# Privacy Aware Data Sharing: Balancing the Usability and Privacy of Datasets

Bhume Bhumiratana
Computer Security Laboratory
Department of Computer Science
University of California, Davis
bhumirbh@cs.ucdavis.edu

Matt Bishop
Computer Security Laboratory
Department of Computer Science
University of California, Davis
bishop@cs.ucdavis.edu

## ABSTRACT

Existing models of privacy assume that the set of data to be held confidential is immutable. Unfortunately, that is often not the case. The need for privacy is balanced against the need to use the data, and the benefits that will accrue from the use of the data. We propose a model to balance privacy and utility of data. This model allows both the data provider and the data user to negotiate both requirements until a satisfactory balance is reached, or one (or both) determine such a balance cannot be reached. Thus, this model enables less than perfect privacy, or less than complete utility, as is appropriate for the particular circumstances under which the dat a was gathered and is being held, and the specific use to which it is to be put.

## Categories and Subject Descriptors

K.4.1 [**Public Policy Issue**]: Privacy

## General Terms

Ontology, Data Anonymization, Privacy

## Keywords

Privacy, Information Security, Security Policy, Data Sharing, Ontology, Data Anonymization

## 1. INTRODUCTION

Increasingly, entities in modern society are recognizing the drawbacks of allowing others to access their information. Businesses and organizations collect and store large amount of data in their day to day operations. For example, hospitals keep track of patients' histories to aid in future diagnoses and treatments. They also keep doctors', nurses', and professionals' treatment records for business evaluation as well as personnel performance evaluations.

Having access to this data would greatly benefit many researchers and organizations. However, the Health Insurance Portability and Accountability Act (HIPAA)[1] forbids sharing of individually identifiable health information. Similarly, other consumer protection acts prohibit sharing of customer data in most other areas. The data collector must first anonymize the data before sharing it.

Several recent studies address this problem of privacy-preserving data publishing. All focus on determining how to delete identifying data in such a way that no entity can be uniquely identified. For example, $k$-anonymity [18] transforms the data so that each entity is indistinguishable from $k-1$ other entities.

We take a different approach. We look at the data anonymization problem as the need to balance between privacy and analysis requirements [5]. In this paper, we present an approach using the Web Ontology Language (OWL)[2] to model the knowledge about the dataset. We then use the ontology as a basis for negotiations between the data collector and the data user to balance privacy and analysis requirements.

## 2. RELATED WORKS

Several research areas are related to this problem. Each makes different assumptions and has different constraints.

Recent research in micro-data anonymization inspires our current work. This area focuses on efficiently and effectively anonymizing data in a very small (micro) dataset by altering the content of the dataset to make it impossible to identify a specific individual in the dataset. $K$-anonymity is by far the best known method [18] and various different algorithms implement this technique [17, 11, 19, 13, 3, 7, 1, 15, 20, 12, 16]. These are a "1 size fits all" solution, in the sense that the algorithms perform well on any given micro-dataset regardless of the content or use of that micro-dataset. The techniques use generalization and suppression.

Our focus differs from these methods. We focus on protecting privacy under specific constraints determined by the intended use of the dataset. Lefevre et al [8] and Xiong [10] come closest. The former proposes algorithms that support the generation of anonymous views based on a specific workload focus. The latter proposed a top-down priority scheme for anonymization; this allows a priority to be assigned to some set of Quasi-Identifiers to minimize the perturbation on those specific fields. These results provide methods and

[1]HIPAA - Health Insurance Portability and Accountability Act - http://www.hhs.gov/ocr/privacy/index.html

[2]The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies, and is endorsed by the World Wide Web Consortium. - http://www.w3.org/TR/owl-guide/

algorithms for achieving specific parameters.

Our work is orthogonal to these results. We focus on developing methods for determining those parameters and their limitations using a formal, precise, and expressive negotiation method. The above work fits into our model by performing the underlying anonymization; our work asks how the data collector and data user can communicate and negotiate in order to balance privacy and data usability in a way acceptable to both or, alternatively, that no such acceptable balance exists.

If one views the loss of privacy as a threat to confidentiality, and the need to use the data as a security requirement, one can view this problem as balancing a security and privacy policy. One proposed method [2] describes a system design that protects privacy in collaborative environments in health information systems using a policy-based design that is adaptable to differing policy requirements across various regions. Similarly, Muthaiyah [14] uses an ontology to integrate and enforce security policies in highly heterogeneous environments.

Finally, privacy-preserving data mining opens the question of what can be uncovered through the analysis of several large data sets melded together. Here, the issue is that no one dataset may contain data, or enable the derivation of data, that violates the privacy policy, but the aggregation of many such datasets may enable an attacker to derive confidential data. Broder [4] and Cronin [6] discuss the need for privacy-preserving data mining in more depth.

## 3. ONTOLOGY

An ontology is a formal and explicit representation of a set of concepts within a domain and the relationships between those concepts. In addition, an ontology can be used to reason about the properties about the domain. Examples of concepts in a medical information system domain are medicine, illness, patients' profiles, and doctors' profiles, and examples of relationships between these concepts are illnesses diagnosed for each patient, medicines prescribed, and side effects of the treatment.

An ontology is commonly used as a shared vocabulary to describe, model and conceptualize a real-world domain so that its properties can be analyzed and reasoned with. Ontologies are commonly used in artificial intelligence, medical informatics, web semantics and other area of information sciences.

## 4. PRIVACY MODEL

As indicated above, $k$-anonymity is one of the most widely accepted privacy models. The underlying model is simple. Every dataset record can be viewed as a collection of information. This information is either *Identifier* (*ID*), *Quasi-identifier* (*QI*), or *Attribute*. An *identifier* is information that can be attributed directly to an individual, whereas *quasi-identifiers* are information from which the identity of an individual can be inferred, provided enough of the *QIs* are known, or can be linked with other, external, data. For example, if database $D$ contains the fields name, age, gender, zipcode, social security number, birthdate, blood type, and diagnosis, then name and social security number are usually considered *identifiers* (because they directly identify the individual), while age, gender, zipcode and birthdate are *Quasi-identifiers* (because they are characteristics from which, given enough ancillary information, an individual can be identified), and blood type and diagnosis would usually be considered *Attributes* (because they do not embody personal information about the individual).

Given that distinction, *identifiers* are often removed or completely suppressed, while *QIs* are generalized or perturbed to satisfy some privacy constraints. Commonly, these constraints aim to create groups of indistinguishable records with respect to *QIs*. Thus, group records with similarly featured *QIs* are combined into the same set, so that knowing most or all of these *QIs* still only identifies a set of at least $k$ individuals, where $k$ is the size of the group that the *QI* belongs to.

These are considered generalized algorithm and are very well explored. While these algorithms perform efficiently with very well defined parameters, requirements and privacy guarantees, they offer an "all around best" solution that may not meet the needs of the data user. In other word, these algorithm often do not allow the consumer of the data to restrict the domain and range of its transformation, leading to the anonymized dataset becoming of very limited use to that user. This is because the purpose of these algorithms is to create an anonymized dataset for general use, hence the term *data publishing*.

We propose a new model, called *data sharing*, in which the method of anonymizing the dataset involves two parties, the data aggregator (data collector) and the data user (data consumer).

## 5. ONTOLOGY BASED MODEL

In order to achieve the balance of the privacy protection that the data collector requires (called the *Privacy Policy*, or *PP*) with the utility requirement of the data user (called the *Analysis Policy* or *AP*), we need a model that allows both parties to negotiate their requirements unambiguously. An ontology can model the representation of the underlying data as domain knowledge; we use this as the basis of communication. In this section, we show a basic example of an ontology for a simple data type. We will discuss building the ontology in more detail in section 6.1.

A dataset $D$ is a collection of $n$ records $(d_1, ..., d_n)$. Each record $d_i$ consists of a set of *Identifiers*, *QIs*, and *Attributes*. As previously defined, the elements of the set of *Identifiers* can each, individually, be used to identify the entity corresponding to each record, so we treat each as unique to an individual. One unique entity may be represented by multiple records in the same dataset, so a dataset of $n$ records does not necessarily have $n$ entities. For each unique entity in a dataset, we replace the set of *Identifiers* representing that entity with a single unique identifier. These unique identifiers will be used to represent each distinct individual in our Ontology.

For each type of *QI* and *Attribute*, we build an ontology describng the relationships or classifications between all possible values[3]. For example, if the dataset $D$ has an age field, we define an age class to be of type *integer*. If the dataset $D$ contains diagnosed illnesses, we use the disease ontology called ICD-9-CM[4], which is widely used by societies deal-

[3]If there are infinite possible values, then we simply restrict ourself to those values in the dataset.

[4]The International Classification of Diseases, Ninth Revision, Clinical Modification, managed by The National Cen-

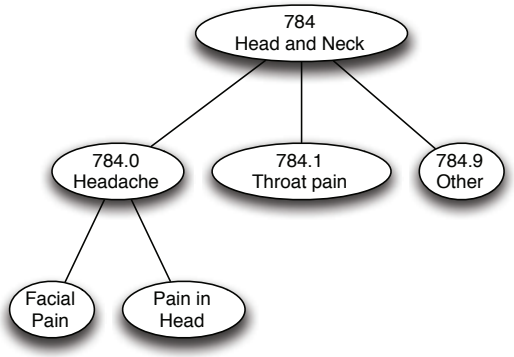


**Figure 1: A sample partial ontology of ICD-9-CM disease classification**

ing with knowledge representation in the domain of medical research. Figure 1 shows a portion of the ICD-9-CM classification for disease and illness related to head and neck.

Given these ontologies, we observe that each record $d \in D$ can be represented by creating an object $r \in Records$ (where *Records* is a class) associated with each record in $D$. Then, for each field in the record, we create a class or set of data properties linking the object $r$ to an entity in each of the previously built ontologies of the *Identifiers*, *QIs*, and *Attributes*. The new *Records* class represents all the privacy in this dataset. If we remove this class and all of its relationships, the ontology is safely anonymized. The rest of the ontology is simply sets of tokens representing values in each column in the dataset $D$, without any cross-column associations among the tokens.

## 5.1 Privacy Policy Model

In the general model, a privacy violation occurs when an attacker is able to link, with some degree of confidence, some of the attributes in the dataset to some small group of individual in the real world. Here, "individual" may be human, or some other object that the dataset describes. However, representing knowledge of all individuals in the real world is difficult. Therefore, we use a stronger model that prevents associating attributes with more than some small number of individuals in the given dataset.

The privacy policy is the set of properties or relationships in the dataset that needs to be removed or altered to prevent a privacy violation. Given the ontology-based model, data collectors can build a privacy policy by defining a set of predicates to classify these relationship in the ontology. For example, the data collector can define a class representing all *visible illnesses*, and define its membership to be the list of diseases whose symptoms are externally visible. Similarly, she can define another class representing all diseases that are terminal. Using some of these classes, an example privacy policy might be:

**Privacy Policy 1**

$$\forall x \in ID | VisibleIllness(x) \wedge TerminalIllness(x) \rightarrow (x \in SuppressGender \vee x \in GeneralizeAge) \quad (1)$$

ter for Health Statistics (NCHS) and the Centers for Medicare and Medicaid Services

This policy states that, if an individual is both terminally ill and has a visible illness, then either the gender of the individual will be removed, age will be generalized, or both. HEre, $VisibleIllness(x)$ and $TerminalIllness(x)$ are functions that take an identity $x \in ID$ and return *true* when $x$ is diagnosed with an illness that is a member of the classes *visible illness* and *terminal illness* respectively, and *false* otherwise. In addition, *SuppressGender* and *GeneralizeAge* are classes defined in the ontology as an annotation that any record or individual represented in this database subsumed by $VisibleIllnesses$ and $TerminalIllness$ will have their gender suppressed and aAge generalized respectively. We will show how these two classes are defined in the next section.

## 5.2 Data Perturbation Model

A meaningful privacy policy must describe how a particular type of data must be altered. In this section, we describe one method of defining these properties in the ontology so that they can be used to reason automatically with a given analysis policy.

In the previous subsection we described *Privacy Policy 1*. It used two classes (*SuppressGender* and *GeneralizeAge*) which we described but did not formally define. In order to do that, we must construct two more classes, namely *preserved* and *altered*, and define their relationship.

**Perturbation Construct 1: Disjointness**

$$Preserved \bigcap Altered = \emptyset \quad (2)$$

This rule says that *Preserved* is the class representing all datatype that must not be *Altered*. *Altered* is a class that encompasses all concepts and objects that will be altered, but it does not describe *how* they will be altered. We can be more specific by adding the *Suppress* and *Generalize* subclasses to the *Altered* class; of course, these too must be disjoint from *Preserved*. We can also make *SuppressGender* a subclass of *Suppress* and *GeneralizeAge* a subclass of *Generalize* as follows:

**SuppressGender**

$$\forall x \in ID | x \in SuppressGender \rightarrow TransformGender(x) = * \quad (3)$$

**GeneralizeAge**

$$\forall x \in ID | x \in GeneralizeAge \rightarrow TransformAge(x) \in GeneralizeAgeGroup \quad (4)$$

In this, *GeneralizeAgeGroup* is another group of classes that partition entities based on age. It forms a hierarchy for generalization. Figure 2 shows an example of the ontology.

## 5.3 Analysis Policy Model

We define an analysis policy as a set of requirements stating the characteristics of the dataset that must be preserved in order for the dataset to be useful. Equation 5 shows one of the simplest such constraint policies.

**PreserveAge**

$$\forall x \in ID | TransformAge(x) = Age(x) \quad (5)$$

This policy states that the data user requires the age of all individuals $x \in ID$ to be preserved unchanged.

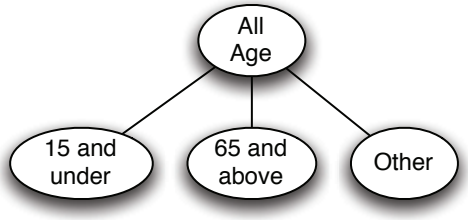


**Figure 2: A sample GeneralizeAgeGroup ontology**

# 6. SHARING FRAMEWORK

The previous sections outlined the elements and constructs making up the ontology. This section pulls those constructs and elements together into a coherent framework for negotiation. The steps are:

1. The data collector builds an ontology $O_D$ representing the knowledge about the records in the dataset $D$.

2. The data collector defines privacy constraints $P_1, ..., P_k$ to identify all the data attributes and Quasi-identifiers that need to be protected in $O_D$. These constraints form the privacy policy.

3. The data collector anonymizes the ontology $O_D$ to produce $O_D^A$ and shares $O_D^A$ with the data user.

4. The data user defines the requirements (that make up the analysis policy) based on $O_D^A$ and send them back to the data collector for verification.

5. The data collector verifies the analysis policy with respect to the privacy policy to see if the two constraints conflict. If they conflict, the data collector identifies the rules in the analysis policy that cause the conflict. She can then modify $O_D^A$, or report the conflicts to the data user, or both. Return to step 3.

6. If they do not conflict, the data collector transforms the $O_D$, analysis policy and privacy policy into data anonymization rules, and anonymizes $D$ appropriately to produce $D^A$. She sends $D^A$ to the data user.

## 6.1 Building Data Ontology

As previously discussed in section 5, any given dataset can be transformed into an ontology representing the dataset. The method is as follows.

We can view each record $r$ in dataset $D$ as a tuple of information linking record $r$ with *Identifier* $i_d$, a set of $k$ *Quasi-Identifiers* $\{q_1^d, ..., q_k^d\}$, and a set of $j$ *Attributes* $\{a_1^d, ..., a_j^d\}$ Each of these $QIs$ and *Attributes* is a datatype belonging to some set. For example, the $QI$ age is an integer, the $QI$ date of birth is a tuple of value day-month-year, and the *Attribute* diagnosed illness is a member of an enumerated type naming elements of the set of all known diseases. Figure 3 shows this for a small dataset containing only *Identifiers*, age and gender.

The OWL ontology has built-in support for representing many of the common datatypes such as boolean, date, int, double, day, month, year, string, and time. For all $QIs$ and *Attributes* fields that fit in one of these basic datatypes, building the ontology to represent them requires only creating a datatype property linking the record $r$ to the appropriate value for the datatype.

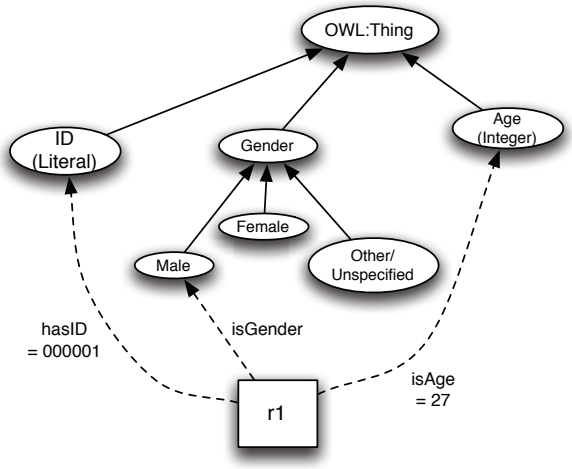


**Figure 3: A sample Ontology of a dataset containing ID, Gender and Age, where record 1 (r1) has ID 000001, male and is 27 years old**

OWL has support for creating collection to represent more complex enumerated type like illness, blood type or gender. Building such a list may be overwhelming for some fields like illness or medicine, because the list of possible items is extremely large. Fortunately, existing ontologies (such as ICD-9-CM) categorize most of this information. If no such ontology exists, one can be constructed simply by building an enumerated set containing all distinct value for the field in the dataset. The disadvantage to this approach is the lack of logic and classification information, which help in designing more complex policies and reasoning about them. For example, if one of the $QI$ fields is a city name, then adding object property relationship between cities such as distance, direction, and geographical hierarchy ($City \subseteq County \subseteq State \ldots$) allow the tracking of travel among locations.

## 6.2 Defining Privacy Policy

Equation 1 in section 5.1 is an example of a privacy policy that can be expressed in an ontology model. In this section we show how to construct a variety of privacy policies using threat modeling. In computer security, threat models describe a set of possible attack on a system. The models can take the perspective of the resources being attacked, the attackers, or the system being attacked. Once the possible attacks are identified, the designers and implementors can assess the probability of the attack, the damage the attack would cause, and approaches to eliminate or minimize these risks.

### *6.2.1 Resource-Based Modeling*

A resource-based approach is the most directly applicable method to our problem domain, because the focus of resource-based modeling is to identify the valuable resources in the system and the ways they can be accessed. Given an ontology that describes a dataset, the data collector can

identify *Attributes* or *QIs* that are considered sensitive. For example, in a disease classification ontology, the data collector may want to protect patients who have been diagnosed with severe complications from diabetes. Rendering these classes of diabetes as generic diabetes type 1 or 2 is not sufficient because these severe complications have other effects such as blindness, skin infection, fungal infection, and limb amputation. If a patient is diagnosed with generic diabetes with complications, and later diagnosed with a skin infection, then the patient's complications are clear. Therefore, in resource-based threat modeling, the data collector must identify not only fields that need protection, but also other related information from which protected information can be inferred. The analysis must take into account that some relationships will already be public knowledge, like diabetes leading to blindness, skin infections, and other well-known complications.

### 6.2.2 Attacker-Based Modeling

An attacker-based threat modeling approach focuses on examining the mind of the attacker to figure out how she might attack. For data anonymization, this type of modeling requires the information that the attacker wants to obtain. Given a dataset with medical diagnosis records, an attacker may want to retrieve individually identifiable information about all the records, or she may be more interested in a particular subset of the records. For example, the attacker may want to identify some individuals who possess easily verifiable illnesses like broken arms or legs. To defend against this particular threat, the data collector must design a privacy policy that thwarts this attack by hiding more information (age, gender, location of hospital) about patients who have visually verifiable illnesses than about those who do not.

The process of creating the threat model based on an attacker's view helps the data collector discover what external information attackers may need to achieve their goals. That, in turn, can inform additional attacks to augment the threat list and policy creation process in the system-based modeling in Section 6.2.3

### 6.2.3 System based Modeling

System-based threat modeling starts with the design of the system, and looks for possible attacks against each element of the system. In data anonymization, the system is a collection of information. Therefore, this type of modeling is analogous to stepping through the ontology model and discovering how information of the different types can be inferred from other data in the ontology or from external information. This method works with the threat list creation in section 6.2.2 to form a better privacy policy.

## 6.3 Anonymizing and Sharing the Ontology

After the ontology is developed and the privacy policy defined, the ontology must be shared with the data user. But the ontology contains all the information about the dataset, so the data collector must first anonymize the ontology. To do so, the data collector must first remove all individually identifiable information from the ontology. The construction of the ontology partially anonymized it by transforming all *Identifier* fields into a unique randomly generated *ID* field, so the data collector no longer has to worry about *Identifier*. The relationship between the *ID* and various *Attributes* and

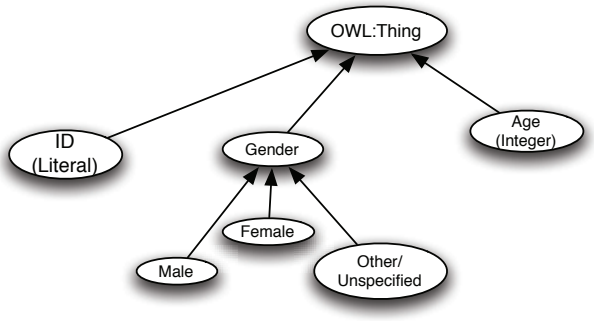


**Figure 4: An anonymized version of the ontology in Figure 3. In Figure 3, we know there is a 27 years old male in the dataset. This figure shows nothing except the type of data we may expect from the dataset (gender, containing male, female or other/unspecified; and Age, containing an integer).**

*QIs* is a concern because of possible inferences.

Fortunately, the design described in section 6.1 simplifies anonymization. If we examine the construction, we realize that for each record $r_i \in D = \{r_1, ..., r_n\}$ (where $n$ is the size of dataset $D$), all the relationships between the fields in each record are captured by the object and data properties linking entity $r_i$ to classes and entities representing each field in the ontology. Therefore, by removing all entities $r_i$ and all of their properties, we remove all possible ways of identifying individuals in the dataset. Figure 4 shows an example of the ontology in Figure 3, but anonymized.

If anonymizing an ontology like the one shown in Figure 4 produces too little information for the data user, and the data collector deems it reasonable to reveal more information about the dataset, the data collector may augment the ontology with more information. For example, instead of revealing only that a field in the dataset represents age for each record, the data collector may choose to reveal all ages in the dataset. Furthermore, the data collector can also annotate the ontology with the number of records in the dataset that has each of those ages. Figure 5 shows an example of an ontology augmented with more information, yet that may be reasonably anonymized.

Removing all the object and datatype relationships from record entities effectively turns the ontology into sets of tokens for each field, effectively making the dataset almost unusable. However, *the ontology does not replace the dataset* The ontology and the privacy policy, taken together, are simply a way of describing the data collector's understanding of the dataset. It serves as a medium to inform the data user about what they may expect the dataset to contain, and helps the data user to decide what they may want (and can get) from the dataset.

## 6.4 Defining Analysis Policy

Given an ontology, the data user can inspect it and know exactly what type of data the data collector will give. This includes the types of the fields, and—if the ontology is augmented with extra information like the one in Figure 5—what values each field in the dataset may contain. In addition, the privacy policy tells the data user exactly what

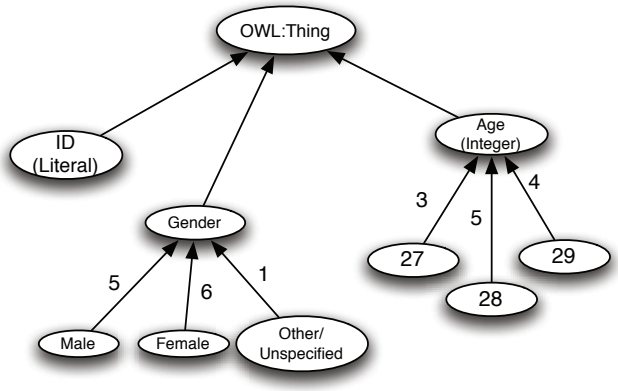


**Figure 5: The same anonymized ontology as in Figure 4, augmented with more information. This figure shows there are 12 individual in this dataset, 3 of whom are 27 years old, 5 of whom are 28 years old and 4 of whom are 29. Moreover, 5 are male, 6 are female, and 1 is unspecified.**

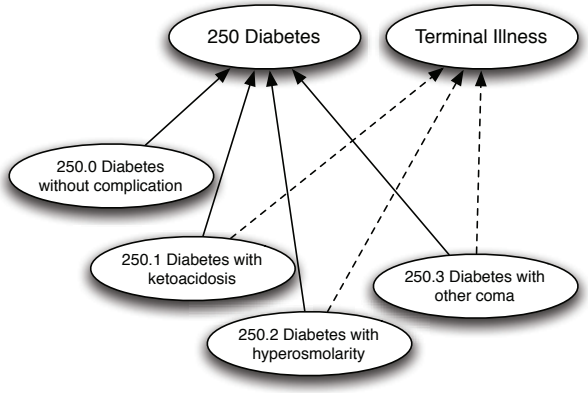


**Figure 6: Diabetes ontology showing that not all diabetes are classified as $TerminalIllness$. In this ontology, only 250.0 diabetes without complication is classified as $NonTerminalIllness$, and hence is restricted by Equation 7. However, the privacy policy in Equation 8 dictates that all *diabetes*, including 250.0 must have age generalized, causing a conflict between both rules.**

they cannot get from the dataset. For example, Equation 1 shows that the data user cannot get both gender and age from the records with an illness classified as both *Visible* and *Terminal*.

Knowing the representation of all data fields and the restrictions, the data user can make decisions and analyze trade-offs, or decide that this dataset is not suitable for the data user's need. For example, if the data user needs information about all records that have some terminal illness, then the data user needs to decide whether age or gender is more important, because Equation 1 requires that at least one be anonymized. If the data user chooses to require that age be preserved (by defining a $PreserveAge$ analysis policy such as in Equation 5), then for those records, all gender values will be suppressed.

However, consider Equation 1 more carefully. While gender is either revealed or suppressed, age is simply generalized, and the generalized groups are still reasonably useable. Given Figure 2, the data user may choose to preserve gender instead for that group of records and preserve age for the rest of the dataset by implementing the following analysis policy instead of the policy in Equation 5:

**PreserveGenderTerminal**

$$\forall x \in ID | TerminalIllness(x) \rightarrow TransformGender(x) = Gender(x) \quad (6)$$

**PreserveAgeNonTerminal**

$$\forall x \in ID | \overline{TerminalIllness(x)} \rightarrow TransformAge(x) = Age(x) \quad (7)$$

## 6.5 Resolving Policy Conflict

After both the privacy policy and analysis policy have been defined, some of their constraints may conflict. For example, if the data collector uses the privacy policy defined in Equations 1, 3, and 4, and combines it with privacy policy 2:

**Privacy Policy 2**

$$\forall x \in ID | Illness(x) \in Diabetes \rightarrow x \in GeneralizeAge \quad (8)$$

where *Diabetes* is the name of the class in ICD-9-CM Ontology section 250, representing all forms of diagnosis of diabetes (including all complications). This policy simply states that if an individual is diagnosed with some form of diabetes, then her age must be generalized. Furthermore, the data collector defines only some subclass of *Diabetes* to be $Terminal$, specifically section 250.1, 250.2 and 250.3. On the other hand, 250.0, diabetes without complication, is not classified as $Terminal$. Figure 6 shows the diabetes sub-ontology.

In this case, privacy policy 2 conflicts with the analysis policy in Equations 6 and 7 because *diabetes* section 250.0 is not terminal, thus Equation 7 requires that the age be preserved. But privacy policy 2 requires that all form of diabetes patients' age be generalized. This can happen if, for example, privacy policy 2 is not revealed to the data user because the data collector deems this policy to be internal or to reveals too much about the dataset, or if while designing the analysis policy, the data user forgot to verify those constraints against this rule.

Many OWL Ontology reasoners are available, and these kind of conflicts can easily be discovered without requiring human intervention. In this case, the reasoner will flag Equations 7 and 8 as violating Equation 2 because some records are member of both class, but Equation 2 requires that both classes be disjoint.

In such a case, the data collector can either ignore Equation 7 by altering the rule herself, or inform the data user that the policy results in conflict, and identify the reason for the conflict and which rule or rules need modification. This is part of the negotiation to achieve the balance that

satisfies both parties.

### 6.6 Performing and Verifying Anonymization

Once all conflicts are resolved, the data collector must perform the anonymization on the underlying dataset to satisfy the rule agreed upon using this ontology framework. To do that, data collector first classifies all the data records $r \in D$ in the ontology, as follows. For all $r$ that are members of some *Altered* rule, that is, that the privacy policy requires to be altered in some way, the data collector preprocesses those records' fields to satisfy the required rule. Using the example we have been following in the previous sections, all $x$ who are terminally ill and have visible illnesses must have their age field generalized as indicated in Figure 2. Therefore, the data collector generalizes the age of all record $r$ whose $ID$ is $x$.

After classification, the preprocessing of the required alter rules is performed. Then the data collector applies to the dataset any of the previously discussed methods to anonymize the dataset[5]. Upon completion, the data collector can take the anonymized dataset, and rebuild the ontology to classify the anonymized dataset, and identify which records violate the *Preserved* rules. This may happen because most existing algorithms do not yet support fine-grain control of anonymization to restrict some fields from being altered to meet the algorithm constraints and metrics. In such cases, the data collector can fix those violations manually by deanonymizing just some of those fields. While this may break the guarantee of the anonymization algorithm used, it does not break the privacy policy defined by the data collector, and therefore is likely to be acceptable (assuming the privacy policy rules are chosen well).

On the other hand, data collector does not have to deanonymize those violation, instead, it may choose to inform the data user that some of the fields are altered despite the preserve rule requested. Furthermore, data collector can tell data user exactly which records, or give some general statistical information to help in that regard. Again, the choice depends on the balance constraint requires by both parties.

## 7. EXPERIMENTS

As part of this project, we created an ontology model for two datasets we obtained in collaboration with a number of hospitals and agencies. Unfortunately, due to contractual restrictions, we are unable to publish the exact ontology model of the datasets. The examples given in this paper are similar to the model we constructed as part of the case study demonstrating that this framework is feasible and effective.

The first dataset is a collection from a diabetes patients management system. Hospitals collect and use this dataset to keep track of the condition of diabetes patients with varying degrees of complications. This dataset contains over 9000 patients and records over 40 attributes, among them patiente ID, birthdate, gender, weight, height, hospital or clinic service location, education, occupation, and alcohol and smoking history. The dataset also contains all diagnoses and health information about each of the patients of the diabetes center, including appointment information (totalling over 45,000 appointments) and information about medicines prescribed.

[5]See Section 2.

The second dataset is a collection of 50000 randomly selected records of health insurance claims over the period of 18 months. It contains over 20 attributes for each claim, including gender, age, occupation, date and time that the illness was diagnosed, and amount paid for medications.

Both of these datasets provide a rich variety of attributes that we used to create and test the framework, and provide us with realistic scenarios to create more extended case studies.

## 8. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a framework that allows for formal, automatic communication between a data collector and a data user. It enables both parties to negotiate and arrive at a good balance between protecting the privacy and secrecy of the dataset for the data collector and the utility of the dataset for the data user. We showed how the OWL ontology can be used to model the dataset and used to define both a privacy and an analysis Policy. In addition, we showed how, with careful construction of the ontology and policies, we can automate the detection of conflict between privacy and utility requirements.

We believe that this is a new aspect of the data anonymization problem that has not been considered much by the research community. Our work is orthogonal to the work in anonymity contributed by the research community so far, and complements it to simplify the process of sharing data between organizations that collect information already on a daily basis (such as businesses, hospitals, and social services organizations) and the community that needs the data for research, analysis and development.

This project gives rise to several possible direction for future research. First, the framework can be extended to adapt the ontology model for serial data release (also known as "data republication", in which some part of the same set of data is updated or released in other form at a later date.) In addition, work in section 6.6 can be made more formal. It requires a deeper analysis of methods to link the properties and constraints of the underlying data anonymization algorithm proposed by other researchers to this model.

Lastly, a more detailed and comprehensive case study will enable us to analyze the cost of each step of this framework with realistic dataset and usage requirements, because in a real application, the list of constraints and requirements in the privacy policy and analysis policy could be large, and the efficiency of the automatable conflict detection in this framework depends highly on the complexity of these rulesets.

We believe that further research in this area will better protect the privacy of data in existing datasets, while making those datasets and more widely usable for research in economics, social studies, science, and other areas.

## 9. REFERENCES


[1] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu. Achieving anonymity via clustering. In *PODS '06: Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 153–162, New York, NY, USA, 2006. ACM.

[2] G. T. E. C. Anas Abou El Kalam, Yves Deswarte. Personal data anonymization for security and privacy

in collaborative environments. In *Collaborative Technologies and Systems, 2005. Proceedings of the 2005 International Symposium on*, 2005.

[3] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering*, pages 217–228, Washington, DC, USA, 2005. IEEE Computer Society.

[4] A. J. Broder. Data mining, the internet, and privacy. In *Web Usage Analysis and User Profiling: International WEBKDD'99 Workshop San Diego, CA, USA, August 15, 1999*, volume 1836/2000, chapter p.56. Springer Berlin/Heidelberg, 1999.

[5] R. Crawford, M. Bishop, B. Bhumiratana, L. Clark, and K. Levitt. Sanitization models and their limitations. In *NSPW '06: Proceedings of the 2006 workshop on New security paradigms*, pages 41–56, New York, NY, USA, 2007. ACM.

[6] M. J. Cronin. e-privacy? 2000.

[7] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60, New York, NY, USA, 2005. ACM.

[8] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–286, New York, NY, USA, 2006. ACM.

[9] F. Li and S. Zhou. Challenging more updates: Toward anonymous re-publication of fully dynamic datasets. *arXiv.org:0806.4703v2*, 2008.

[10] K. R. Li Xiong. Towards application-oriented data anonymization. In *International Workshop on Practical Privacy-Preserving Data Mining*, 2008.

[11] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. $\ell$ -diversity: Privacy beyond $\kappa$ -anonymity. *Data Engineering, International Conference on*, 0:24, 2006.

[12] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. *Data Engineering, International Conference on*, 0:126–135, 2007.

[13] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 223–228, New York, NY, USA, 2004. ACM.

[14] S. Muthaiyah and L. Kerschberg. Virtual organization security policies: An ontology-based integration approach. *Information Systems Frontiers*, 9(5):505–514, 2007.

[15] S. V. N Li, T Li. t-closeness: Privacy beyond k-anonymity and l-diversity. *International Conference on Data Engineering (ICDE)*, 2007.

[16] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 665–676, New York, NY, USA, 2007. ACM.

[17] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

[18] L. Sweeney. k-anonymit: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, October 2002.

[19] T. M. Truta and B. Vinay. Privacy protection: p-sensitive k-anonymity property. *Data Engineering Workshops, 22nd International Conference on*, 0:94, 2006.

[20] X. Xiao and Y. Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 689–700, New York, NY, USA, 2007. ACM.