and websites. Neither does it support Privacy Policing of a site.

2.4. Ensuring transactional Privacy using encryption and co-processors

This approach uses an uncompromised program (eg IBM 4758 programmable secure coprocessor) as a broker for all database transactions. The uncompromised program encrypts the stored data with its private key and signs the outgoing data with its private key again [12] [18] [19] [21].

Alternatively, privacy of data collection is ensured by using a direct encrypted connection between the database and the user's client [15].

However, in both these approaches, it only takes care of some aspects of privacy and faces the shortcomings mentioned earlier.

2.5. Implementation of Privacy Specified by Policy

This approach uses an uncompromised program (eg IBM 4758 programmable secure coprocessor) as a broker for all database transactions. The uncompromised program encrypts the stored data with its private key and signs the outgoing data with its private key again [12] [18] [19] [21].

Alternatively, privacy of data collection is ensured by using a direct encrypted connection between the database and the user's client [15].

However, in both these approaches, it only takes care of some aspects of privacy and faces the shortcomings mentioned earlier.

An important aspect of implementation of privacy policies is to capture the privacy policies itself. Significant work has been done in that area Batra et al [6].

This approach of capturing policies for managing databases uses a layered architecture for a policy based data administrator. The policies are defined by the decision makers/ data administrator using a friendly graphical user interface and then these policies are modeled as ECA (Event-Condition-Action) like rules. Before these policies are stored in a database, they are verified at various levels so that conflicts do not arise at both creation and execution time. The events triggering the policy execution could be from within the database, i.e. internal events, or they could be from outside the database, i.e. external

events. These policies are then executed by a policy engine, based on the specified event and on satisfaction of the corresponding condition.

However, this approach helps in capturing user policies and converting them into machine-readable policies; it does not provide a solution for implementing the policies themselves.

3. Privacy Broker for Privacy Preserving Transactions

Given the drawbacks of the existing approaches to privacy, our work focused on an independent layer that would be dedicated to managing privacy. The proposed Privacy Broker [7] for privacy preserving transactions aimed to enable the following aspects of privacy without modifying the database kernel. (a) The Broker should be able to accept the agreed privacy specification and ensure adherence of the stated privacy policies, (b) it should enable individuals to authorize specific individuals to access their data and (c) it should also be able to enforce non-repudiation of agreements between visitors and web-sites.

Such a Broker-based approach ensures that the solution is independent of the database used. It also allows the solution to be easily used in legacy systems.

3.1. Broker Architecture

The proposed Broker [7] uses an uncompromised program as a broker for all database transactions [12] [18] [19] [21]. The uncompromised program encrypts the stored data with its private key and signs the outgoing data with its private key again. All data accesses are through "Capability Certificates" [7].

In this paper, we continue with the name "Sentry" for the Privacy Broker, as was used by Bhattacharya and Gupta [7]. The Capability certificates are stored in Sentry. Sentry also stores the decryption key-pair. The Capability certificate contains a description of how to evaluate the "worthiness" of requests for data, and policies on what computation can be allowed to be done to select and/or process traffic data for a given requester.

The stored tuples are encrypted with Sentry's encryption key-pair. Requests for access to the stored data are given to Sentry who can then (1) evaluate if the request is worthy of being honored, (2) compute what data is to be released and (3) perform whatever computation is needed on that data to produce the final result, which it then signs and releases (Figure 1).

Capability certificates will allow a suitably authorized person to access privacy constrained data.

- The capability certificates would allow the appropriate policy to be executed, fetching the required data
- Transfer of capability would allow the temporary user to access data for a pre-specified time period.

Therefore, the capability certificates needs to have the signed and encrypted privacy policy and access code. This would facilitate non-repudiation and allow data stored in the repository to be encrypted by a single key (the secure broker's key) allowing secure access of privacy-constrained data to multiple parties. It would also eliminate the issue of dependency maintenance.

The broker also records all the queries in a query log in order to use them for detection of possible privacy violation and attempts for privacy violation using data mining.

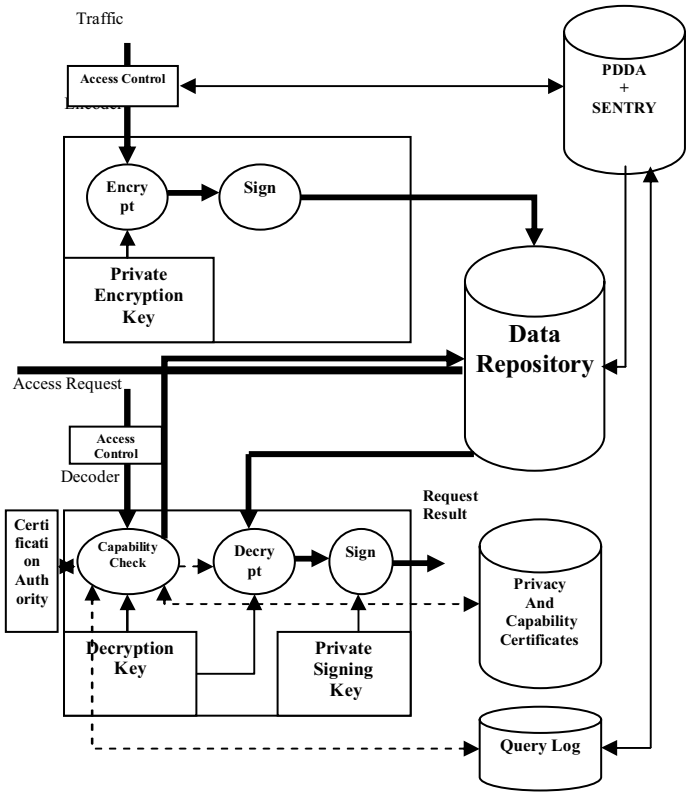

Figure 1. Architecture of Privacy Broker, Sentry

3.2. Database Schema for the Privacy Broker

A health care provider's privacy policy often reflects different legal regulations, promises made to

patients, as well as more restrictive internal practices of the health care provider. Further it may allow patient preferences. The heath provider's internal privacy policy is captured in the Capability Certificates [7], which define the capability of user to access privacy constrained health data. Privacy preferences are stored in Privacy Policies. Privacy Broker separates provider's privacy policy from patient privacy policy. Any change in health care provider's policy will not lead to change in patient privacy policy.

The schema implemented considers Privacy Policies [7] that declare the privacy requirements of records and Capability Certificates that are enforceable for users. Since Privacy Policies are declaration of the privacy requirement, EPAL specifications have been used for defining Privacy Policies. Capability Certificates [7] requires Transfer Group in order to allow transfer of capabilities from one individual to another individual or group [7]. For example, if a Doctor needs to go on leave, her privacy access rights can be temporarily transferred to his Assitant-Doctor for a fixed duration. However, a Doctor under any circumstances will not be allowed to transfer her privacy access rights to receptionists. However EPAL does not support the concept of Transfer Group. Hence we need to modify EPAL to support this requirement. Also, we have modified the obligations to allow obligations to be applicable on the EPAL rule itself. This would allow certain parts of the EPAL rule to be modified to support Capability transfer. The Broker [7] has been developed on MySQL using Java.

3.3. Case Study

We take the case of a medical information system based on work of Krekke [13].Personnel working at the hospital, using the medical information system deployed in the hospital, can access personal identifiable information (PII) about patients for diagnosis and treatment.

However, different patients will have different privacy policy for providing his or her information to different groups of people. Here patients may agree to reveal their health related records to specialists for case-study. However, they may not allow the same record to be viewed by receptionists. Thus for each record, the individual can define his or her privacy requirements. This gets captured in the Privacy Policies, with each privacy policy having a unique Privacy Policy ID. Hence, along with patient's medical history, the Privacy Policy ID needs to be stored as shown in Table 1.

Table 1. Medical history with Privacy Policy-ID

Patient ID	Present Illness	Medical History	Family History	Privacy Policy ID
N0984	Diabetes Mellitus, Type II	Hyper-Tension, Sweating	Mother-diabetes	PP11

Therefore, records that have the same privacy policy will have the same Privacy Policy ID. As a result, the number of privacy specifications files gets reduced to a practically implementable level. The Privacy Policy ID identifies the privacy policy. The privacy policy needs to be defined in the privacy policy table (Table 2). The privacy policy is defined in EPAL.

The Privacy Policy ID gets tagged to each individual record.

Table 2. Privacy Policy table

Privacy Policy ID	EPAL Policy
PP11	<epal-policy ID =”PP11”> <rule id=”R1”> …. </rule> </epal-policy>

Once the data has been recorded, different users, within the hospital, will have different capabilities for accessing the data. For example, a doctor, John, having read and write access to contact-data, medical-history, physicians order, and progress notes of the patient's medical journal for the purpose of diagnosis and treatment. This is captured as Capability Certificates [7]. However, the size of the database would go up significantly if for each user, a separate Capability Certificate is defined. Hence the mechanism adopted to reduce the number of such Capability Certificate [7] (and hence to reduce the complexity of implementation) is to limit the policies to a few user groups. For example, a set of users who are doctors will fall into one group and another set of specialists, will fall into another group. Such groups are termed as user-groups.

In order to capture the privacy requirements of each individual, they are clustered into user-groups and the personal id or the user id is linked to each user-group in table as shown in Table 3.

Table3. Personal ID mapping to User-Group

Personal ID	User-Group
PID1234	Doctor

The actual privacy policy of each user-group is then captured as capability certificate in the capability certificate table as shown in Table 4.

Table 4. Capability Certificate table

User-Group	Capability-Certificate
Doctor	<epal-policy id =”Doctor”> <rule > </epal-policy>

4. Language for the Certificates

The Privacy Policy and the Capability Certificates [7] are defined using modified EPAL language.

4.1.. Privacy specification

Consider a patient Alex, suffering from ulcers and his medical history says that, he is diabetic. He may allow other specialists in the department to view his medical history for research work but not his operative-report. Also, he allows nurse on duty to view his operative-report. The Privacy policy with name,”PP11”, will have Privacy Policy ID, “PP11” which helps in faster retrieval of privacy specification file. The privacy policy file is shown in Figure 2.

```
<epal-policy id=”PP11” >
  <rule id =”Rule1” ruling=”allow” >
    <user-category refid=”Specialist”/ >
    <data-category refid =”Medical history “/>
    <purpose refid=”Research”/>
    <action refid =”Access”/>
    <obligation refid =”Retention>
        <parameter refid= Days><value>7</value>
        </parameter></obligation>
  </rule>
  <rule id =”Rule2” ruling=”allow”>
    <user-category refid=”nurse”/ >
    <data-category refid=”operative-report”/>
    <purpose refid=”observation”/>
    <action refid =”read”/>
  </rule>
</epal-policy>
```

Figure 2. Example of Privacy Policy

4.2. Capability specification

A patient’s treating doctor must be given access to all fields in the medical-journal, except non-medical information such as home address and telephone number. The advising doctor only needs access to the fields in the medical journal that is necessary for diagnosing or treating the patient. This can be captured in the modified EPAL policy. We define a transfer-group, which specifies the capability of a user-group to transfer his capability to a set of user-group. A GUI is provided to the user to transfer his capability to some other user in his transfer-group and to specify expiry of this transferred capability. In order to specify the transfer-group, we add a new element to the EPAL vocabulary, <transfer-group>.

In the example, transfer group is used as when a doctor is going on a vacation and he transfers his capability to an assistant-doctor. This capability transfer is done by adding additional rule to capability certificate of the user-group. The expiry of capability can be temporal or event driven which is to be captured in the obligation part of the rule. Therefore, the doctor’s transfer-group will constitute the assistant-doctor group. In capability certificate of assistant-doctor group, doctor can add an additional rule in which the condition part will specify the id of the assistant doctor to whom the capability is transferred and the obligation will specify when this rule will be deleted from the capability certificate. An example of capability certificate of doctor is shown in Figure 3.

```
<epal-policy id=”doctor-group”
    default- ruling=”allow”>
  <rule id="Rule1" ruling="allow">
    <user-category refid="doctor" />
    <data-category refid="contact-data" />
    <data-category refid="medical-history" />
    <data-category refid="physicians-order" />
    <data-category refid="progress-notes" />
    <purpose refid="treatment" />
    <purpose refid="diagnosis" />
    <action refid="write" />
    <action refid="read" />
    <transfer-group refid=”assistant-doctor”/>
    <condition refid="isTreatingDoctor">
    <obligation refid="log">
    </obligation>
  </rule>
</epal-policy>
```

Figure 3. Example of Capability Certificate

Addition rule in capability certificate of assistant doctor will be as shown in Figure 4.

```
<rule id="RuleTemp1" ruling = "allow">
  <user-category refid="assistant-doctor" />
  <data-category refid="medical-history"/>
  <purpose refid="treatment" />
 <action refid="read"/>
 <condition refid="assistant-id-112/>
 <obligation refid ="Expiry-of-rule">
     <parameter refid=days>
        <value>7</value>
     </parameter>
 </obligation>
</rule>
```

Figure 4. Example of EPAL Rule with modified obligation

Thus privacy can be dynamically managed under different circumstances without re-coding any part of the database or the privacy broker. As in this case, rule is added by the doctor, and is deleted by the mechanism for executing modified obligation.

5. Implementation

For the implementation of the proposed Privacy Broker [7], a modular approach, shown in Figure 5 was adopted. This architecture maps a subset of the functionalities of the architecture in Figure 1.

The brief function of each layer is discussed below.

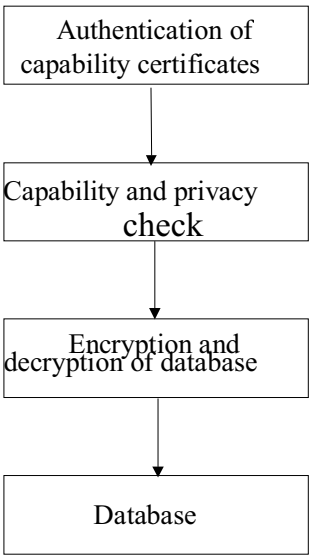

Figure 5: Layered implementation architecture

5.1. Authentication Layer

This module works on the top of our database layer. The main function of this module is the authentication of user's capability. While submitting a query, user first logs in this module. After login, the capability certificate of a user is fetched from the capability certificate database. Since this capability certificate is encrypted with a secure co-processor's key, it needs to be decrypted.

As we have mapped the certificates to a particular set of users, we do not need to store certificates for individuals.

5.2. Capability-privacy compliance check layer

After authentication of the capability certificates, the second layer performs the important function of capability checking. In our architecture, capability certificates and privacy policy are specified using an XML based language EPAL. This layer serially checks the list inclusion of capability in the privacy.

5.3. Encryption-Decryption layer

This layer ensures that even a database administrator cannot violate a user's privacy policy. This module encrypts the data while storing in database and decrypts the data while retrieving from database.

6. Translation of SQL Query in EPAL Authorization Request

In our implementation, a modified SQL is used to access request wherein the regular SWL request is appended with a *purpose* attribute in the where clause of SQL. Authorization Request [4] contains user-category, data-categories, action, purpose and containers containing context data for evaluation of condition as specified in [4]. We propose a mapping of SQL request into EPAL Authorization Request for the ease of the user. Only simple queries have been handled since they form the major class of queries required while accessing the data. Note that user-category is retrieved from the login information of the user.

So, we only need to find the data-categories, action and containers, if they exist, from SQL statement. We again consider the medical information system to show the mapping between this modified SQL and EPAL.

Consider a receptionist, Patricia, requesting to insert contact-data of the patient in Hospital Database. She states, *"Insert into Contact-Data" values ("Alex", "203", "22567788") where purpose="Registration'"*. Mapping is shown in Table 5.

Table 5. Mapping of Insert Statement

Modified SQL Request and Login Info	EPAL Equivalent	Value
Login Information	User-Category	Receptionist
Relation	Data-Category	Contact-Data
Purpose in 'Where' Clause	Purpose	Registration
Insert into	Action	Store

Patricia, will delete the record from Contact-Data Table when, Alex, is discharged. She enters, *"Delete from Contact-Data where Patient-Name='Alex' where purpose='Delete-Registration'"*. Mapping is shown in Table 6.

Table 6. Mapping of Delete Statement

Modified SQL Request and Login Info	EPAL Equivalent	Value
Login Information	User-Category	Receptionist
Relation	Data-category	Contact-Data
Purpose in 'Where' Clause	Purpose	Delete-Registration
Delete from	Action	Modify
Attribute list in 'Where' clause except purpose	Container Attribute-ID Value	Patient-Info Patient-name Alex

A surgeon, Sarah, enters, *"Update Operative-Reports set Result='Successful' where purpose='Treatment' and Patient-Id='PID1234'"*. Mapping is shown in Table 7.

Table 7. Mapping of Update Statement

Modified SQL Request and Login Info	EPAL Equivalent	Value
Login Information	User-Category	Surgeon
Attribute-List	Data-Category	Result
Purpose in 'Where' Clause	Purpose	Treatment
Update	Action	Modify
Attribute list in 'Where' clause except purpose Login Name	1.Container Attribute-Id Value 2.Container Attribute-ID Value	Patient-Info Patient-ID PID1234 User-Info User-Name Sarah

Gary, as a doctor, enters, *"Select Family-History from Medical-History where Patient-Id='PID1234' and Purpose='Diagnosis' "*. Mapping is shown in Table 8.

Table 8. Mapping of Select Statement

Modified SQL Request and Login Info	EPAL Equivalent	Value
Information from Login	User-Category	Doctor
Attribute-List	Data-Category	Family-History
Purpose in 'Where' Clause	Purpose	Diagnosis
Select	Action	Access
Attribute list in 'Where' clause except purpose Login Name	1.Container Attribute-ID Value 2.Container Attribute-ID Value	Patient-Info Patient-ID PID1234 User-Info User-Name Ajay

Consider the case of request of data access from two tables. Gary wants to read the medical-history and progress-notes of Patient, with ID,"PID1234". He enters, *"Select * from Medical-History, Progress-Notes where Medical-History.PatientId = Progress-Notes. Patient-Id and Patient-Id = 'PID1234' and purpose = 'Treatment'"*. Mapping is shown in Table 9

Consider the capability certificate as shown in Figure 3; only one rule is evaluated against this request. However, in most cases having product operation, evaluation of more than one rule can take place, and then we have to consider the conjunction of the results of these rules.

Table 9. Mapping of Product Operation

Modified SQL Request and Login Info	EPAL Equivalent	Value
Information from Login	User-Category	Doctor
Relations	Data-Category	Medical-History Progress-Notes
Purpose in 'Where' Clause	Purpose	Treatment
Select	Action	Access
Attribute list in 'Where' clause except purpose Login Name	1.Container Attribute-ID Value 2.Container Attribute-ID Value	Patient-Info Patient-ID PID1234 User-Info User-Name Ajay

Consider Gary's request to data access two tables. He may also enter, *"Medical-History Natural Join on Progress-Report"*. This join operation will have the same mapping as in the product operation described above.

7. Future Work

Developing privacy policies/ capability certificates (henceforth referred simply as capability certificates) is an important step in the above-mentioned architecture. We observed many issues in developing the exact structure of capability certificates.

In this paper, we proposed the mapping of only simple SQL queries. However, further work needs to done to map complex queries.

7.1. Membership to multiple user-groups

In the proposed architecture, the capability certificates are stored in tables. A logical representation of the capability certificate can be represented as a tuple of <USER-GROUP RECORDS CAPABILIY>. Now consider a case where there are two such tuples as shown below.

U1 R1 R2 URa
U2 R3 R1 URb

Where Ui is user-group, Ri is record, URi is capability certificate.

Consider a user A who is a member of both the user-groups U_1 and U_2 and he wants to access record R_1. This could be the case where user A is a doctor and is also a patient. Hence there would be a conflict in terms of which of the two capability certificates to be used by Sentry while allowing on queries by user A. This can also be posed as a priority problem wherein Sentry would need to know which capability certificate has higher priority.

Also, such priorities may depend on the kind of record or the kind of user-group.

7.2. Capability transition

Consider the case where a user U1 authorizes a user U2 to access certain privacy-constrained information. It should also address the issue of whether user U2 can further pass capability to another user U3, and that too for period greater T2 that is greater than T1.

7.3. Privacy of capability certificates

The capability certificates themselves need to be protected based on the principles of privacy, in order to prevent malicious users from knowing which users have capability to access what data, thus making it simpler for malicious users to masquerade as some other user for accessing privacy constrained data that they could have otherwise not been able to access. Also, owners of data would not like to let others know about the privacy policy that they have specified.

Following the principle of safety, Capability certificates should be secure against any external access.

7.4. Mapping of Complex SQL Queries

Complex queries includes SQL query with sub-queries, set operations, grouping and aggregates. However, in such a mapping, privacy algebra defined in [5] will be applied.

8. Conclusion

The EPAL-based privacy middleware, based on the architecture proposed by Bhattacharya and Gupta [7] allows a health care organization to enforce the legal requirements of privacy related to health data. Thus one is able to enforce privacy in such an organization using a language that provides more robustness than plain P3P.

EPAL allows producing more complete rules than P3P. However, in order to fulfill the capability transfer requirement of the Capability Certificates, EPAL needs to be modified to include "Transfer-Group" parameter. Moreover, the obligations are made to operate on EPAL-Rule itself.

This paper further demonstrates the mapping of modified-SQL statements to EPAL query for the ease of users, so that they can query data in simple SQL. The proposed modification in SQL is to add a "Purpose" clause in order to make the SQL statement usable on privacy constrained data.

The paper shows a comprehensive mechanism for enforcing privacy within a Medical Information with minimal modifications to existing languages like EPAL and SQL.

9. Refrences

[1] Adam N. R. and Wortman J. C., Security-control methods for statistical databases. ACM Computing Surveys, 21(4):515-556, 1989.

[2] Agrawal R., Kiernan J., Srikant R. and Xiu Y, Hippocratic Databases (Vision Paper). IBM Almaden Research Center. 2002

[3] Agrawal R. and Srikant R.. Privacy preserving datamining. In Proceedings of the ACM SIGMOD, pages 439-450, 2000.

[4] Ashley P., Hada S., Karjoth G., Powers C. and Schunter M., Enterprise Privacy Authorization Language 1.2 (EPAL 1.2) W3C Member Submission, 2003

[5] Backes M., Pfitzmann B., and Schunter M., A toolkit for managing enterprise privacy policies. In Proceedings of European Symposium on Research in Computer Security (ESORICS), Lecture Notes in Computer Science 2808, Springer, 2003.

[6] Batra, V., Bhattacharya J., Chauhan H., Gupta A.; Mohania M. and Sharma U., Policy Driven Data Administration. In Proceedings of POLICY 2002, IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, 2002.

[7] Bhattacharya J. and Gupta S.K., Privacy Broker for Enforcing Privacy Policies in Databases. In. Proceedings of Fifth international conference on knowledge based computer systems. Hyderabad, India, 2004

[8] Castano S., Fugini M., Martella G., and Samarati P., Database Security. Addison Wesley, 1995

[9] Clifton C. and Marks D., Security and privacy implications of data mining. In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pages 15-19, 1996.

[10] Dinardo C.T., Computers and Security. AFIPS Press, 1978.

[11] Edwards K.B., Oracle 9i Privacy Protections, Oracle CorporationThe Health Insurance Portability and Accountability Act of 1996 (HIPPA), Available from http://www.leagalarchiver.org/hippa.htm

[12] Kaplam Marc A., IBM Cryptolopes TM, Superdistribution and Digital Rights Management. http://www.research.ibm.com/people/k/kaplan/cryptolope-docs/crypap.html, 1996

[13] Krekke T.H., Privacy Violation Detection , master thesis, Norwegian Computer Centre, Norwegian University of Science and Technology, Spring 2004.

[14] Oppliger R., Internet security: Firewalls and beyond. Comm. ACM, 40(5):92-102, 1997

[15] Oracle Corporation. Database Encryption in Oracle 8i, August 2000.

[16] Paola B.i, 'IYuste: an online privacy seal program. Comm. ACM, 42(2):56-59, 1999.

[17] Shoshani A., Statistical databases: Characteristics, problems and some solutions. In Proceedings of the Eighth International Conference on Very Large Databases (VLDB), pages 208-213, Mexico City, Mexico, 1982

[18] Smith S.W., Safford D. Practical Server Privacy with Secure Coprocessors. In IBM Systems Journal, Vol 40, No.3, 2001

[19] Smith S.W., WebALPS: Using Trusted Co-Servers to Enhance Privacy and Security of Web Interactions, Research Report RC-21851, IBM Thomas J. Watson Resrach Center, Yorktown Heights, NY 10598 2000.

[20] The World Wide Web Consortium. The Platform for Privacy Preference (P3P}. Available from http ://www. w3.org/P3P/P3FAQ. html.

[21] Yee B.S., Using Secure Coprocessors, Ph.D. thesis, Computer Science Technical Report CMU-CS-94-149, Carnegie Mellon University, Pittsburgh, PA, 1994