# Preserving Privacy in e-Health Systems Using Hippocratic Databases

Yuan Hong, Shuo Lu, Qian Liu
*Department of Computer Science*
*Concordia University*
*1455 de Maisonneuve Blvd. W,*
*Montreal, QC, Canada*
*{y_hon, lu_shuo, liu_qian}@cs.concordia.ca*

Lingyu Wang, Rachida Dssouli
*Concordia Institute for*
*Information Systems Engineering*
*1455 de Maisonneuve Blvd. W,*
*Montreal, QC, Canada*
*{wang, dssouli}@ciise.concordia.ca*

## ABSTRACT

Safeguarding patients' private information is one of the most challenging issues in the design and implementation of modern e-Health systems. Recent advances in Hippocratic Databases (HDB) show a promising direction towards the enforcement of privacy policies in e-Health systems. This paper tackles issues in applying the HDB design to e-Health systems. More specifically, we design an architecture for integrating APPEL preferences with HDB; we extend the original HDB design to support fine-grained privacy authorizations demanded by patients; we adapt the design to a multi-dimensional model; we also propose a design for hierarchical authorizations. Finally, we discuss implementation issues and justify our designs with experimental results.

## 1. INTRODUCTION

Electronic healthcare systems are playing a critical role in today's medical organizations. How to safeguard patients' private information is an important and challenging issue faced by the designer and administrator of e-Health systems. The privacy issue is critical to such systems because most medical data are about individual patients and highly sensitive [1]. Inappropriate disclosures of those data cause privacy breaches to patients, which in turn lead to serious legal and financial consequences to the organization. At the same time, the privacy issue is particularly challenging in e-Health systems due to the usually complex design and implementation of such systems. There exist standards and solutions for addressing the privacy issue in general-purpose applications. The platform for privacy preferences (P3P) developed by the World Wide Web Consortium (W3C) allows users and websites to declare privacy preferences and policies in a machine-readable format [2]. On the other hand, the Hippocratic Databases is a framework for enforcing privacy policies based on database technologies [3]. The integration of P3P and HDB is a natural solution for preserving privacy in e-Health systems, which forms the basis of our work.

In this paper, we tackle several issues around the integration of P3P and HDB technologies in e-Health systems. First, we design an architecture for integrating P3P preferences with HDB policies. Patients' preferences specified in APPEL (P3P's language for privacy preferences) are mapped to privacy metadata tables stored in HDB. Doctors requesting for private data are authorized against the privacy metadata. Second, we study how to support fine-grained privacy authorizations in HDB. The attribute-level access control turns out to be insufficient for e-Health systems. We provide a solution based on a redesigned schema. Third, we extend HDB to support the multi-dimensional model. The original design of HDB based on relational database model is not suitable for multi-dimensional models used in medical data warehouses. Fourth, we provide a snowflake schema-based design for simplifying the representation of hierarchical authorizations. Finally, we discuss implementation and show experimental results.

The rest of this paper is organized as follows. First, Section 2 reviews related work. Section 3 describes an architecture for preserving privacy based on integration of P3P and HDB. Section 4 proposes a fine-grained authorization mechanism for HDB and extends it to support the multi-dimensional model and to support hierarchical authorizations. Section 5 discusses implementation issues of the proposed solutions. Finally, Section 6 concludes the paper.

## 2. RELATED WORK

A survey of research topics and trends on e-Health systems can be found in [4]. Protecting patients' privacy is a mandatory requirement in most e-Health systems according to privacy legislation and regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) [5]. Threats to patients' privacy may arise from many aspects of a medical organization. For example, published medical data can lead to attacks on patients' privacy even though the data are sanitized. The concept of *k*-anonymity requires identifying attributes to be generalized such that any real-world individual can be linked to at least *k* records in the published data, which is considered a tolerable privacy threat [6]. Privacy threats within a medical organization may come from unauthorized accesses to sensitive data. A basic requirement found in most privacy regulations is that accesses to patient records should only be granted to users with appropriate privileges for intended purposes during a

0730-3157/08 $25.00 © 2008 IEEE
DOI 10.1109/COMPSAC.2008.62

given time period [5]. Such a requirement for fine-grained access control (FGAC) can be handled by the application layer or by database systems through view-based security. For example, Oracle's implementation of FGAC, known as Virtual Private Database (VPD), allows policies to be attached to tables and triggered by accesses [7]. Other popular commercial products like Sybase, Microsoft SQL Server, and IBM DB2, all have different degree of support for FGAC. The Platform for Privacy Preferences (P3P) is a standard for encoding a user's privacy preferences and an organization's privacy policies in a machine-readable format, such that a user's browser can interact with the organization's website to determine whether the former's privacy preferences matches the latter's privacy policies [2]. P3P provides a standard language for specifying privacy preferences about disclosing private data, namely, A P3P Preference Exchange Language (APPEL) [8]. Hippocratic Databases (HDB) is designed for preserving privacy in database applications [3].

## 3. ARCHITECTURE

The architecture of the privacy-protection subsystem of an e-Health system is illustrated in Figure 1. We consider two types of users accessing the e-Health system. First, *patients* state their opt-in and opt-out preferences through a web interface. Second, *doctors* need to access the personal information of their patients for treatment purposes. Notice here the patient and doctor only refer to their roles in either providing or requesting the private data. Other users of the e-Health system, such as a nurse, may be considered as a patient, a doctor, or both depending on their roles with respect to the private data.

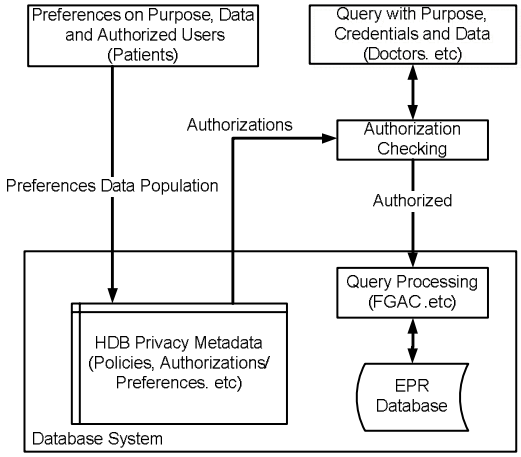


**Figure 1. Architecture for preserving privacy**

Patients specify their preferences about disclosing private data in the APPEL language through a web interface. The preferences are checked against the mandatory part of P3P policies for conformance between the client's browser and the web server (this can be implemented in either a client-centric or server-centric way [9]). If the preferences match the policies, the preferences are mapped to and stored in the attributes and records of privacy metadata tables in backend HDB for later references. When doctors request for accesses to private data, the application will provide authentication credentials and associated purposes together with queries. Based on the purposes and the requested resources, such as records and attributes, the system checks corresponding metadata and determines whether the doctor is a legitimate recipient of the requested private data.

## 4. FINE-GRAINED AUTHORIZATION

Central to the design of HDB is the attribute access control, that is, only those attributes with matching purposes will be visible to a query [3]. However, making attribute the most granular unit for access control turns out to be insufficient for e-Health systems. In Section 4.1, we provide a solution based on redesigned schemata. In Section 4.2, we extend the original HDB design to the multi-dimensional model. Finally, in Section 4.3 we design a snowflake schema-based solution for hierarchical authorizations.

### 4.1 Fine-Grained Authorization in HDB

Table 1 shows the schemata of the original HDB design [3]. Table *EPR* and *Patient* store private data about patients. Table *Privacy-Authorization* records which users are authorized to access each combination of table, attribute and purpose (we shall only include a few attributes for simplicity). In this example, attribute *Diagnosis* of table *EPR* may be accessible to doctors for treatment purpose and to analysts for statistics purpose, whereas attribute *Age* of table *Patient* is only for treatment purpose.

**Table 1. Schemata of the original HDB design**

| Table | Attributes |
|---|---|
| *EPR* | EPR-ID, Patient, Diagnosis, Purpose |
| *Patient* | Patient, Gender, Age, Address, City, Purpose |

| Table | Attributes |
|---|---|
| *Privacy-Authorization* | Purpose, Table, Attribute, Authorized-Users (AU) |

Table *Privacy-Authorization*

| ***Purpose*** | ***Table*** | ***Attribute*** | ***AU*** |
|---|---|---|---|
| Treatment | EPR | Diagnosis | Doctor |
| Treatment | Patient | Age | Analyst |
| Statistics | EPR | Diagnosis | Analyst |

However, the above design has a limitation in that it does not support fine-grained authorizations. In a typical e-commerce website, a transaction will not even begin unless the user's preferences match all of the website's policies.

Each attribute can thus have a fixed set of authorized users no matter in which record it appears. This all-or-none approach is not suitable for medical organizations where, for example, a patient should not be refused of treatment simply because he/she disagrees to provide age or gender for statistics purposes. Any patient should be allowed to opt out optional privacy policies or opt in with conditions. For example, Bob may decide that his diagnosis information should be disclosed for statistics purposes only if the symptom is flu, and the information should be kept private for all other symptoms. As another example, he may choose to allow or disallow a specific doctor in accessing his diagnosis information for certain symptoms. Another patient Eve may have a completely different preference about these private data. None of those requirements can be represented in table *Privacy-Authorization* in Table 1.

Table 2 shows our redesigned schemata for supporting fine-grained access control. We add an additional attribute Authorization-ID to the tables and a foreign key constraint on the attributes from table EPR to table Patient. In table EPR, for symptom Flu, Bob is willing to disclose his diagnosis information for both statistics and treatment purposes and his name for treatment purpose but only to Alice. For the symptom of Diabetes, Bob does not want to disclose any information for any purpose. Also for symptom Flu, Eve has a preference different from Bob's which is to disclose the diagnosis data to doctors.

**Table 2. Schemata for fine-grained authorizations**

| Table | Attributes |
|---|---|
| *EPR* | EPR-ID, Patient, Diagnosis, Purpose, Authorization-ID (A-ID) |
| *Patient* | Patient, Gender, Age, Address, City, Purpose, A-ID |

| Table | Attributes |
|---|---|
| *Privacy-Authorization* | A-ID, Purpose, Table, Attribute, Authorized-Users (AU) |

Table *EPR*

| ***EPR-ID*** | ***Patient*** | ***Diagnosis*** | ***Purpose*** | ***A-ID*** |
|---|---|---|---|---|
| 000001 | Bob | Flu | Treatment | 1, 2 |
| 000001 | Bob | Flu | Statistics | 3 |
| 000002 | Bob | Diabetes | | |
| 000003 | Eve | Flu | Treatment | 1 |

Table *Privacy-Authorization*

| ***A-ID*** | ***Purpose*** | ***Table*** | ***Attribute*** | ***AU*** |
|---|---|---|---|---|
| 1 | Treatment | EPR | Diagnosis | Doctor |
| 2 | Treatment | EPR | Patient | Analyst |
| 3 | Statistics | EPR | Diagnosis | Analyst |

The above design of table *Privacy-Authorization* assumes a *closed policy*. That is, unless a user explicitly appears in attribute *Authorized-Users* in table *Privacy-Authorization*, the user will by default be prohibited from accessing the attribute. Such a closed policy is inconvenient when a patient chooses to prohibit certain users while allowing all others to access his/her private data. For example, Bob does not want Alice to see his name but does not care about other doctors accessing that data. It would be prohibitive to explicitly list all other doctors' names in attribute *Authorized-Users*.

We address this issue by making attribute *Authorized-Users* a *signed* attribute. That is, each user appearing in the attribute is either to be allowed or disallowed, as denoted by a preceding sign + or –. Table 3 shows an example where the second authorization prohibits a specific user Alice from accessing the patient's name while allowing all other doctors to access the same data. An additional issue here is the potential conflicts between signed values. For example, if Alice is a doctor, then the first signed Value *+Doctor* will grant Alice the access (since she is a doctor) but the second value *–Alice* will prohibit this access. The solution is to define meta-policies for resolving conflicts, such as the *most-specific-take-precedence* meta-policy by which Alice will be prohibited (since Alice is more specific than Doctor).

**Table 3. Signed attribute *Authorized-Users***

Table *Privacy-Authorization*

| ***A-ID*** | ***Purpose*** | ***Table*** | ***Attribute*** | ***AU*** |
|---|---|---|---|---|
| 1 | Treatment | EPR | Diagnosis | +Doctor |
| 2 | Treatment | EPR | Patient | {+Doctor, –Alice} |

## 4.2 HDB for Multi-dimensional EPR

The original design of HDB is based on the relational database model, which is suitable for small to medium size operational applications. However, analytical applications dealing with a large amount of data, such as a medical data warehouse, will typically adopt a multi-dimensional model instead of the relational model [10]. We now extend the schema design introduced in Section 4.1 to the multi-dimensional model. We also make optimizations to reduce the storage requirements of privacy metadata.

A multi-dimensional model typically adopts a star schema that consists of a number of dimensional tables and a fact table, with foreign key constraints linking the two. Table 4 shows a star schema augmented with attributes *Purpose* and *Authorization-ID*. Table *EPR* is the fact table. It can be observed that the fact table includes much redundancy due to various purposes associated with the same record. Such redundancy may be acceptable in an operational database,

but it is usually prohibitive for analytical applications based on a multi-dimensional model where the cardinality of the fact table can easily go beyond millions. This situation will be exasperated when table *EPR* includes attributes of object data types, such as laboratory test measurements in multimedia format (for example, X-ray, CT scan or MRI).

**Table 4. HDB in star schema**

| Table | Attributes |
|---|---|
| *EPR (fact table)* | Patient-ID, Doctor-ID, Date-ID, Diagnosis, Prescription, Purpose, Authorization-ID (A-ID) |
| *Patient* | Patient-ID, Name, Gender, Address, City, Region, Province, Country, Authorization-ID (A-ID) |
| …… | …… |

Fact table *EPR*

| ***Patient-ID*** | **…** | ***Diagnosis*** | **…** | ***Purpose*** | ***A-ID*** |
|---|---|---|---|---|---|
| 001 | … | Flu | … | Treatment | 1, 2 |
| 001 | … | Flu | … | Statistics | 3 |
| 001 | … | Flu | … | Prescription | 4 |
| 001 | … | Flu | … | Research | 5 |
| 001 | … | Diabetes | … | Treatment | 1, 2 |
| … | … | … | … | … | … |

To avoid the extra storage overhead introduced by multiple purposes, we decompose the fact table through schema normalization. As shown in Table 5, we add attribute *EPR-ID* as a primary key to table *EPR* and at the same time remove attributes *Purpose* and *Authorization-ID* from the table. We then introduce a new table *Purpose-Authorization* to record the purpose and authorizations for each EPR record. We do not normalize the dimensional tables since they typically have a much smaller cardinality than that of the fact table and the redundancy introduced by multiple purposes is thus usually acceptable.

**Table 5. Schema for multi-dimensional HDB**

| Table | Attributes |
|---|---|
| *EPR (fact table)* | EPR-ID, Patient-ID, Doctor-ID, Date-ID, Diagnosis, Prescription |
| *Patient* | Patient-ID, Name, Gender, Address… Purpose, Authorization-ID (A-ID) |
| ..... | …… |
| *Purpose-Authorization* | EPR-ID, Purpose, Authorization-ID (A-ID) |

Another issue relevant to a multi-dimensional HDB is that there are two different levels of authorizations for an attribute in the dimensional tables. For example, Bob may want his address to be disclosed to doctors for treatment purpose, which can be represented by an authorization for table *Patient* and attribute *Address*. However, Bob may want his address disclosed for symptom *Flu* but not for *AIDS*. This second authorization cannot be represented as above because the *Authorization-ID* (*A-ID*) in table *Patient* is not associated with attribute *Diagnosis* in the fact table. Our solution is to allow an authorization in the fact table to specify an attribute in the dimensional table. As shown in Table 6, the authorization has its *Authorization-ID* (*A-ID*) appearing in the fact table (that is, table *Purpose-Authorization*) but specifies an attribute *Address* in a dimensional table *Patient*.

**Table 6. Access control for dimensional tables**

Fact table *EPR*

| ***EPR-ID*** | ***Patient-ID*** | **…** | ***Diagnosis*** | **…** |
|---|---|---|---|---|
| 000001 | 001 | … | Flu | … |
| 000002 | 001 | … | AIDS | … |

Table *Patient*

| ***Patient-ID*** | **Name** | ***Gender*** | ***Address*** | **…** |
|---|---|---|---|---|
| 001 | Bob | … | … | … |

Table *Purpose-Authorization*

| ***EPR-ID*** | ***Purpose*** | ***Authorization-ID (A-ID)*** |
|---|---|---|
| 000001 | Treatment | 1 |

Table *Privacy-Authorization*

| ***A-ID*** | ***Purpose*** | ***Table*** | ***Attribute*** | ***AU*** |
|---|---|---|---|---|
| 1 | Treatment | Patient | Address | Doctor |

## 4.3 Hierarchical Authorization

Hierarchies naturally exist in many aspects of an HDB, such as EPR data, authorized users, and purpose. We have proposed a hierarchical approach for simplifying the specification of privacy preferences and authorizations by leveraging such hierarchies [12]. However, how to represent privacy authorizations in an HDB in the presence of such hierarchies is not addressed in [12] and we now discuss a snowflake schema-based design for this purpose.

Privacy authorizations can be derived from hierarchies on EPR data, authorized users, and purposes. For example, if *Address* has been authorized then less sensitive data of *City* or *Country* are implicitly authorized altogether (exceptions to such inheritance are handled in [12] and we shall not address it in this paper). In the authorized user dimension, if *Nurse* is authorized to access certain EPR data, then users with higher privileges, such as *Nurse Supervisor* will be authorized to access the same data as well. For the purpose dimension, if a purpose *Treatment* has been authorized then a more specific purpose (sub-purpose) such as *Surgery* will also be authorized. It is worth noting that the inheritance of

authorizations is reversed for the purpose dimension in contrast to the other two. For example, *Country* dominates *City* (in the sense that a country may correspond to many cities) and authorizations on *City* imply those on *Country*, whereas *purpose* dominates *sub-purpose* but authorizations on *purpose* imply those on *sub-purpose.*

Neither the original HDB design nor our designs in previous sections can provide efficient implementation of such hierarchical authorizations. We instead provide a solution based on the snowflake schema. Table 7 shows an example of our design. For simplicity, we only include the address hierarchy in EPR data. In this schema, all tables are fully normalized, and the hierarchical relationships between attributes are represented as foreign key constraints. The metadata tables include both the authorized user dimension and the purpose dimension.

**Table 7. Snowflake schema for EPR data and metadata**

| EPR Table | Attributes |
|---|---|
| Address | AddressID, Address, CityID |
| City | CityID, City, ProvinceID |
| Province | ProvinceID, Province, CountryID |
| Country | CountryID, Country |

| Metadata Table | Attributes |
|---|---|
| Nurse | NurseID, NurseName, SupervisorID |
| Supervisor | SupervisorID, SupervisorName |
| … | … |
| Sub-Purpose | Sub-PurposeID, Sub-Purpose, PurposeID |
| Purpose | PurposeID, Purpose |

In the presence of hierarchies, authorizations can be given as any combination of three sets of attributes where each set is from the EPR data, authorized user, and purpose dimension, respectively. Under the original design of HDB, this may result in a large number of authorizations. For example, if an authorization is given as the following: {*City, Province, Country*}, {*Nurse, Nurse Supervisor*}, {*Diagnosis*}, then totally six authorizations will be derived, as shown in Table 8.

**Table 8. Authorizations under original HDB design**

| *A-ID* | *Purpose* | *Table* | *Attribute* | *AU* |
|---|---|---|---|---|
| 1 | Diagnosis | Patient | City | Nurse |
| 2 | Diagnosis | Patient | Province | Nurse |
| 3 | Diagnosis | Patient | Country | Nurse |
| 4 | Diagnosis | Patient | City | Nurse Supervisor |
| 5 | Diagnosis | Patient | Province | Nurse Supervisor |
| 6 | Diagnosis | Patient | Country | Nurse Supervisor |

With our approach, the above collection of authorizations can be conveniently represented using a single record. More specifically, we redesign the *Privacy-Authorization* table as shown in Table 9. In addition to the *Authorization-ID* attribute (required for fine-grained authorization as described in Section 4.1), each authorization now includes three pairs of attributes. Each pair indicates which attribute of which table is authorized in each of the three dimensions: EPR data, authorized users, and purpose.

**Table 9. Redesigned schema for hierarchical authorizations**

| Table | Attributes (Columns) |
|---|---|
| *Privacy Authorization (fact table)* | Authorization-ID (A-ID), EPR Table, Data-ID(D-ID), Purpose Table, Purpose-ID (P-ID), Authorized-Users Table (AU Table), Authorized User-ID (AU-ID) |

For example, Table 10 shows an example (where the D-ID is linked to AddressID, CityID, etc.). Due to the aforementioned hierarchies, the single record in Table 10 is sufficient for representing the six authorizations given in Table 8.

**Table 10. Authorization under the redesigned schema**

| *A-ID* | *EPR Table* | *D-ID* | *Purpose Table* | *P-ID* | *AU Table* | *AU -ID* |
|---|---|---|---|---|---|---|
| 1 | City | 1 | Sub-Purpose | 2 | Nurse | 1 |

# 5. IMPLEMENTATION

To allow users to specify privacy authorizations, we implement an APPEL preference selector through Web interfaces based on BEA WebLogic 8.1.2. We generate authorizations from the selected options in the Web interfaces using a Java application. The application then stores the generated authorizations in a backend Oracle 9i database. Oracle Enterprise Edition supports fine-grained access control (FGAC) through the VPD feature, which is the aggregation of server-enforced FGAC with a secure application context. Our fine-grained authorizations are

implemented using this feature by defining policy functions. We implement three schemata as described in Section 4 for fine-grained authorizations, for multi-dimensional model without hierarchies, and for hierarchical authorizations, respectively. In practice, however, an application may choose to implement only one of those schemata based on its specific needs.

To evaluate our design, we populate our database with the adult dataset from the UCI Machine Learning Repository [11] (which has been used in many related work on data privacy). There are 32561 patients in the dataset. We made following assumptions. Each patient owns three records in our *EPR* table, and each record is associated with four purposes: *Treatment*, *Research*, *Statistics,* and *Prescription*. Table 11 shows that our redesigned fine-grained authorization schema requires significantly less total storage than that of the original HDB design if the fine-grained authorization is to be enforced. Clearly, with more purposes or attributes that hold more data, the difference would be more significant.

**Table 11. Storage requirements**

| # | Table | Attributes Number | Tuples Number | Storage (Bytes) |
|---|---|---|---|---|
| 1 | *Patient* in Table 4 or 5 | 15 | 32,561 | 5,242,880 |
| 2 | *EPR* in Table 4 | 11 | 390,732 | 75,497,472 |
| 3 | *EPR* in Table 5 | 9 | 97,683 | 16,777,216 |
| 4 | *Purpose-Authorization* in Table 5 | 3 | 390,732 | 24,117,248 |
| | Summation of #1 and #2 | | 80,740,352 Bytes | |
| | Summation of #1, #3, and #4 | | 46,137,344 Bytes | |

## 6. CONCLUSION

This paper addressed the issue of preserving patients' privacy in e-Health systems. We proposed to allow patients to specify their privacy preferences in APPEL, which will be recorded as authorization policies in the backend HDB. Doctors are then authorized against such policies when their applications request for privacy data. We identified the lack of fine-grained authorizations as a limitation of the original HDB design and provided a solution based on a modified schema. We extended the HDB design to support the multi-dimensional model so it can be used to preserving privacy in analytical data applications. We also showed how hierarchies can be leveraged to simplify the representation of authorizations. Experimental results justified our design. It is our belief that the proposed solution can be integrated into existing e-Health system to provide patients with better privacy protection.

## 7. REFERENCES


[1] R. Agrawal and C. Johnson, "Securing Electronic Health Records without Impeding the Flow of Information," *International Journal of Medical Informatics*, 2007.

[2] World Wide Web Consortium *(W3C)*, "*P3P Implementations from W3C*,"

http://www.w3.org /P3P/implementations.

[3] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, "Hippocratic Databases," *Proc. of the 28th Int'l Conference on Very Large Databases (VLDB)*, 2002.

[4] G. Kaur and N. Gupta, "E-health: A New Perspective on Global Health," *Journal of Evolution and Technology*, Vol. 15, Issue 1, 2006.

[5] *"Health Insurance Portability and Accountability Act,"* United States Public Law 104-191, 1996.

[6] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," *Proc. of the 7th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1998.

[7] T. Kyte, "*Fine-grained access control*," Technical report, Oracle Corporation, 1999.

[8] L. Cranor, M. Langheinrich and M. Marchiori, "A P3P Preference Exchange Language 1.0 (APPEL 1.0)," *W3C Working Draft*, April 2002.

[9] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, "Server-Centric P3P," *Proc. of the W3C Workshop on the Future of P3P*, November 2002.

[10] N. Pendse, "*The OLAP report - what is olap*," OLAP Report Technical Report, 2001.

http://www. olapreport.com /fasmi.htm.

[11] "*UCI Machine Learning Repository*,"

http://archive.ics.uci.edu/beta/datasets/Adult.

[12] Y. Hong, S. Lu, Q. Liu, L. Wang and R. Dssouli, "A Hierarchical Approach to the Specification of Privacy Preferences," *Proc. of the 4th IEEE Int'l Conference on Innovations in Information Technology,* November 2007.