P4A: A New Privacy Model for XML

Angela C. Duta and Ken Barker

University of Calgary
2500 University Drive
Calgary, Alberta, Canada

Abstract. We propose a new privacy model for XML data called Privacy for All (P4A) to capture collectors privacy practice and data providers privacy preferences. Through P4A data collectors specify the purpose of data collection along with recipients, retention time and users. Data providers can agree to the collectors' practice or impose their own privacy preferences. P4A offers more flexibility to both data collectors and providers in specifying privacy statements and preferences, including but not limited to full permission, denial, and conditional access to information.
A privacy practice defines purposes, recipients, retention period, and uses of data collection. Data providers share their private information with data collectors under restrictions specified by privacy preferences. P4A offers individuals multiple options for restrictions such as conditional access, return results as range intervals for each data item and purpose.

KEYWORDS: privacy preference, privacy statement, flexible privacy policy, privacy map

1 Introduction

Several representations for privacy policies have been proposed in the literature to address the growing concern of private information protection. Current research in the database community considers privacy [3] [4] [14] [16] in databases where data providers[1] agree to a set of predefined policies. This is a restrictive solution as data providers have limited options. If they do not agree with any of company's policy they are left with no real option [15] except to sign an unsatisfying privacy agreement or to refuse the company's services. Neither option is considered acceptable.

We propose a solution to preserve data privacy where providers set their own conditions. Data collection has two major players: the data collector and the data provider. Both have different viewpoints regarding privacy. The collector's view is expressed as the privacy practice and the provider's view is captured in privacy preferences. A privacy policy considers two major elements: data and the purpose of its use. Each provider decides which personal information is private and all possible purposes for which it can be used.

[1] **Data provider** or **provider:** people that share their private information with collectors possibly in the exchange for a product or service, i.e. patients, customers, *etc.*

1.1 Motivation

Agrawal *et al.* identified the ten principles of privacy in databases. Two of them, the principles of limited collection and limited use require that only data necessary to fulfill specified purposes is collected and used. However, a company could have several "minor" purposes in addition to its main one, as it has several additional business activities in addition to its main one. Thus, the principle of limited use allows for a broad variation depending on company goals. Instead of leaving this decision to the collector, we suggest that data providers determine what data is reasonable to be used for each purpose. Obviously, providers options generates considerable overhead that must be resolved. Categorizing privacy policies in hierarchies is not a viable option as offering predefined privacy contracts is not flexible enough and a multitude of options can be expressed by providers (thus, no hierarchy). Current approaches to privacy do not offer the flexibility people desire because they do not treat each privacy contract individually. This is the challenge we address in this paper: each data provider expresses individual options for privacy with a minimum of overhead for the database system.

1.2 Contributions

This paper proposes a new XML data model that considers privacy protection called Privacy for All (P4A). In P4A privacy policies offer maximum flexibility to each provider of personal information in choosing the desired protection. Collector's privacy practice is included in the metadata and forms the general guidelines for data query. In P4A providers actively decide upon the use of their data by allowing, denying access to it, or setting additional conditions that must be meet before access to their data is allowed. Privacy preferences are stored in a privacy map. P4A has several advantages: (1) sensitive data is used according to providers preferences; (2) data providers can request conditional access to their private data; (3) information leakage is avoided as unaccessible nodes are not reached.

1.3 Paper Organization

The balance of this paper is organized as follows. Section 2 summarizes related current approaches in privacy and security. Section 3 defines the problem and introduces a working example that is used next in the privacy model description (Section 4). Some conclusions regarding this new privacy model for XML are drawn in Section 5.

2 Related Work

Work in the privacy area must look at its sociological aspect. Privacy is characterized differently by philosophers, sociologists, economists, computer scientists, *etc.* [21]. Our research incorporates the current trend to create more complex privacy rules such as "no-release-by-legal-right" [21] to protect individuals. A simple solution to privacy protection is to perturb sensitive values [3]. Creating range values to hide sensitive values if it is in concordance with the purpose of a data query is reasonable but not sufficient. Research in the privacy area is developed in two main directions: (1) regulate the use of data stored in databases and (2) regulate data collection during Web surfing. In the first area, work on Hippocratic databases [4] [16] [19] translate the Hippocratic Oath into modern privacy policy tables. Regulation of data collection for Web users is first considered by W3C through the privacy specification standard called the Platform for Privacy Preferences (P3P1.0) [6]. Subsequent research criticizes P3P because it does not guarantee Web sites have the means to implement and respect their privacy policies ([2], [7], [10]). Social Contract Core (SCC) [15] extends P3P by allowing users to choose privacy preferences that suit them. Users "vote" for the policy that is closest to their preferences so they are able to visit the site. Both the collection of private data through the Internet and its use are considered in some approaches. The Platform for Enterprise Privacy Practices (E-P3P) [14] defines a methodology to enforce P3P by using an *Obligations Engine* to fulfill contractual obligations (i.e. delete records older than 3 years) and a *Policy Evaluation Engine* to control user access to personally identifiable information. The Paranoid Platform for Privacy Preferences (P4P) [2] envisions a world where personal agents help individuals to control the destination, type, scope and duration of use of released personal information. Our work considers XML data collections where each query has a purpose assigned as in Hippocratic databases [4]. We extend the Social Contract Core [15] by allowing providers to decide on the accessibility of each data item not just offering them several policy options. From this perspective we support and incorporate in P4A the use of authorization table for each customer that accommodates individual privacy preferences as in the approach of Massacci *et al.* [19].

In the security area, a standard XML access control XACML [17] that deals with specification of complex policies is created as a component of distributed systems. The advantage brought by XACML is related to its ability to integrate in heterogenous systems and act as a successful intermediary language due to the XML extensible and expressive format.

3 P4A Privacy Model

3.1 Problem Definition

As pointed out by Walters [21] the term *privacy* has several definitions, some more detailed, not only among categories of scientists, but the general public as well. We start defining privacy by looking at some definitions.

First the definition provided by *The Canadian Oxford Dictionary* [1] states that: Privacy is *the state of being private and undisturbed; a persons right to this; freedom from intrusion or public attention; avoidance of publicity.*

This definition probably captures our view of privacy in our day-to-day activities relating it to *anonymity*. The *Canadian Privacy Act*[2] refers to the legal aspects of privacy and provides a more complete definition. In this act, the term used is *personal information*, as the key element of privacy.

"Personal information" means information about an identifiable individual that is recorded in any form including, without restricting the generality of the foregoing.[3]

The definition provided by the Canadian Privacy Act sets the grounds for privacy in databases. The *Organization for Economic Cooperation and Development* sets the guidelines for collections of private data in the 1980s, later used by governments in legislative privacy standards. The OECD defines eight principles[4] for data collection and usage with respect to privacy: (1) Collection Limitation Principle, (2) Data Quality Principle, (3) Purpose Specification Principle, (4) Use Limitation Principle, (5) Security Safeguards Principle, (6) Openness Principle, (7) Individual Participation Principle, and (8) Accountability Principle.

Whatever the definition for privacy or personal information we use, just by looking at the definition of database systems it is clear that they are not yet ready to handle it. A database is defined as *"a collection of related data"* [9], and a database management system is *"a collection of programs that enables users to create and maintain a database"*. Nothing about privacy is specified in these definitions. Database administrators are usually concerned in current database systems by user authorizations referred to as discretionary access control. Even more, one reason for collecting data is to apply data mining or knowledge discovery to search data for patterns and "discover" new information.

Thus, a different approach must be developed in creating database systems, one that incorporates privacy. A simplified view of a database management system (DBMS) environment is composed by: application, access control, query management, concurrency control, and metadata with stored data. Privacy must be considered and implemented at each of these components. Implementation at the application level generates no changes to the database model and is the most flexible. However, it is the least reliable because the responsibility lies with programmers. Frequent changes to an application may leave open holes in privacy protection. If data is utilized also by a different application, then the process of implementing data privacy starts again. Access control may solve some of the privacy issues by accepting only authorized users to access data. However, a user once authorized, has access to data no matter what privacy concerns are specified. Query management and concurrency control rely on the data model. Any additions, such as privacy, to the query management should be reflected in the underlined data model. Last, but not least, is the data model. This includes data descriptions, that is, the domains. All the other components of a DBMS are based on the data model. By adding privacy to the data model, the database becomes fully equipped to handle privacy more reliably regardless of the appli-

[2] http://lois.justice.gc.ca/en/P-21/text.html

[3] http://lois.justice.gc.ca/en/P-21/text.html

[4] http://www.cdt.org/privacy/guide/basic/oecdguidelines.html

cation, access control, query management, and concurrency control mechanisms implemented. However, as components of security policies are implemented at all levels of a DBMS for increased reliability, efficiency, and protection, so should privacy policies.

This work is aimed at constructing a data model for XML data to incorporate privacy being the milestone of a privacy concerned database. The following is the definition of a privacy concerned database we refer to in the rest of the paper.

Definition 1. ***[Privacy Concerned Database]*** *A* privacy concerned database *is a database where private data is stored, retrieved, and used according to purposes to which their owners have agreed as specified by associated privacy policies (privacy practice and providers' preferences).* ■

An example of privacy concerned database is presented next.

3.2 A Working Example

BrightStar is a financial institution offering credit card services to individuals from which it collects private data as depicted in Figure 1. BrightStar requires personal information such as name, address, phone number, SIN, employer, income, credit card information, and transactions on it. To fulfill its business goal, BrightStar performs credit evaluations, studies clients purchase habits, and exchanges credit information with other financial institutions regarding common customers. It also performs data mining on collected data to determine new trends in customers' behavior likely to influence their credit score, to suggest new financial products (credit cards or loans), or to detect suspicious transactions. Affiliated banks, such as TotalBank and NorthBank, query the BrightStar's database regarding credit information to perform their credit evaluations. BrightStar has agreements with several merchants, such as SellStar LTD and SellAll LTD, to sell sell non-financial products to BrightStar's customers. BrightStar is a modern institution that wants to respect its customers privacy concerns. It decides to allow its clients to choose how private data is used for different purposes implied by its business activities by implementing P4A.

3.3 Privacy Metadata

Privacy policies permit data owners to actively determine the purpose for the data collection but do not provide the means to verify their correct implementation. A privacy policy evaluates the legitimacy of a query with respect to its purpose and requested data before the query is executed. Unanswerable queries are rejected with no additional waste of computational time. Further, information leakage is avoided as unaccessible nodes are never reached.

A complete definition of a privacy policy must include a combination of privacy constraints $< Purpose, Object, Recipient, Retention >$ and access constraints $< Purpose, Object, User >$ as suggested in Hippocratic Databases [4], where *Recipient* and $User$ refer to who has access to data. We argue that

```
Object                              Purposes/Access
clients
├─ client *
    ├─ clientID
    ├─ personalInfo
    │   ├─ name                     T H E V S (r+)
    │   ├─ dateOfBirth              T H M S (r+)
    │   ├─ SIN                      T E M V (r+)
    │   ├─ address
    │   │   ├─ street               E M V (r+)
    │   │   ├─ city                 H E M P V S (r+)
    │   ├─ phone                    H E N (r+)
    │   ├─ income                   T E M V S (r+)
    │   ├─ employer *               V (r+)
    ├─ creditCards
        ├─ card *
            ├─ accountNo            H E P V (r+) A (w+)
            ├─ type                 H E M V (r+) A (w+)
            ├─ limit                H E M V (r+) A (w+)
            ├─ rate                 H E M V (r+) A (w+)
            ├─ balance              H E M P V (r+) A (w+)
            ├─ transactions
                ├─ transaction *
                    ├─ date         H M S (r+)
                    ├─ merchant     H M V S (r+)
                    ├─ services     H M V S (r+)
                    ├─ amount       H E M V S (r+)
```

Fig. 1. Purpose-constraints

the Hippocratic Database [4] introduces unnecessary redundancy in its privacy tables by specifying purposes for each object in connection with recipients (in privacy constraints) and users (in access constraints) as detailed above. In Hippocratic Databases [4] privacy policies are defined in addition to security policies $< User, Access\ right >$ that specify authorizations for users. It is more important that the *access right* be correlated to the query purpose rather than to the subject. Thus, we suggest the following privacy model, P4A, formed by Purpose-constraints $< Object, Purpose, Access\ right >$ (Figure 1), Recipient-constraints $< Recipient, Purpose, Retention\ time >$ (Table 1), and User-constraints$< User, Recipient >$ (Table 2). Purpose-constraints (Figure 1) structure captures the purposes for which each data is collected. For example, the name of a client is used for "income tax purpose" (T) to refer to its declared income, for "purchase habits purpose" (H) to call the client in case a suspicious transaction is executed in its account, for "credit inquiries purpose" (E) when other banks or financial institutions want to verify his credit history, for "credit evaluations purpose" (V) conducted by BrightStar, and for "sell products purpose" (S) by BrightStar's partners that sell non-financial products. The type of access required for these purposes is specified between parenthesis following purpose specification (i.e. for read is r+). Recipient-constraints (Table 1) describes connections between recipients and purposes, by identifying which recipients are entitled to query for which purposes. The retention time is included in this table as it specifies the duration data is stored and used for a particular purpose. This time can vary for each recipient and purpose. When a data element is not required by any purpose of any recipient then it must be deleted from the database according to the Minimum Retention Time Principle enunciated by Agrawal *et al.* [4]. The third component of our proposed model, the user-constraints structure extends security policies by specifying all users associated with each recipient.

Table 1. Recipient-constraints

Recipient	Purpose	Retention
Canada Revenue	(T) income tax	5 years
BrightStar	(H) purchase habits	2 years
TotalBank	(E) credit enquiries	1 month
BrightStar	(M) data mining	2 years
BrightStar	(P) payment	while ∃ card
BrightStar	(V) credit evaluation	6 months
SellStar LTD	(S) sell products	1 months
BrightStar	(A) approve credit card	3 month

Table 2. User-constraints

User	Recipient
Alice	BrightStar
Susan	BrightStar
Bob	TotalBank
Oliver	SellSTar

P4A defines attributes' accessibility for each purpose in the Purpose-constraints table (Figure 1) and recipients rights to query for specific purposes in the Recipient-constraints table (Table 1). In P4A, access rights are associated with query purposes as the concern in privacy policies is focused on the purpose rather than on the user as in security policies. P4A has the benefit of less redundancy compared with Hippocratic Databases [4] as relationship purposes - objects are specified once. Further, the retention period is correlated with query purposes instead of objects. A purpose requires instances of several objects to be available. Instead of multiple retention periods for combinations purpose-object [4] only one tuple per purpose is specified. An object is queried with multiple purposes, so instances of an object are stored for as long as one purpose needs this data. Thus, the retention period is included in Recipient-constraints table and it allows multiple specifications for one purpose depending on the recipient. In P4A, different recipients query private data with the same purposes but have different data visibility. The retention time for a collector is the maximum period allowed to store data. The retention time for a recipient is the maximum allowed time to query data as specified by the purpose. Table 2 specifies users[5] that are allowed to query data on behalf of each recipient.

P4A implements conditional access in addition to traditional permission and denial. The following sections describe the access codes and conditions that apply to private data.

[5] An individual or group that accesses data stored in a database on behalf of the recipient or collector (not the one "on whose behalf a service is accessed and for which personal data exists" [6] as in W3C terminology).

3.4 Complex Conditions

A major contribution of our approach is to offer more flexibility to data providers in expressing a variety of conditions that must be respected to have access to data. In previous approaches the only options providers had were permission or denial. Additional restrictions are included in P4A such as: interval values, perturbed values, and conditions that refer to the knowledge of more “private” information. For example, a condition is “my name can be accessed only if the user provides my correct client ID”. This condition denies execution of queries such as “who are BrightStar’s clients?” or “is X a client?”. We use “access” as the general term for any type of access be it read, write, update, append, or delete.

Using the database of BrightStar represented by the XML tree from Figure 1 we demonstrate several options a data provider or collector should have when defining a privacy policy. To simplify the presentation we only consider purposes for a single collector. Some examples of restrictions data owners may request are: my SIN number can be accessed only if the user provides my correct name and date of birth; give an approximate address (i.e. only street name but no number) for data mining purposes instead of my exact address; use terms like permanent and temporary resident rather than social insurance numbers; for third parties asking for credit references provide amounts spent on my credit card for transactions older than a year but not the merchant or service, *etc.*

Inclusion of such conditions in the privacy model requires conditional access codes in addition to permission and denial as previously considered in security policies. The next section introduces the access types in P4A.

3.5 Access Codes

Definition 2. ***[Access Code]*** *The* access code *associated with a node in the XML tree expresses its accessibility in relation to a query purpose in a privacy concerned database. The set of access codes is α={yes (Y), conditional(C), range (R), conditional and range (Q), no (N)}.* ■

Providers and collectors specify access types for leaf nodes where information is stored. Table 3 depicts the proposed access codes for leaf nodes. Code No (N) means the leaf node must not be accessed while Yes (Y) allows unconditional access to it. Code Range/Perturbed value (R) permits access if a table exists for this node to perturb sensitive value either by specifying interval values (i.e. age below 20, 21-40, 41-60, and above 61) or key terms (i.e. young, mature, old); otherwise the access is denied. Code Condition (C) allows access to this leaf node if the value of another node is known. The condition we suggest here is equality (or non-equality) for privacy protection. This condition should be applied to nodes that store more “private” data rather than public information. For example, it is preferable to use a condition based on SIN or date of birth rather than name. The code Q (perturbed values and condition) is for nodes where perturbed values are returned when the specified conditions are true;

Table 3. Access specification for leaf nodes

Code	Access	XQuery representation
N	No access	-

R	Perturbed or interval value	for $x in doc (“*doc.xml*”) *path*
		return if ($x/*item* < *value*)
		then <item>“below value”</item>
		else <item>“above value”</item>

C	Conditional access	for $x in doc (“*doc.xml*”) *path*
		where $x/*item condition*
		return $x

Y	Unconditional access	for $x in doc (“*doc.xml*”) *path*
		return $x

otherwise the access is denied. The proposed set of access codes is represented by $\alpha = \{\mathrm{Y, R, C, Q, N}\}$, where α is a lattice based on Definition 3). The set of constraints that are associated with access code C,R, and Q is represented by Ω.

Definition 3. *[α-**Order**] There exists a partial order for access codes from the most permissive to the least permissive noted $>_p$ as follows: $Y >_p R >_p Q >_p N$ and $Y >_p C >_p Q >_p N$.* ■

In this approach, *operation codes* are considered in addition to the access codes to create finer and more restrictive access.

Definition 4. *[**Operation Code**] The* operation code *associated with a node in the XML schema tree expresses the permitted operations to be performed on this node in relation to a query's purpose in a privacy concerned database. The set of operation codes is β =\{no operation allowed (ϕ), read (r), append (a), update (u), delete (d), write (w)\}.* ■

Definition 5. *[β-**Order**] There exists a partial order for operations from the least permissive to the most permissive denoted $<_o$ as follows: $\phi <_o r <_o a <_o w$, $\phi <_o r <_o u <_o w$ and $\phi <_o r <_o d <_o w$.* ■

Each privacy policy specifies access using purpose of access (why is data accessed?) and operation on data (is data read, deleted, updated or just created for this purpose?). These restrictions come in addition to security policies where general user access is specified (i.e. user X is allowed to read data Y and update data Z). By including the operation code in the privacy policy, more restrictions can be imposed in addition to the security policy where the operation allowed for a data item is also according to the purpose of the query. For instance considering the example described in Section 3.2, user X from BrightStar has rights to update item *account rate*. However, depending on the purpose of X's query, X is allowed to write on *rate* when creating a new credit card (purpose *approve credit card*), and only to read this information when determining credit score (purpose *credit evaluation*).

4 Privacy Maps

A *privacy map* is proposed to store privacy preferences and practices for XML documents.

Let Λ be the set of leaf nodes from the tree associated with an XML document schema, Ψ the set of purposes for data collection, and Δ the set of data providers. The set of collectors is symbolized by Υ and includes also third party recipients that may obtain data from the original collector. In this approach we consider close privacy policies where all permissions are specified in the policy.

Definition 6. ***[Privacy Practice Map (PPraM)]*** *The* privacy practice map *is a function* $PPraM : \Lambda \times \Psi \times \Upsilon \rightarrow \alpha \times \beta \times \Omega$. ■

PPraM expresses collectors privacy practice. For each leaf node, $\lambda \in \Lambda$ an access code $a \in \alpha$ and an operation code $b \in \beta$ are specified in relation to each query purpose $\psi \in \Psi$. If the operation code is for conditional access (R, C, or Q), then conditions $\omega \in \Omega$ are specified; otherwise no condition is considered (ϕ). The traditional $r+$ and $r-$ for read allowed/denied are extended to (Y, r, ϕ), (N, r, ϕ), and $(C/R/Q, r, conditions)$.

Example 1. Two examples of practice statements from PPraM are depicted below for node SIN *when its instances are queried with purposes* data mining *and* purchase habits*:*

PPraM (SIN, data mining, BrightStar) = (R,r, "permanent/temporary resident")

PPraM (SIN, purchase habits, BrightStar) = (N, r, ϕ).

Definition 7. ***[Privacy Preference Map (PPreM)]*** *The* privacy preference map *is a function* $PPreM : \Lambda \times \Psi \times \Delta \times \Upsilon \rightarrow \alpha \times \Omega$. ■

PPreM is a collection of privacy preferences. It specifies the access code $a \in \alpha$ and the conditions associated with it (if any) $\omega \in \Omega$ for each leaf node $\lambda \in \Lambda$ for purposes $\psi \in \Psi$. The recipient $\upsilon \in \Upsilon$ is specified for cases where third parties query private data stored by the collector.

Example 2. An example of a preference from PPreM for an instance of node SIN *storing private information about provider John Doe when it is queried with purpose* payment *is PPreM (SIN ="123 456 789", purchase habits, John Doe, BrightStar) = (N, r, ϕ) or (N, , ϕ) where no access and, thus, no operation (space) and no condition ϕ is granted to BrightStar when querying data provided by John Doe with purpose purchase habits.*

The proposed privacy model P4A for XML is depicted in Figure 2. P4A extends the *Purpose constraints* table proposed in Section 3.3 by having two materializations: PPraM and PPreM. Additional privacy metadata is formed by two relations $Recipient-constraints < \Upsilon, \Psi, retention >$ and $User-constraints < U, \Upsilon >$, where U is the set of users. The first shows recipients allowed to retrieve

Privacy Practice Policy	
PPraM	$< \varLambda, \varPsi, \varUpsilon, \alpha, \beta, \varOmega >$
Recipient-constraints	$< \varUpsilon, \varPsi, retention >$
User-constraints	$< U, \varUpsilon >$
Provider Privacy Policy	
PPreM	$< \varLambda, \varPsi, \varDelta, \varUpsilon, \alpha, \varOmega >$

Fig. 2. P4A privacy model

data for different purposes, and the second depicts connections between users and collectors/recipients.

Let $\lambda \in \varLambda$ be a leaf node from an XML schema, $\psi \in \varPsi$ a query purpose, and $u \in U$ a user authorized to retrieve data for recipient $\upsilon \in \varUpsilon$. A data request is expressed by the set $Q(\lambda, \psi, u, \upsilon)$. The privacy practice (PPraM, Recipient-constraints, and User-constraints) is first queried and the answer is $Q(\lambda, \psi, u, \upsilon) = (a, b, \omega)$, where $a \in \alpha$, $b \in \beta$, and $\omega \in \varOmega$. If $a \in \{R, C, Q\}$ additional conditions are included in the query as *where* clauses (see Table 3) and the query becomes Q_{PPreM}. It is next performed on PPreM if $a \neq N$ with $Q_{PPreM} < \lambda, \omega_{PPraM}, \psi, \upsilon >$, where ω_{PPraM} represents the conditions specified in PPraM for leaf node λ. The answer to Q_{PPreM} is the set $< \lambda, a_i, \omega_i, \delta_i >$, where δ_i is the subset of data providers where $a_i \neq N$ in PPreM for the queried leaf node λ. Only for those the query is executed on the data document.

4.1 Privacy Practice Map (Schema Level Statements)

PPraM contains schema level authorizations defined by collector with respect to purposes for which data is collected. Figure 3 depicts an example of PPraM where access codes and operations are attached to each leaf node. The capital letters refer to access codes and the small letters to the operations allowed for each purpose. Consider the second leaf node, *name*, that has assigned the codes Yr Cr Yr N N Yr Yr N meaning that no access is allowed when querying with purposes M, P, and A, unconditional access when query purpose is T, E, V, and S, and conditional access when purpose is H. The allowed operations for purposes with permission are specified using small letters: read (r). The operation code is omitted if the access is denied and is represented by a space in Figure 3.

If the access code is C then a condition must be specified. Table 4 depicts conditional privacy statements expressed by BrightStar. For example, queries with purpose H are executed on transaction and its subnodes only if transaction date is more than one year old. Access code R requires one or more conditions to specify the interval values (i.e., when dateOfBirth is queried with purposes H,M, or S age intervals are retrieved as in Table 4). Access codes Q combine the requirements of both codes C and R (i.e., values retrieved for card limit in queries with purpose E, when clientID is known, are bad, OK, or good credit). This means that a query with purpose E that tries to retrieve *limit* for all clients will not be executed. Instead, queries address to a specific client are performed if the correct client ID is provided.

	T	H	E	M	P	V	S	A
clients								
client *								
clientID	N	N	N	N	N	N	N	N
personalInfo								
name	Yr	Cr	Yr	N	N	Yr	Yr	N
dateOfBirth	Yr	Rr	N	Rr	N	N	Rr	N
SIN	Yr	N	Cr	Rr	N	Yr	N	N
accountNo	N	Cr	Cr	N	Cr	Cr	N	Yw
limit	N	Yr	Qr	Rr	N	Cr	N	Yw

Fig. 3. Extras from Privacy Practice Map (PPraM)

Fragments of the XML Schema for PPraM presented in Figure 3 are depicted in Figures 4 and 5. Attributes *privacyPolicy*, *purpose*, *access*, and *operation* are added to the extended XML Schema for the collector to specify the purpose of data collection, access codes, and operations allowed for data query (Figures 4 and 5). Purposes are specified once at the beginning of the XML Schema (Figure 4) in the element named *<privacyPolicy>* . Each entry specifies a purpose name (for example in line 2, *purpose = "T"*) and its description (*description = Income Tax*). The attributes "access" and "operation" are added in the nodes' description in addition to attributes name, type and max/minOccurs (Figure 5 lines 1, 3, 6). Each entry in the *access* and *operation* attributes correspond to a purpose defined in the <privacyPolicy> element and in the order specified there. The associated set Ω of conditions and restrictions is specified in Figures 6 and 7 using XQuery syntax. The value in Figure 6 for client ID identified by the XQuery variable $clientID (line 6) is required by conditions and collected through application from users. Figure 7 gives an example of a *Perturbed Values* structure where restrictions are defined for *age* to return values such as *youth, elder, mature* calculated based on the current date and the date of birth. In PPraM Purpose-restrictions, perturbed values are specified using references to data stored in *Perturbed Values* structure (see Figure 5 lines 4, 7, 8, and 9). This technique minimizes redundance in restriction specifications as identical conditions are specified once.

Table 4. Some conditions associated with PPraM

Node	Purpose	Access code	Condition	Perturbed value or range interval
name	H	C	know clientID	
dateOfBirth	H, M, S	R	age < 21 /21..59 / > 59	youth / mature / elder
SIN	E	C	know name, dateOfBirth, and address	
	M	R	first digit of SIN $\neq 9$ / $= 9$	permanent / temporary resident
limit	E	Q	know clientID and limit ≤ 500 / 500.. 2000 / > 2000	bad/ OK / good credit
	M	R	limit ≤ 500 / 500.. 2000 / > 2000	bad/ OK / good credit
	V	C	know clientID	

```
<privacyPolicy>
  <purpose = "T" description = "Income Tax">
    <recipients>
      <recipient ="Canada Revenue" retention = "5 years"/>
    </recipients>
  </purpose>
</privacyPolicy>
```

Fig. 4. Extras from privacy extended schema: purposes, recipients, and retention time (Collector's Recipient-constraints)

```
<element name="clientID" type="string" access="NNNNNNNN" operation=""/>
<element name="personalInfo">
  <element name="name" type="string" access="YCYNNYYN" operation="rrr--rr-">
    <conditionID purpose="H"> 1 </conditionID>
  </element>
  <element name="dateOfBirth" type="string" access="YRNRNNRN" operation="rrr--rr-">
    <rangeID purpose="H"> 1 </rangeID>
    <rangeID purpose="M"> 1 </rangeID>
    <rangeID purpose="S"> 1 </rangeID>
  </element>
```

Fig. 5. Extras from privacy extended schema clients.xsd: definition of elements, access and allowed operations (Collector's Purpose-constraints)

4.2 Privacy Preference Map (Data Level Authorizations)

Preferences expressed by data providers are stored in PPreM. Each data provider defines their own privacy policy according to which data is accessed. A schema denial does not allow access to an attribute regardless of the data provider authorization. A schema permission allows access to data parts with data permission. In conflict resolution denials have a higher priority than permissions. Schema and data level restrictions (C, R, and Q) specified for a node must be both satisfied before allowing access to data.

```
<conditions>
  <condition>
    <conditionID> 1 </conditionID>
      <restriction>
        for \$x in doc("clients.xml") /clients/client/personalInfo
          where \$x/clientID=\$clientID
          return \$x/name
      </restriction>
  </condition>
</conditions>
```

Fig. 6. Fragment from the XML presentation of the conditions in PPraM Ω

```
<pertubedValues>
 <range>
  <rangeID> 1 </rangeID>
   <restriction>
    for \$x in doc("clients.xml") /clients/client/personalInfo/dateOfBirth
     return if ((\$x-\$currentDate) < 21)
      then <age> youth </age>
      else if ((\$x-\$currentDate) > 59)
        then <age> elder </age>
        else <age> mature </age>
   </restriction>
 </range>
```

Fig. 7. Fragment from the XML presentation of the perturbed values in PPraM Ω

For example, suppose data provider A allows access to item *limit* to queries with purpose *data mining.* Also, suppose at the schema level queries with purpose *data mining* are authorized to access attribute *limit* as an approximate value (see Figure 3). The collector's privacy policy requires only an approximate value, so data is retrieved as an interval or perturbed value. During query execution before each value of the attribute *limit* is retrieved its privacy authorization must be checked.

In P4A privacy maps, each attribute value has attached multiple access authorizations, one for each purpose defined in document schema. From this perspective, our approach is similar to polyinstantiation [18] [20]: multilevel databases provide multiple "aspects" of each data called instances; our data document provides multiple authorizations for each piece of data. In polyinstantation there are multiple access rights to each node, one access right for each clearance level (top secret, secret, public, *etc.*). Data is accessible or not depending on the user authorization and the node clearance level. In privacy, there are multiple access codes for each node, one access code for each purpose. A node is transparent or not to a query depending on the purpose and the access code assigned to it. However, in our approach there is a single "materialization" of data as oppose to multiple instantiations in multilevel databases.

Figures 8 and 9 depict a fragment of the XML data file and corresponding preferences of provider A. Provider's preferences are specified through attribute *pref* automatically generated from the extended XML Schema. We suggest that privacy concerned XML editors and parsers accept attribute *pref* in XML documents without requiring its description in XML schema. Thus, privacy preferences are always portable together with data.

5 Conclusion and Future Work

Each XML database must have in place a mechanism to express and ensure privacy protection. This paper proposes a new privacy model based on an extension

```
<clients>
  <provider id="A">
  <client>
    <clientID>111</clientID>
      <personalInfo>
        <name>John Doe</name>
        <dateOfBirth>01-01-1954</dateOfBirth>
        <SIN>123 456 789</SIN>
        <address>
          <street>123 First St</street>
          <city>Calgary</city>
        </address>
      </personalInfo>
  </client>
  </providerID>
</clients>
```

Fig. 8. Data document clinets.xml

```
<clients>
  <provider id="A">
  <client>
    <clientID pref="">111</clientID>
      <personalInfo>
        <name pref="YCCNNCNN"/>
        <dateOfBirth ="YRNRNNNN"/>
        <SIN pref="YNCRNYNN"/>
        <address>
          <street pref="NNYRNNYNN"/>
          <city pref="NYYYNYNN"/>
        </address>
      </personalInfo>
  </client>
  </providerID>
</clients>
```

Fig. 9. PPreM associated with clients.xml

of XML schema that includes purposes definition and node access codes. The use of P4A gives data providers means to express their privacy preference regarding the limited use of private data. This model offers more flexibility than current approaches in that it allows unconditional and conditional access. Data providers can agree to the collector practice or impose their own privacy preferences.

We are working on implementing our proposed model and evaluate its efficiency. Further, several algorithms must be developed to reduce the privacy overhead and create compressed privacy maps. A social study should be conducted to evaluate the difficulty of expressing complex privacy constraints for non-computer related data providers. More complex or simpler conditions could be found necessary to consider in future approaches.

References

1. *The Canadian Oxford Dictionary. The foremost authority on current Canadian English.* Oxford University Press, Reading, 1998.
2. G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, N. Mishra, R. Motwani, U. Srivastava, D. Thomas, J. Widom, and Y. Xu. Vision paper: Enabling privacy for paranoids. In *Proceedings of the 30th VLDB Conference*, pages 708–719, Toronto, Canada, 2004.
3. Rakesh Agrawal. Privacy in data systems. In *PODS 2003*, page 37, 2003.
4. Rakesh Agrawal, Jerry Kierman, Ramakrish Srikant, and Yirong Xu. Hippocratic databases. In *Proceedings of the 28th VLDB Conference 2002*, pages 143–154, Hong Kong, China, 2002.
5. Luc Bouganim, Francois Dang Ngoc, and Philippe Pucheral. Client-based access control management for XML documents. In *Proceedings of the 30th VLDB Conference*, September 2004 2004.
6. World Wide Web Consortium. The Platform for Privacy Preferences 1.0 (P3P1.0) specification. 16 April 2002. Available at `http://www.w3.org/TR/P3P/` (Last checked on July 14, 2005).

7. Karen Coyle. P3P: Pretty Poor Privacy? A social analysis of the Platform for Privacy Preferences (P3P). June 1999. Available at `http://www.kcoyle.net/p3p.html` (Last checked on July 14, 2005).
8. Ernesto Damiani, Sabrina De Capitani Di Vimercati, Stefano Paraboshi, and Pierangela Samarati. A fine-grained access control system for XML documents. *ACM Transactions on Information and System Security*, 5(2):169–202, May 2002.
9. Ramez Elmasri and Shamkant B. Navathe. Fundamentals of database systems. 2007.
10. Center for Democracy and Technology. P3P and privacy: An update for the privacy community. 28 March 2000. Available at http://www.cdt.org/privacy/pet/p3pprivacy.shtml (Last checked July 14, 2005).
11. Siddhartha K. Goel, Chris Clifton, and Arnon Rosenthal. Derived access control specification for XML. In *Proceedings of the 2003 ACM workshop on XML security*, pages 1 – 14, May 2003.
12. Vaibhav Gowadia and Csolla Farkas. RDF metadata for XML access control. In *Proceedings of the 2003 ACM workshop on XML security*, pages 39–48, May 2003.
13. Abhilash Gummadi, Jong P. Yoon, Biren Shah, and Vijay Raghavan. A bitmap-based access control for restricted views of xml documents. In *Proceedings of the 2003 ACM workshop on XML security*, pages 60–68, May 2003.
14. Gunter Karjoth, Mathias Schunter, and Michael Waidner. The platform for enterprise privacy practices: Privacy-enabled management of customer data. *The 2nd Workshop on Privacy Enhancing Technologies (PET 2002), Lecture Notes in Computer Science*, April 2002.
15. James H. Kaufman, Stefan Edlund, Daniel A. Ford, and Calvin Powers. The social contract core. In *Proceedings of the 11th ACM International Conference on World Wide Web*, pages 210–220, Hawaii, May 2002.
16. Kristen LeFevre, Rakesh Agrawal, Vuk Ercegovac, Raghu Ramakrishnan, Yirong Xu, and David DeWitt. Limiting disclosure in hippocratic databases. In *Proceedings of the 30th VLDB Conference 2004*, pages 108–119, Toronto, Canada, 2004.
17. Markus Lorch, Seth Proctor, Rebekah Lepro, Dennis Kafura, and Sumit Shah. First experiences using XACML for access control in distributed systems. In *XMLSEC '03: Proceedings of the 2003 ACM workshop on XML security*, pages 25–37, New York, NY, USA, 2003. ACM Press.
18. Teresa F. Lunt, Dorothy E. Denning, Roger R. Schell, Mark Heckman, and William R. Shockley. The SeaView security model. *IEEE Transactions on Software Engineering*, 16:593–607, June 1990.
19. Fabio Massacci, John Mylopoulos, and Nicola Zannone. Hierarchical hippocratic databases with minimal disclosure for virtual organizations. *VLDB Journal*, 15(4):370–387, 2006.
20. Walid Rjaibi and Paul Bird. A multi-pupose implementation of mandatory access control in relational database management systems. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases*, Toronto, Canada, 2004.
21. Gregory J. Walters. Privacy and security: An ethical analysis. *ACM SIGCAS Computers and Society*, 31(2):8 – 23, June 2001.
22. Ting Yu, Divesh Srivastava, Laks V.S. Lakshmanan, and H. V. Jagadish. Compressed accessibility map: Efficient access control for XML. In *Proceedings of the 28th VLDB Conference*, 2002.