A Privacy Preserving Model Bridging Data Provider and Collector Preferences

Kambiz Ghazinour  
University of Ottawa  
School of Electrical Engineering and Computer Science  
Ottawa, ON, K1N 6N5, Canada  
kghazino@uottawa.ca

Ken Barker  
University of Calgary  
Department of Computer Science  
Calgary, AB, T2N 1N4, Canada  
kbarker@ucalgary.ca

ABSTRACT

With the increasing amount of data collected by service providers, privacy concerns increase for data owners who provide private data to receive services. Legislative acts require service providers to protect the privacy of customers. Privacy policy frameworks, such as P3P, assist them by describing their privacy policies to customers (e.g., publishing privacy policy on websites). Unfortunately, providing the policies alone does not guarantee that they are actually enforced because privacy is not a key feature of conventional access control models. A privacy-preserving model should consider the privacy preferences of both the data provider and collector. This paper briefly develops a Lattice-based Privacy Aware Access Control (LPAAC) Model that enforces privacy policies, facilitates customization of privacy agreements, and accommodates preferences of both data and service providers. We demonstrate our model’s design and feasibility with corresponding privacy catalogues. Examples clarify the usability, and we provide the implementation of our privacy catalogues that show the efficiency and scalability of our model.

Categories and Subject Descriptors

E.1 [DATA STRUCTURES]: Graphs  
K.4.1 [Computers and Society]: Public Policy Issues – Privacy

General Terms


Keywords

Privacy, Policy, Access Control, Lattice, Data Provider/Collector.

1 INTRODUCTION

Privacy is a leading concern for individuals who utilize computing resources ranging from shopping online to visiting a health care provider. As service providers collect more data from customers, concerns about how their data is treated after being collected increases. Legislative acts, such as the Health Insurance Portability and Accountability Act (HIPAA), and the 1974 US Privacy Act, explicitly require service providers address privacy concerns by informing customers how their data is treated after its collection. Much research has proposed privacy policy frameworks and policy languages and models such as P3P [3] and PPVM [7] to assist data owners and collectors to communicating their privacy concerns.

Although these privacy-enhanced technologies inform the data provider of the privacy policies practiced in the organization, description alone is insufficient, and two issues that must be addressed. First, these frameworks do not provide enforcement mechanisms. Thus, data providers receive no guarantee that the policies and promises are actually enforced by the access authorization and data processing phases. In fact, data providers have no choice but to rely on data collectors and hope their data is being treated as promised. Second, since both the data provider and collector want to maximize their control over the collected data, there must be a mechanism to handle each party’s privacy preferences. We assume data providers should specify their privacy preferences and the collector wants to honor their requests so we seek to develop systems capable of enforcing those preferences in privacy aware access control systems.

With the increasing chance of privacy violation in the collection of personal information [12], legislation explicitly require service providers to address privacy concerns by informing customers how their data is being treated after being collected [3].

Much research has proposed privacy policy frameworks and policy languages, such as P3P [3] and visualization models such as PPVM [7], to assist data providers and collectors in communicating their privacy preferences and practices. These frameworks do not provide any mechanism to enforce privacy policies. Thus, data providers receive no guarantee that the policies and promises are actually implemented by the access authorization and data processing systems. In fact, data providers have no choice but to rely on collectors and hope that data is treated as promised.

Hippocratic databases [1] argue for database management systems in which privacy is preserved and the database itself does not reveal any information that violates the data provider privacy. Since the Hippocratic databases proposal, there has been a much research to develop privacy preserving models [2, 4, 9].

1.1 Motivation and contribution

We have already introduced preliminary work to represent data privacy elements in a lattice [6] and applied them to capture P3P privacy framework [5]. In this work, we pursue our vision by presenting a model and algorithm that demonstrates how it works and its feasibility by providing an implementation.

We believe the Lattice-based Privacy Aware Access Control (LPAAC) model allows privacy to become central to the access control mechanism rather than granting access to the data users only because of their role without adequately considering their purpose and other related privacy predicates.

2 PRIVACY AND ACCESS CONTROL

With the introduction of privacy preserving databases, there will be new demands on the data definition and query languages, query processing, indexing and storage structures, and access control mechanisms. In this paper, we are focusing on the access control piece of this issue. Conventional access control mechanisms such as Discretionary Access Control, Mandatory Access Control, Role-Based Access Control (RBAC), etc. [4] are not designed to preserve data privacy. Many RBAC extensions support purpose-
based, as well as other constraints introduced in literature [2, 10]. Considering the privacy requirements identified in data privacy, purpose, visibility, granularity and retention (see [6] for definition), and the characteristics of privacy preserving database systems as discussed in Hippocratic databases [1], we are led to develop a new generation of privacy-centric access control models. This type of access control model enables database management systems to protect the privacy of the data provider. As a result, upon receiving a data request [8], the database system does not reveal information that violates the privacy of the data provider according to the privacy preferences defined by the data provider when the data was collected.

**Definition 1. (Privacy Policy)** Let \( P \) be the set of all purposes defined in the system, \( V \) be a set of possible visibilities, \( R \) be a set of retentions, \( G \) be a set of possible granularity levels, \( A \) be a set of actions defined in the relational database, \( O \) be a set of data objects and \( C \) be a set of possible constraints, and \( H\text{Plc} \) be a set of privacy policies defined by the data collector. Each \( hp \in H\text{Plc} \) is a privacy policy defined with the tuple: \( hp = (p, v, a, g, o, r, c) \), where \( p \in P \), \( g \in G \), \( r \in R \), \( v \in V \), \( a \in A \), \( o \in O \), and \( c \in C \).

**Example 1:** Consider the following privacy policy in a phone service provider company:

\[
hp = \{\text{marketing, marketing department, read, 0, age, 3 years, inform customer (email)}\}
\]

which means that the customer’s exact age (with 0 level of granularity) can be used for marketing purposes and is visible to the marketing department for a three year period; and each time the data is being used, the customer must be notified via email.

Considering this privacy policy, the permission is assigned to the proper user (marketing department) for the intended purpose (marketing). This illustrates how privacy becomes the center of an access control mechanism rather than giving access to the marketing department only because the staff who work there have the role of “marketing staff”. In other words, if a staff member of the marketing department does not have the proper purpose (marketing) then he/she cannot have access to the age attribute.

Such policy is defined by the phone service provider. In the next section, we discuss how to capture the provider’s privacy preferences. It is fruitful to see how node subsumption in a lattice structure helps manage access requests by data users. Assume Alice works in the marketing department and has the purpose of competition to access the age of a customer. When she submits her query the system identifies that she works in the marketing department and is permitted to have the purpose in the first place (more details will be discussed in Section 3.1). Further, the systems evaluate the retention time and other constraints. Although Alice has competition purpose to access data, since according to the structure of the lattice defined by the service provider, it is a more specific purpose to access data than what is mentioned in the policy agreed by the provider (marketing), the system grants the permission to her. Notice that Alice’s request would be permitted even if she was a member of the business department since according to the visibility lattice, which reflects the organization’s policy and agreed by the data provider, whatever is visible to the marketing department is visible to the business department as well.

### 2.1 Data providers’ preferences

One of the motivations of this work is to bridge the privacy desires of both the data collector and provider in the model. Hence, there should be a mechanism to consider the privacy preferences of data provider as well. In fact, legislative acts and privacy guidelines clearly identify the privacy rights of the data providers and ask the data collectors to inform the data providers by publishing these guidelines on their websites [12]. On the other hand, service providers and organizations that need to collect data in order to provide service to customers are bound by the same legislative acts and privacy guidelines which limit their data collection and usage to the extent that it does not violate the privacy of the data providers. Hence, data providers should define their own preferences as well. Ideally, their preferences should not limit day to day business of the data collector and it should be compliant with lawful policies. Hence, the data collector, according to its policy, should specify a degree of flexibility where possible and let data providers specify their privacy preferences within that range.

To illustrate all the possible ranges/preferences of the privacy predicates defined by the collector for the data provider, we adopt the concept of Minimal Acceptance Limit (MinAL) and Maximal Acceptance Limit (MaxAL) [11]. MinAL, defined on the purpose lattice [13], specifies a minimum (most general) purpose for which the data provider’s data will be used. Intuitively, MaxAL defines a limit to what purpose the data provider might prefer the data be used. Obviously, if the data collector does not define any maximum limit then the data provider might select a too limiting purpose for the provided data which makes the day to day business of the data collector impossible. According to the privacy policies defined by the data collector, MinAL and MaxAL can be defined on the same node or encompass a range of nodes. If the MinAL and MaxAL are defined on the same node then choosing that predicate is mandatory for the data provider (e.g. if the MinAL and MaxAL for the purpose lattice in the example shown in Figure 1 are set to competition, that means the data provider is obliged to accept this purpose for the specified privacy policy.)

![Figure 1. MinAL and MaxAL in a purpose lattice](image)

MinAL and MaxAL in a lattice should be defined in a way that for each node located in this range, MinAL is the ancestor for all of them and MaxAL is the descendant node for all. For instance, in the purpose lattice shown in Figure 1, where Competition is the MaxAL, a node such as Invoice cannot be MinAL since there are nodes like Marketing where their descendant is the MaxAL but their ancestor is not MinAL since there is no path from the Marketing node to the Invoice node in this lattice.

In our model, as shown with dash-rounded rectangles in Figure 1, MinAL and MaxAL are defined by the data collector for the data provider with respect to their policies for each of the purpose, visibility, granularity and retention lattices.

**Example 2.** Consider Example 1 in which only the privacy preferences of the collector were captured. In this example, we assume the data collector defines a MinAL and MaxAL for the data providers so they can select their preference within that range. Hence, the modified privacy policy of the collector may appear as follows: \( hp = \{\text{marketing, competition, marketing department, read, g(0)-g(2), age, three years, inform customer by email}\} \), which means that the customer’s age (either exact value, shown with zero level of granularity g(0), or with two levels of granularity g(2)) can be used for the marketing or competition purposes and visible to the marketing department for a three year period and each time that the data is being used, the provider (customer) must be notified via email. In this example, the data collector has not
compromised on visibility or retention by setting both MinAL and MaxAL to the same node.

In the next step, the data provider reviews the privacy policy and selects preferences within the specified range. It is crucial to remember that since the data provider selects competition as the proper purpose, if a data user has the marketing purpose they cannot access the data item.

2.2 Model design

To represent the design for the model, we develop the following privacy catalogues. These catalogues, namely SysHousePolicies, SysOwnerPolicies, SysPurpose, SysVisibility, SysGranularity and SysRetention represent the collector privacy policies, the data owner/provider privacy policies, and the allowed range of each privacy predicate specified by the collector. In addition, two catalogues SysUserPur and SysUserVis represent purposes data users are allowed to have and each visibility level that a data user belongs to, respectively. These eight catalogues would ideally be implemented as part of the system catalogues inside the database management system. However, for simplicity, and to demonstrate the approach’s utility transparently, they are implemented as a set of stand-alone (or user defined) tables. This allows legacy systems to utilize the privacy preserving model as a wrapper without the need for a major upgrade to the database engine itself. Our approach should eliminate the redundancies necessary to implement a Hippocratic database or the requirement placed on the database designer to build this into each new implementation.

Figure 2 shows the corresponding tables for each privacy catalogue. In SysHousePolicies, the collector specifies the list of privacy policies defined for its day-to-day business. Each record represents a privacy policy equivalent to the tuple hp ≝ < p, v, a, g, o, r, c >. Action a, is defined as one of the data manipulation operations in the relational database such as Select, Insert, etc. Object o is stored as a pair of table and column. Predicates p, v, a, g and r are stored as a set of one or more nodes in the corresponding purpose, visibility, granularity and retention lattices that are justified for that specific policy. The related MinAL and MaxAL of each lattice are stored in SysPurpose, SysVisibility, SysGranularity and SysRetention, respectively. The last predicate, constraint c, is stored in two fields: collector constraint (H_C) and owner constraint (O_C). The collector constraint is a set of constraints defined by the collector in the policy (e.g. inform by email).

The conjunction or disjunction combination of the elements in H_C and O_C set are also considered. For a set of constraints such as c1  c2  c3, all the conditions c1, c2 and c3 must be held to allow action a to be performed on object o for the purpose p, visible to v; the data is returned according to the granularity level g as long as the retention condition r is valid. Intuitively, in c1  c2  c3, if either of the constraints is valid then the policy can be enforced. The owner constraint (field O_C) specifies the list of possible constraints that the owner is allowed to select when they use the corresponding collector policy. If field O_C is left empty then the data owner cannot choose any additional constraints on the policy that is defined in the field Const of the SysOwnerPolicies table.

2.3 Data collectors’ privacy practices

In the first step, the data collector must specify the structure of the purpose, visibility, granularity and retention lattices. The Chief Privacy Officer (CPO) can extract these lattices according to the organizational structure and the privacy practices set by that organization (as shown in Figure 2). For instance, assume Cyrus is an organization that sells a specific product. The following is one of their privacy policies which is more flexible compared to a privacy policy that restricts data usage to what is defined by the data collector and the data providers have no option to express their opinion. For instance in terms of retention, data providers do not have any control on how long their data is kept in the data collector’s storage:

Ordered date of a product is visible to the staff in the business and marketing department for marketing and competition purposes. However, the customer can limit the access to the business department and for marketing purposes. The customer has the option to let us store only the year of their order date rather than the exact date. This data is kept in the system from three to fifteen months depending on the customer’s choice.

The CPO of this organization reviews the above policy and extracts information about each privacy predicate. The purposes of marketing and competition should be considered. Visibility levels of business and marketing departments should be added to the visibility lattice. In terms of granularity, the customer has the right to restrict access to only the year the book was ordered rather than the exact date. For retention, the user has the option of restricting the organization to keep the data from three months up to fifteen months.

![Figure 2 – An instance of privacy catalogues](image-url)
The organization must have a clear idea about its purposes and their relations to make an expressive purpose lattice. An organizational chart and structure also helps to structure the visibility lattice. The retention and granularity lattices are normally created according to the policies established in the organization. It should be noted that privacy legislative acts and guidelines often limit the privacy practices of the data collectors and prevents possible privacy violations of data providers.

In the second step, the data collector fills the tables SysHousePolicies, SysPurpose, SysVisibility, SysGranularity and SysRetention according to its privacy practices. As shown in Figure 2, these tables store the privacy practices of the collector and the MinAL and MaxAL of each predicate for the corresponding privacy policies. Recall the CPO is responsible for filling privacy policies in these tables and assuring that they are current.

2.4 Storing subsuming nodes in lattices
In our design model we store each of the purpose, visibility, granularity and retention lattices in tables namely SysPLtc, SysVLtc, SysGLtc and SysRLtc, respectively. We assign binary codes to each node of the lattice where the length of the code is logarithmic to n, the number of nodes in a lattice. The codes are in a manner that represents the position of the node in the hierarchy (e.g. the top node is 1111, the bottom node 0000, each intermediate node with a number between the top and bottom using a codification that indicates the intermediate level where is located and where it is connected above and below). This way the lattice can be stored in $O(n)$ space.

In this research we assume the following points: First, since the number of purposes, retention conditions, visibility, and granularity levels are not tremendously large, the resulting lattices and the corresponding tables are not extremely large and storage space is not a concern in this model. Second, these lattices are not subject to frequent change since the privacy policies do not often change. However, when they do, the corresponding tables are adjusted to comply with the new privacy policies.

2.5 Data provider’s preferences
When all the policies are defined by the collector, the data provider can select a policy that suits their preferences. This step occurs when the data provider wants to obtain a service and must provide data. Currently the data provider must agree to a list of privacy policies defined by the collector. In some cases, as with online social networking websites, where data providers are more aware of and sensitive about their data privacy, some tools are provided by the collector to give more control to the data provider. However, the collector still has control over the data and normally the provided data is visible to the collector without any restriction (i.e. in the case of Facebook’s privacy settings, a member can set the comments and pictures to be visible to friends, friends of friends, etc. but there is no way that the provider can limit Facebook’s access to that data).

In our model, when the data provider provides the data, the corresponding collector privacy practices are given to them. The data provider then expresses privacy preferences within the permitted range. These preferences are stored in SysOwnerPolicies. For instance, in the catalogues shown in Figure 2, Alice, the data provider, has specified her purpose, visibility, granularity and retention in the range specified by the collector policy with PlcyID = 2. She also has the chance to select one of the constraints c2, c5 or c6 but may decide not to specify any constraint. In another example, John cannot choose any constraint since he uses the policy with PlcyID = 1 in which no constraint is specified for the data provider by the collector.

The field Validity in the SysOwnerPolicies shows if the privacy policy defined in this catalogue is valid. A privacy policy becomes invalid if the range of the MinAL and MaxAL defined and used by the collector is changed. If the collector changes these levels then there is no guarantee that the data provider who initially agreed to choose a node within the specified range still wants to select the same node. For instance, in the catalogue illustrated in Figure 2, if the collector increases the MinAL of the purpose from Sale to Marketing, it affects P_ID = 1 and changes collector policies with PlcyID = 1 and 3. Hence, the policy selected by John that uses PlcyID = 1 and has selected Sale as the initial purpose becomes invalid. As a result, the value for the Validity field of this record should be false and access cannot be granted unless the data provider is informed of this change and consents to the new policy.

3 LPAAC

3.1 Access control algorithm
The LPAAC algorithm is presented here followed by an example to illustrate the process.

Algorithm – Access decision
Input: Data user’s ID, the action and purpose of performing it on the data, the data provider’s ID, the table and column where the data is located.
Output: Either allow the access and reveal data or deny

ACCESS(User, Action, Purpose, Provider’s ID, Table, Column)
1. If Purpose does not exist in PUR_ALLOWED(User) or VIS_LEVEL(User) =  then
   Return Deny;
2. User_Vis := VIS_LEVEL(User);
3. If the corresponding <Action, Provider’s ID, Table, Column > exists in the join of SOP and SHP then
   a. If NOT SUBSM(R, SOP.Owner_Retention, current retention condition) or SOP.Validity = False then
      i. Return Deny;
   b. If SUBSM(P, purpose, SOP.Owner_Purpose) and SUBSM(V, User_Vis, SOP.Owner_Visibility) then
      i. Return Allow;
   iii. Trigger the constraints selected by the data provider in SOP.Const and the constraints defined by the collector in SHP.H.C.
exists a corresponding record, then in line 3b the system checks if the requesting purpose and visibility level subsumes the intended purpose and visibility level defined by the data provider. If so, then the access is allowed and the data item is revealed with the granularity level defined by either the data provider in the Owner_Granularity field of the SysOwnerPolicies catalogue or the one selected by the collector. The granularity levels can be developed as user defined functions or stored procedures that generalize the data according to the data type. The following example illustrates the access control system.

Example 3. When Eve, as a data user, logs in to the system, her visibility level is checked against SysUserVis (the corresponding catalogues are shown in Figure 2). Once she submits a query to see the order date of a book purchased by Alice, she specifies her purpose (e.g. Competition). A possible form of the query may be as follow (see [8] for the version of SQL that encompasses privacy).

```
SELECT date FROM order
WHERE Prov_ID = Alice
REASON competition;
```

The following procedures are performed to control Eve’s access:

1. Eve’s purpose is checked against SysUserPur to observe if she is authorized to have this purpose (access denied for invalid purpose).
2. Since Eve is looking for the order date of a book purchased by Alice, the system checks in SysOwnerPolicies and SysHousePolicies to see if a “valid” policy defined by Alice exists for this data item (field date of table order). Further, the system checks if the retention condition of that policy is still acceptable.
3. If Alice’s intended purpose and visibility are subsumed by Eve’s purpose and visibility level, then the access is granted. Note that the data is revealed in the granularity level defined by Alice (i.e. according to the policy shown in Figure 2. Eve can only see the year the book was ordered). After granting the access, the constraints of SHP.H.C and SOP.Const for that policy are triggered.
4. If either purpose or visibility predicates is not subsumed then the access is not granted. Note that in this example, Alice’s intended purpose (Marketing) is subsumed by Eve’s purpose (Competition) so the access is granted.

3.2 Implementation

We study the efficiency and scalability of our privacy-preserving model using the data and query set generated from the TPC-H benchmarking standard. We implemented the privacy catalogues (Figure 2) in MySQL and wrote a software application using Microsoft VB.Net to act as a wrapper around the database.

We assumed our purpose lattice and visibility lattice have 100 nodes each. For each query the most costly part is to identify if the requested nodes are subsuming the intended nodes in the lattice or not, which depends on the size and structure of the lattice. However, since we have stored our lattices in the format described in Section 2.4, this search can be done in \(O(n)\) where \(n\) is the number of nodes available in the lattice.

Furthermore, since the privacy policies established by the data collectors involve a small number of purposes and visibility levels, and by using the indexing method described, we can easily optimize tables searches as presented in Table 1.

4 CONCLUSION AND FUTURE WORK

We have introduced the lattice-based privacy aware access control (LPAAC) model. The LPAAC model captures four critical privacy concepts: purpose, visibility, granularity, and retention captured in a lattice structure. We have illustrated our contributions with a relational design. In particular, key details affecting the underlying catalogues and privacy agreements between the data provider and collector are discussed. We also implemented privacy catalogues.

We argue that the data providers should receive the opportunity to express their privacy preferences so they have control over their personal data. The LPAAC model opens the research opportunity to design access control systems that allow data providers to describe their privacy preferences. Some open questions in designing such access controls are: designing algorithms for conflict resolution between the collector and the data provider’s constraints; finding a hierarchical structure to capture constraints that facilitates managing them; solutions for handling the effect of changes in the collector’s privacy policies and the data provider’s privacy preferences; and designing a user interface to effectively provide available options based on the lattice structure and the collector’s privacy policies to the data provider.

Bibliography

5. K. Ghazinour and K. Barker: Capturing P3P semantics using an enforceable lattice-based structure. PAIS 2011: 4