Relationships and Data Sanitization: A Study in Scarlet

Matt Bishop
Dept. of Computer Science
University of California, Davis
Davis, CA 95616-8562 USA
bishop@cs.ucdavis.edu

Justin Cummins
Dept. of Computer Science
University of California, Davis
Davis, CA 95616-8562 USA
cumminsj@cs.ucdavis.edu

Sean Peisert
University of California, Davis
and Berkeley Lab
Davis, CA 95616-8562 USA
peisert@cs.ucdavis.edu

Anhad Singh
Dept. of Computer Science
University of California, Davis
Davis, CA 95616-8562 USA
singh@cs.ucdavis.edu

Bhume Bhumiratana
Dept. of Computer Engr.
King Mongkut's University of
Technology Thonburi
Bangkok, Thailand
bhume@cpe.kmutt.ac.th

Deborah Agarwal
Computational Research Div.
Berkeley Lab
Berkeley, CA 94720 USA
daagarwal@lbl.gov

ABSTRACT

Research in data sanitization (including anonymization) emphasizes ways to prevent an adversary from desanitizing data. Most work focuses on using mathematical mappings to sanitize data. A few papers examine incorporation of privacy requirements, either in the guise of templates or prioritization. Essentially these approaches reduce the information that can be gleaned from a data set. In contrast, this paper considers both the need to "desanitize" and the need to support privacy. We consider conflicts between privacy requirements and the needs of analysts examining the redacted data. Our goal is to enable an informed decision about the effects of redacting, and failing to redact data. We begin with relationships among the data being examined, including relationships with a known data set and other, additional, external data. By capturing these relationships, desanitization techniques that exploit them can be identified, and the information that must be concealed in order to thwart them can be determined. Knowing that, a realistic assessment of whether the information and relationships are already widely known or available will enable the sanitizers to assess whether irreversible sanitization is possible, and if so, what to conceal to prevent desanitization.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issue

Keywords

Data anonymization, sanitization, privacy, ontology

1. INTRODUCTION

Sharing data is crucial to modern life. The hallmark of science is the ability to reproduce experimental results in order to validate both the experimental methodology and results [21, 53], so the data driving those experiments often must be shared. Medical research depends on sharing data; the National Institutes of Health stated that "[w]e believe that data sharing is essential for expedited translation of research results into knowledge, products, and procedures to improve human health" [3]. Indeed, the nation's security depends on sharing data. The Department of Homeland Security stated that "it is critical that each DHS component gives the highest priority to the sharing of potential terrorism, homeland security, law enforcement and related information" [6]. And two presidential directives [2, 4] emphasize the requirement that agencies share information.

Privacy is equally crucial to modern life. From the late 1890s to now, people and laws have moved to protect information.[1] that is typically not to be shared. At the turn of the century, privacy was described as "the right to be let alone" [92]. Currently, the focus of most efforts to protect individual privacy focus on "personally identifiable information" (PII). A key question is what constitutes PII. This varies among different domains of knowledge. For example, the U.S. Health Insurance Portability and Accountability Act (HIPAA) defines "protected health information" as "individually identifiable health information ... that is transmitted or maintained in any form or medium" [5]. The California law requiring notification in case of data theft, SB 1386, defines it as "an individual's first name or first initial and last name in combination with any one or more of ... (1) Social security number. (2) Driver's license number or California Identification Card. (3) Account number, credit or debit card number, in combination with any required security code, access code, or password that would permit access to an individual's financial account." The laws and regulations of states and the federal government require, in most cases, that PII be protected.

Corporate and organizational information systems normally contain business sensitive data The data may be proprietary, classified, sensitive, or simply embarrassing to release. This type of data may also be defined by law, by

[1]We treat information derived from the analysis of raw data to be another form of data that must be considered in this context.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

custom, or by the organization itself. Often, the sensitive aspects are embedded in or derivable from larger masses of data that the institution may need to release to third parties. An example is a set of network traces that the company believes contain attacks sent to a security analysis firm to be examined for those attacks. The institution will want to obscure such things as customer purchase orders and proprietary data in case the data leaks (or to protect itself against an insider attack originating in the security analysis firm).

Define "sensitive" data as data that is to be kept private (such as PII). It is important to realize that the inferences drawn from sensitive data may be incorrect. For example, suppose one searched the web for information on "heroin." A law enforcement agency, seeing the searches, might conclude one was looking for information on making heroin—when, in reality, the searcher might be a high school student preparing a report on the dangerous effects of the drug. The first inference is wrong, yet it could still be damaging to the reputation of the searcher. Thus, the sanitization process must constrain the ability of adversaries to infer information. The sanitizers may wish the adversary to be unable to draw specific inferences. Conversely, they may wish the adversary to draw *specific* inferences from the data as part of a manipulation or deception technique.

Nearly all cybersecurity researchers have been involved in discussions about how to obtain data, or how to share data, in a safe way. This is also true in the medical community. The dilemma of needing to release data yet conceal sensitive information within that data set has resulted in the notion of "sanitizing" the data before release. Sanitizing in this context means transforming the data in such a way that an adversary cannot determine the sensitive data.[2]

Much work has been done on how to sanitize data. Methods used in the past, with varying degrees of success, include prefix-preserving homomorphisms, substitution of random strings for data, and simple suppression of data. Thus, the academic literature contains many ideas and methods describing *how* to sanitize data—and, of course, many ideas and methods on how to determine the raw, unsanitized data given the sanitized data. Policy and privacy issues, and the specific data fields and other information that is to be kept private, are discussed in the medical, social science, and computer science fields. Remarkably, though, the only linkage between these two thrusts is how to apply the first to the second—that is, given a set of data to sanitize, how does one sanitize it? And often, the results are insufficient, as numerous desanitization papers have shown.

The current paradigm used in data sanitization assumes a closed world—only the data being sanitized is relevant, and data external to that set will not help reverse the sanitization. This assumption is fallacious. To see this, consider a case where the information to be suppressed includes gender and race but does not include medical condition. Many medical conditions are peculiar to gender (for example, pregnancy is exclusive to the female gender) or tend to occur in particular groups of people (for example, Tay-Sachs Disease is largely confined to Ashkenazi Jews). Combining external data with the sanitized data has proven an effective tool in desanitizing data, as several studies have shown.

Our proposed paradigm focuses on the observation that desanitization techniques operate on the basis of relationships among data attributes and data itself. Our goal is to develop a methodology to integrate these relationships into the sanitization of data, so that an adversary cannot use those relationships to reverse the sanitization. Thus, in our medical example above, an ontology would capture that the value "pregnant" in the "medical condition" field means that the value of the "gender" field must be "female." Thus, if gender must be suppressed, so must the value of "pregnant" in the medical condition field.

Thus, our paradigm uses an open world assumption. It also differs from the existing paradigm in approach. The existing paradigm asks whether the known data is sufficient to desanitize the sanitized information. This is reasonable because the set of data available is known. Given our open world assumption, that is not true; who knows what data is available to an ingenious searcher of the Web, or a toiler through obscure archives? Less poetically, the sanitizer may not know the extent or nature of *all* external information available to the adversary. Thus, it is impractical to find all specific data sets that will enable an adversary to desanitize data. But it *is* possible to identify specific (possibly hypothetical) relationships that would enable an adversary to undo the sanitization. Knowing what relationships would enable an adversary to determine the sensitive information enables the sanitizer to assess the risk of releasing the sanitized data by determining whether the relationship in fact exists and if so, how difficult the sanitizer believes it to be for the adversary to determine the relationship.

To emphasize the point of this work: we deal with *what* data is to be sanitized. We examine how to sanitize that data only when the method of sanitization is relevant to prevent desanitization. Further, we leave the decision of what "desanitization" means to those who author the privacy requirements. In particular, they may want to prevent certain inferences even when those inferences are incorrect. With this in mind, we first discuss our basic model.

2. BASIS FOR THE MODEL

Our model extends earlier work [14, 16, 17, 29]. Define a *collector* as one who gathers data $d \in D$, where D is the domain from which the data is collected. She wants to turn this data over to an *analyst* who will determine if a set of properties a_r hold for that particular data set. The function $a_r(d)$ holds if and only if the properties a_r hold over d; that is, the analyst is trying to determine whether $a_r(d)$ is true. In practice, the elements of a_r may embody approximations.

The collector (owner) requires that certain parts of the data set be confidential. The *privacy requirements* p_r hold over d when $p_r(d)$ is true. Define a *data sanitization mapping* to be a mapping $s : D \to D$ such that $p_r(s(d))$ holds; that is, after applying s to the data d, the privacy requirements are satisfied. The data thus removed is called "redacted" data. The original data set d is called the *raw data set*, and the data set $s(d)$ is the *sanitized data set*. For simplicity, we assume the collector does the sanitization, and thus often refer to the collector as the *sanitizer*.

An *adversary*'s goal, given $s(d)$, is to determine a set of

[2]A note on terminology. The terms "anonymization" and "sanitization", and "deanonymization" and "desanitization", are both used in the literature. One can argue that "anonymization" refers to information about a specific entity, and "sanitization" to more general data, for example the ability to draw undesired inferences. In this paper, we refer to "sanitization" in general, and "anonymization" where we mean using encryption to hide an identity.

data d' such that $p_r(d')$ does not hold. We refer to this process as "desanitization" and the data thus uncovered as "raw" or "unredacted" data. Note that the data uncovered may not match the data in the raw data set exactly; it may be enough to know someone's income to within $1,000 of the actual value. We assume this is captured in the privacy requirements, so we accept the imprecision of calling the inexact yet violative result "sensitive" or "unredacted" data.

In some cases, the analyst may not need to determine whether a set of properties $a_r(d)$ is true with probability 1; it may be sufficient to determine that the probability of $a_r(d)$ holding is greater than $1-\alpha$ for some threshold α. In other words, what matters is that $Pr[|a_r(d) - a_r(s(d'))| < \beta] > 1-\alpha$, where β defines "acceptably close." The key point is that the sanitization does not interfere with the analysis. In practice, this may not be possible, and handling this inability is a key theme of our research.

More formally, let f be the *adversary's desanitization mapping* such that $f : D \to D$ and f is known to (or determined by) the adversary. f may, or may not, be the inverse to s. Let $d' = s(d)$ be a sanitized datum, and let $\Delta : D \times D \to \mathbb{R}$ be a distance metric. Note that d and d' may not be numbers, hence the need for Δ.

For some value δ and some other value $\epsilon \in [0 \ldots 1]$, we want $Pr[\Delta(f(d'), d] < \delta] < \epsilon$. This means that the probability of the adversary's desanitized mapping producing a raw value that approximates the actual raw value to within some precision δ is acceptably small (that is, less than some threshold ϵ). For example, take d to be gender, and Δ to be 1 if the two genders differ and 0 if they do not. Then δ would be 1 (because we want the adversary to see the genders as different) and ϵ might be 0.01, meaning that the probability that an adversary can determine the correct gender is less than 0.01.

This model has several ramifications:

1. The adversary may affect the data being generated, for example by creating data of known form (called *markers*) that will be embedded in $s(d)$ when the collector releases it. This is similar in concept to a known or chosen plaintext attack in cryptanalysis.

2. The adversary has access to all of $s(d)$. Thus, she may be able to deduce redacted information from unredacted information in $s(d)$.

3. The adversary has access to other data, *external to* $s(d)$, that may enable her to deduce redacted information.

Figure 1 shows this model graphically. Note that both the adversary and the analyst in this model have access to the same data, $s(d)$. The collector's goal is to sanitize the data in a way that enables the analyst to determine the same information from the sanitized data that she would from the unsanitized data, while preventing the adversary from determining the sensitive data.

Note also that the sanitization and analysis may conflict. That is, it may be impossible to find a mapping s such that both $p_r(s(d))$ and $a_r(s(d))$ hold to the desired thresholds. In this case, if the data is to be released, either the privacy or the analysis requirements (or both) must be changed.

Briefly, our approach expresses privacy and analysis requirements using a constraint or policy language. We then compose these constraints, and if the resulting expressions

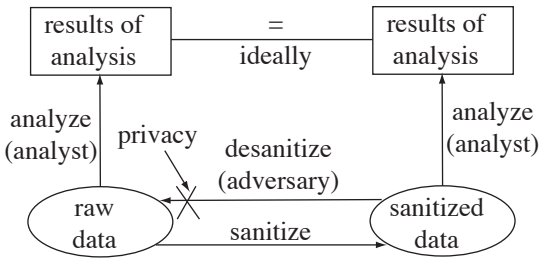

Figure 1: How data sanitization works. Without sanitization, the analyst analyzes the raw data to produce results (left part of figure). With sanitization, once the collector sanitizes data (bottom arrow), the analyst analyzes the sanitized data to produce results (right part of figure). Ideally, the results of the analysis are the same. The adversary tries to determine sensitive data (top arrow between bottom ovals); the goal of the privacy requirements is to prevent this.

conflict, the conflicts identify the specific privacy and analysis constraints that conflict. The conflicts must then be resolved, for example by people following policies or using their best judgement. Our work does not deal with how to resolve these conflicts.

We also focus on *what* data is to be sanitized; in particular, we examine the relationships that are not apparent from the syntax and semantics of the data set. While *how* the data is sanitized is a critical element of protecting privacy, we confine our consideration of it to the next section, and especially how it affects the concealment of the relationships of interest.

We now review earlier work on data sanitization to provide a basis and context for our approach.

3. BACKGROUND

Work on data sanitization takes one of two approaches: *perturbation*, in which incorrect data values replace correct ones, in such a way that the analysis of the perturbed data produces the same results as the original data;[3] and *generalization*, in which values are replaced by ranges that include the correct values.[4] In this section, we first discuss methods of sanitization, and then methods of desanitization, by surveying the literature.

In what follows, the term "quasi-identifier" (QID) means information that is sufficient to identify an entity uniquely, or that can be combined with external information available to the adversary to identify an individual uniquely. For example, a Social Security Number is a quasi-identifier because, while it is not itself an identifier, it can be used to identify an individual uniquely. Much sanitization is an attempt to conceal quasi-identifiers, and adversaries try to uncover quasi-identifiers as well as identifiers.

[3]For example, if the domain is a set of numbers such as salaries, the perturbations must preserve the statistical moments of interest to the analyst.

[4]For example, suppression of information is a form of generalization, because the values can be any legal values in the domain of the data—including the correct values.

The general problem of computational disclosure control is an inferencing game that parameterizes problem spaces to unify the notions of inference problems [29]. It raises significant questions about current approaches, including the notions of a closed world (which several deanonymization methods have demonstrated is fallacious; see Section 3.2) and of a uniform analysis metric that holds an adversary desires all sensitive data equally, and that the disclosure of *any* sensitive data is of equal cost to the collector.

Data sanitization can be formulated as a problem in information theory. For example, let L be a set of data, $W \subseteq L$ be the sensitive data in L, and let $|S| = n$. Let P be the set of data that will replace elements of S. The mapping $s : W \rightarrow P$ sanitizes L perfectly if, for any $w \in W$ and a given $p \in s(W)$, $Prob[p = s(w)] = 1/n$. Taking the analysis further, one can draw parallels between the sanitization problem and problems in anonymity in groups. However, this approach is tangential to the paradigm put forth in this paper.

3.1 Sanitization Methods

Data sanitization is context sensitive which means that the data can be generalized or perturbed in several different ways, the choice of which depends on the context [14]. Two primary techniques are used. *Generalization* replaces a value with a range of possible values that the attribute may assume. For example, replacing a birth date with the birth year replaces the actual value (a date) with a range of values (365 or 366 possible dates). Deletion is a form of generalization, because then the attribute could be any legal value. *Perturbation* retains a single value, but transforms it in some way. For example, adding a random value to a datum perturbs it. When this is done, the sanitizer must be sure that the results of the analysis of the perturbed data match those of the raw data.

K-anonymity [85], a widely-used method of sanitizing data, generalizes information so that the generalization is valid for at least k entities. Several variants of k-anonymity have been proposed to overcome specific problems. For example, one study extended the model to limit the confidence of inferring a sensitive value [93]. Another [71] proposed a technique to achieve k-anonymity not just in one dataset, but over many datasets, by applying k-anonymity to the record owner level rather than the record level over the join of all the datasets. l-diversity is a variant of k-anonymity in which every group of QIDs must have some number of distinct values for the sensitive attribute [60]. Other variants abound [54, 56, 59, 70, 96], as do other generalization techniques [10, 12, 41, 47, 91].

Perturbation techniques change the data to achieve anonymity. One such method of achieving this is by masking, which if done appropriately can enable analysis to achieve results similar to those of the analysis on the raw data. An example is adding noise [44, 50, 66]. General additive data perturbation is a generalization of that technique [65]. Other masking approaches are also useful [8, 51].

Identifying fields to sanitize is different than determining how to sanitize them. Work in this area has focused on data mining. For example, Bayesian and conditional random field based classifiers have been used to identify attributes in unstructured medical data [36]. Other machine learning techniques have been used with some success [86], including streaming data [25] and trying to approximate the background knowledge of the adversary [55].

Frameworks for sanitizing network data have also been developed [52, 74, 77, 78, 95]. Pang, Allman, Paxson, and Lee [73] examine the impact of anonymization with the needs of the analysis, but use an entirely different methodology than we propose. Wang, Fung, and Yu [89, 90] discuss privacy templates, which are similar to our notion of privacy constraints, but they do not focus on the needs of the users of the anonymized data. Sun, Wang, and Li [83] discuss balancing the priorities of anonymization, thereby questioning the uniform analysis metric, but do not consider the analysis requirements.

3.2 Undoing the Sanitization

An analysis of the literature in desanitization reveals four properties that adversaries depend on.

First, external information, when correlated with the sanitized data, enables the adversary to determine the sensitive data. This is by far the most widely-publicized technique of desanitization. It gained prominence in the AOL release of data in 2006,[5] in which New York Times reporters were able to correlate contents of search queries to public records, and from that determine the identity of the anonymized querier with pseudonym 4417749 [11]. An interesting aspect of this result was the analysis of the search queries. They were often about medical conditions such as hand tremors, bipolar disorders, and nicotine effects on the body—none of which were true of the user; i an interview, she said that she often helped friends research their medical questions and conditions on line. This is an example of the need to prevent unwanted inferences from being drawn; these inferences could result in correct or incorrect deductions.

More recently, Narayanan and Shmatikov attacked the Netflix Prize Dataset containing data for anonymized users. The data associated with each user is a set of pairs of movie titles and ratings, and of ratings and rating times. The researchers correlated these pairs with data from the Internet Movie Database[6] (IMDB), a public database in which viewers can rate movies. They were able to match Netflix data with the IMDB data, and associate IMDB identities with Netflix sanitized identities. Netflix claims that the associations are invalid because they perturbed the data [80]; however, that correlations could be made illustrates how effective the use of external data can be. Other papers discuss techniques for performing these correlations [24, 34, 35, 46].

Second, patterns in raw data often reflect similar patterns in sanitized data, so if the adversary knows those patterns, she can infer the sensitive data. Desanitization social networks uses this type of relationship. For example, Narayanan and Shmatikov [68] identify three types of adversaries who may exploit this. Government agencies engaged in widespread surveillance have access to a large set of networks from which they can reap information. Marketing, especially that involving behavioral targeting of advertisements, may obtain a (sanitized) topological map of the network for commercial purposes; they then need only match the topology to the entities in the network, rather than discover the topology. Finally, individuals targeted for identification (by investigators, stalkers, employers or potential employers, or others) are at risk when the adversary has

[5]AOL removed the data after 4 days, but the data is still on-line at http://www.aolstalker.com

[6]http://www.imdb.com

detailed contextual information about them, such as some of their social relationships, memberships in other networks, attributes captured in the social network being studied, and so forth. Narayanan and Shmatikov examine partial overlaps among the sanitized target subnet and attacker's auxiliary networks to show how desanitization can occur.

Third, relationships among the data or data fields in the sanitized data enable the adversary to infer sensitive data. This includes dependencies in the raw data that are not obscured in the sanitized data. A simple example is the sanitization of a network topology to conceal the gateway. As all traffic into and out of that network must pass through the gateway, sanitizing network addresses and ports corresponding to protocols run on the gateway will not obscure the magnitude of the traffic entering or leaving one particular host. This technique was used to recover the network topology and undo some of the sanitized addresses corresponding to externally visible hosts [28]. Other examples of the use of this property also focused on network data [23,27].

An interesting observation is the applicability of the cascading effect to some anonymization techniques. The data that Coull and his colleagues examined [28] used Crypto-PAn [18,33], a prefix-preserving method for anonymizing IP addresses. The preservation of prefixes is particularly useful for network data because it preserves the association of hosts with a subnet in the sanitized data. The structure of Crypto-PAn makes an anonymized address depend on all the bits of previous unanonymized data. Thus, deanonymizing one address cascades throughout the subsequent anonymized addresses. This illustrates the effects of dependencies within the sanitized data in a rather dramatic way.

The fourth property follows from this. An examination of raw data often reveals dependencies not accounted for in the sanitization, and that can be exploited to desanitize the data. For example, Panchenko and Pimenides [72] combine application-layer information with network-layer information to speed deanonymization of the traffic. Basically, they look for profiles of individuals at the upper layer, and once those are found, use them to filter the input for the attack on the network layer by eliminating combinations that do not fit the profiles. Their paper demonstrates this by extending the predecessor attack [94] to use this extra information. A second example, that of SPIRAL [49], examines the relationship of location information with sanitization.

3.3 Summary

None of the previous work offers a formal representation of the relationships between the various attributes. As shown above, these relationships often leads to points of opportunities for attackers. For example, the Netflix dataset providers concealed the (customer name, rating), (customer name, rating time), and (customer name, rating date) pairs. But they did not conceal relationships between the rating, rating time, and rating date fields. As public, external sources of these relationships also contained customer names, attackers could use this information to desanitize the released data—and the attackers did exactly that.

4. APPROACH

Our approach focuses on the question of *relationships*, specifically those that can be used to desanitize sensitive data. To begin, though, we examine the basic problem of data sanitization, with a simplifying assumption that we later relax.

That assumption is that the semantics of the domain involved are all contained within the data. In other words, we ignore relationships among the data and the use of external data. We then use a threat model to define the privacy constraints, and express those constraints in a language that supports reasoning. Call this set of constraints $p_r = p_{r,1} \wedge \ldots \wedge p_{r,m}$, as in Section 2. Similarly, we determine what information the analysts want to derive from the data, and express those in the same language. These are $a_r = a_{r,1} \wedge \ldots \wedge a_{r,n}$, again as above.

Thus, the sanitized data must satisfy

$$C_r = p_r \wedge a_r = p_{r,1} \wedge \ldots \wedge p_{r,m} \wedge a_{r,1} \wedge \ldots \wedge a_{r,n}$$

which is possible only if the $p_{r,i}$ (for $1 \leq i \leq m$) and $a_{r,i}$ (for $1 \leq i \leq n$) are consistent. So we examine these predicates for consistency. If they are consistent, then the privacy constraints can be enforced without inhibiting the desired analysis. If they are not consistent, then the inconsistency must be resolved. Resolution requires altering the conflicting constraints, or modifying the threat model or analysis goals. Resolvinge the conflicts is the domain of policy analysis, and is outside the scope of this project.

But the conflicts may not be obvious, as explained in the introduction. In particular, there *are* relationships internal to the data set that affect the effectiveness of sanitization, and an adversary *will* have access to external data that may illuminate these relationships, or establish new ones. These must be factored into the analysis for conflicts.

As a simple example, studies have established that the ZIP code, birthday, and gender *and nothing more* uniquely identify approximately 63% [38] or 87% [84] of the people in the United States. Thus, when sanitizing names from records that contain name, address, gender, and birthday, one needs also to sanitize some of the above data. For example, Golle shows that, given the 5-digit ZIP code, gender, and year of birth, only 0.2% of the U.S. population is uniquely identifiable, and adding in the month of birth raises that to 4.2% [38].

As a more complex example, revisit the Netflix study described in Section 3.2. Figure 2 shows (a simplified version of) how Narayanan and Shmatikov desanitized much of the Netflix data. They reaped data from the IMDB database corresponding to the Netflix data fields (as shown in the bottom oval of the figure), *including* the sanitized user field. They then compared the two sets of data, and matched records with correponding values for the data attributes (allowing for some variance). In the figure, the three non-sensitive fields of the IMDB record for John match those of the Netflix data (with some variance allowed for the date field), and do not match those of Mary's record.

We can use an ontology to capture these relationships. In the context of the problem of data sanitization, an ontology captures relationships among concepts and data in a form that one can reason about [15]. We use the ontology to augment the reasoning engine that is composing the privacy and analysis requirements. The additional information from the ontology will enable the reasoner to detect conflicts not apparent from the requirements alone.

Bhumiratana and Bishop [15] give an example using a medical ontology. Consider the ICD-9-CM classification [69], which is used to document diagnoses in most health insurance claims in the United States. Section 250 covers the

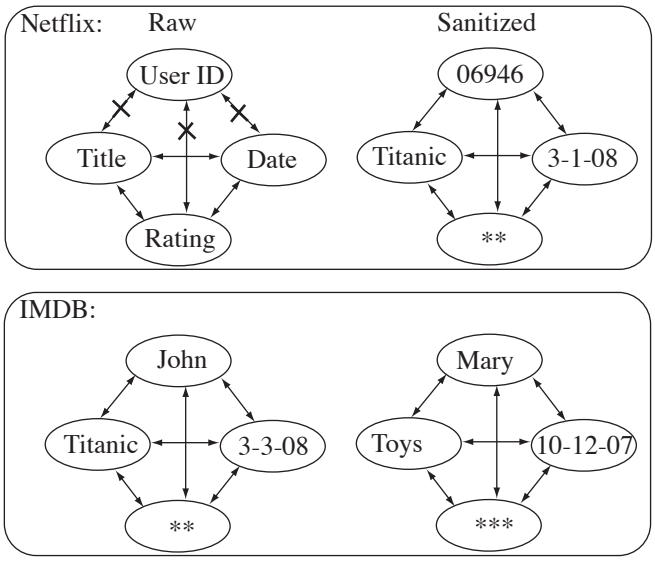

Figure 2: Relationships among the Netflix data fields. The data labeled "Sanitized" is released, breaking the association between the name and each of the title, rating, and date. Comparing this information to the IMDB data in the lower oval, the movie title "Titanic" and the rating "" is an exact match, and the rating date "3-1-08" is very close to the IMDB rating date "3-3-08". Thus, user 06946 is John.**

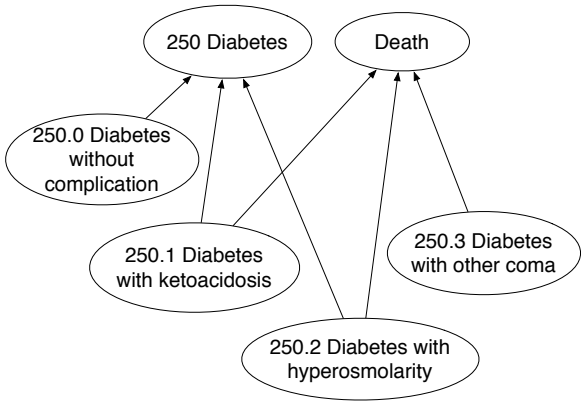

Figure 3: From [15]. This diabetes ontology shows that not all diabetes would be classified as leading to death if untreated.

class of all forms of diabetes including all complications (see Figure 3). One part of the privacy policy $p_{r,i}$ says that if an individual is diagnosed with diabetes, her age must be generalized. One part of the analysis policy $a_{r,j}$ says that if an individual has an illness that, if untreated, could lead to death, then age is not to be generalized. This is a conflict because as 250.0 diabetes without complication is classified as a non-terminal illness, $a_{r,j}$ requires that age not be generalized, but $p_{r,i}$ requires that it be generalized. Were no ontology used, the relationship between diabetes and a non-terminal illness would be unknown and this conflict not uncovered.

We now discuss the expression of requirements, and issues involving the expression of ontologies, in more detail.

4.1 Threat Model and Requirements

A threat model and complementary privacy policy are crucial for determining the types of threats to guard against. A privacy policy is meant to describe aspects of the data which should not be revealed. The threat model goes hand-in-hand with developing the privacy policy by informing what types of entities may attempt to reveal protected information and what resources are at their disposal. For example, an ISP may want to share network data but prevent rival organizations from identifying sensitive infrastructure details. Also, the ISP may want to prevent the attribution of network records to specific customers. Each concern might require different sanitizing processes. Additionally, the sanitizing process required for protecting customers may differ depending on the capabilities of the adversary attempting the attribution (e.g. a reporter versus a nation-state).

Privacy policies typically exist for each data set but with vastly varying levels of specificity and completeness. Using a well-defined threat model, the privacy policy can be made complete with respect to all threats described therein. The initial specificity of a privacy policy may espouse high-level principles, but eventually describe an exact description of methods which should be applied to each sensitive field.

A large concern with data sanitization is that a dataset should not be overly sanitized to the point of providing no useful information to the analysts. While ensuring privacy, this undermines the purpose of sharing the data. In order to satisfy both constraints of usefulness and privacy, an analysis policy is needed to describe what must be preserved, and to what degree, after sanitization. This analysis policy describes what data will be used, the level of precision required, and any invariants from the original data that must not be destroyed. Note, the analysis policy need not describe how the data is to be used or for what purpose.

The privacy policy and analysis policy are opposing, representing the needs of the data stakeholders and analysts, respectively. Both policies must eventually be described (or translated) to the detail of specific fields. It may be the case that there is no contention between the privacy and analysis policies, ensuring agreeable sanitization for both parties. However, contention prompts a compromise on the part of the stakeholders or analysts to enable amicable data sharing. Existing privacy and analysis policies allow adjustments until a technical agreement can be met. In fact, the resulting privacy policy conveys the risks assumed by the stakeholder in the sharing scheme. External controls, such as contracts or memoranda of understanding (MOUs), can be used to augment such compromises [48].

Our approach uses an ontology structure to describe the data available and known relationships pertaining to that data (internally and externally). This ontology structure lends itself to a detailed construction of the privacy and analysis policies. An algorithm can determine if conflicts exist between them. Encoding the known relationships among data fields is necessary to combat the inference problem, such as a combination of insensitive fields whose composi-

tion is sensitive. It is important to fully map all known inferences to prevent exposure of sensitive data that was intended to be sanitized. The ontology structure is ideal for describing such broad relationships. With inferences and sensitive fields cataloged, ontology reasoners may generate further inferences, completing the ontology under the specified rules. Other approaches have not employed this method of finding all known inferences.

Previous work in sanitization often views a system as a closed world—ignoring outside knowledge that may be available to attackers [7, 11]. A process performed in such a way will fail in practice because outside information is not considered. Additionally, portions of the dataset, similar datasets, and datasets representing some of the same entities, if published or shared previously, may be used to attack a sanitized dataset [67, 68]. Our approach explicitly considers this information, which can be readily encoded into the ontology.

4.2 Ontologies

Research involving ontologies is ongoing, and often applied to biological research, the semantic web, and knowledge representation. Ontologies are essentially directed graphs with concepts as vertices and the relationships between them represented as edges. There are many ontology languages including OWL-Lite, OWL DL, OWL Full [30], OWL 2 [39, 63], RDF Schema [22], SADL [61], DAML+OIL [43], and CASL [9, 62]. These vary in the type of logical statements they support and intended uses. Languages also often have restricted sublanguages that provide better guarantees of computational complexity or completeness while giving up some amount of expressivity. Each language also supports the creation of axioms, or rules. Some language extensions specifically provide support for rules, such as SWRL [42] and DL-Safe [64]. SWRL is a proposed W3C standard and combines the rule logic of OWL DL axioms and RuleML [20] rules. SWRL on the whole is neither decidable nor sound. However, each tool supports a different subset of the proposed standard.

Within the areas of semantic web and general knowledge representation, OWL is the *de facto* standard. It followed several other ontology languages, including RDF Schema, OIL, and DAML+OIL. OWL was standardized by the W3C in 2004, and based on community feedback, was updated in 2009 to OWL 2 (which is backwards-compatible with OWL). Three sublanguages (OWL-Lite, OWL DL, and OWL Full) aim to enable OWL to provide tradeoffs between expressivity and computational completeness or decidability guarantees. OWL Full is not well supported in tools, and so is not used much. OWL-Lite, a syntactic subset of OWL, offers few advantages to OWL DL, the more common of the sublanguages. Similarly, OWL 2 offers profiles instead of sublanguages. Commonly used profiles are EL, which has polynomial time reasoning complexity; QL, which is designed to ease access to data stored in databases; and RL, which is a rule subset of OWL 2.

Unfortunately, only some of the many ontology languages available have supporting current, production-quality tools. Sometimes the tools do not support some language features and offer extensions to the language simultaneously. Other times a single out-of-date or unsupported tool is the only implementation of a language. As an example, only the Hets tool [57] supports the CASL language. Frame-based ontology languages are not appropriate for this modeling because they primarily focus on recognition of objects or classes rather than relationships, which are core to our work. Several ontology languages such as DOGMA [82], Gellish [88], and IDEF5 [1] are highly domain specific; others such as CycL [76] use tools with unavailable source code, or that are proprietary. SADL has little documentation. Some projects such as RIF [19], KIF [37], and Common Logic [31] are strictly formats rather than ontology languages. These problems limit the selection of an ontology language.

For popular ontology languages, there are many ways to programmatically interface with a developed ontology in various programming languages. There are few ontological reasoners, tools which generate further inferences and check for consistency, that efficiently process expressive ontologies and have documentation and good support. Pellet [81] and FaCT++ [87] are two leading free and open-source reasoners that seem good candidates. The Jena Semantic Web Framework [26] may be useful; however, as a framework it is not as extensible. Reasoners such as SHER [32], DLog [58], RacerPro [75], SweetRules [13], and Bossam [45] appear to be no longer maintained or updated. Some novel reasoners such as DLog 2 [58] may prove useful in the future as they mature.

Logic programming (LP) is an alternative approach to ontology creation; however it is not alone optimally suited to this task. LP languages, such as Prolog, provide a logical foundation based on very different assumptions than description logic (DL) languages (and ontology languages) [40]. Even so, some reasoners, such as the original DLog, work by converting a subset of DL to Prolog. Ontologies are naturally suited to describing data and its relationships, the focus of our project. Additionally, ontology languages are currently used with very large datasets in the biological and semantic web realms.

Ontology languages involve considering assumptions that are implicit when using logic programming, primarily the open-world assumption and the unique name assumption. Under the open-world assumption, if a statement cannot be proved to be true using current knowledge, the statement cannot be concluded to be false. For example, suppose it was stated that a person Robin is female. The answer to the question "Is Robin male?" would be unknown unless male and female were declared disjoint. This is related to monotonicity, where giving the ontology additional information that does not directly contradict previous information. Explicitly enumerating all values of a given class in OWL eliminates the open world assumption. LP languages, on the other hand, use the closed-world assumption. The open-world assumption may not be critical for this project as fields in data sets are primarily numeric values, enumerable, or open-ended, where open-ended data is typically suppressed. Second, the unique name assumption prescribes that two entities with different names must be different entities. Ontology languages do not make this assumption. The consequence is that two entities with different names might be the same until they are made explicitly distinct.

Our approach models datasets, relationships between fields, and external information, as an ontology. Within a domain, the ontology should require little modification as the same fields (and the same relationships) are likely to be common in datasets. Thus, after a general dataset is modeled, a reasoner and a set of rules may provide further inferences

to augment the ontology to include current understanding. Due to the monotonic nature of ontology languages, future additions to the ontology will not invalidate prior conclusions. Since programmatic interfaces with common ontology languages are plentiful, a parser could be generated from the ontology to sanitize the raw data. This approach enables domain experts to easily contribute to an ontology; thus, fewer additions to each new dataset within a domain will be needed. Furthermore, the ontological construct of the privacy and analysis policies allow parties to understand how data may be sanitized in the context of data fields' relationships to each other.

4.3 Example

Our example is drawn from a hypothetical medical insurance company that wants to measure claims relative to service, age of the customer, and insurance status. The analysts are not to see the raw data, but work from sanitized data.

The privacy requirement is that dates be no more granular than a year. We show how to use two different constraint languages to express this. First, the Semantic Web Rule Language (SWRL), which is used with the Web Ontology Language (OWL), expresses the above constraint as follows:

$$\begin{aligned} R\_BY: \quad & Person(?x) \wedge SuppressBirthDate(?x) \wedge \\ & SuppressBirthMonth(?x) \wedge \\ & PreserveBirthYear(?x) \rightarrow BirthYear(?x) \end{aligned}$$

OWL uses an open world assumption, so this says that only the birth year is to be preserved—really, that if the predicates on the left of the $\rightarrow$ hold, then the predicate on the right holds.

Prolog uses a different model. As a logic programming language, the same constraints are expressed quite differently:

```
BirthYearOnly(X) :- Person(X), BirthYear(X,Year),
    Year \== 0, BirthDay(X,D), BirthMonth(X,M),
     Suppress(D), Suppress(M).
    Suppress([]).
    Suppress(0).
```

Here, the parts of the person's birth date (day, month, year) are extracted, and only the year is preserved; the others are suppressed.

Both Prolog and SWRL processing systems can detect conflicts. Unlike OWL, Prolog uses a closed-word assumption, so unasserted facts are deemed to be false. In the context of this project, ontologies are unlikely to be complete, so the closed world assumption will likely lead to very misleading results.

The analysis requires age, hours of service, the patient's insurance status, and total claims made (which may be modified by not more than 5%). We express this as two separate constraints. The first deals with the claims:

$$\begin{aligned} R\_VC: \quad & Person(?x) \wedge hasClaim(?x, ?orig) \wedge \\ & hasApproxClaim(?x, ?approx) \wedge \\ & swrlb:subtract(?d, ?orig, ?approx) \wedge \\ & swrlb:abs(?a, ?d) \wedge \\ & swrlb:lessThan(?a, 0.05) \rightarrow ValidClaim(?x) \end{aligned}$$

This defines the value of the predicate $ValidClaims$. We now use this to state the analysis requirement:

$$\begin{aligned} R\_AR: \quad & Person(?x) \wedge hasBilledTime(?x, ?time) \wedge \\ & ValidClaim(?x) \wedge Insurance(?x) \wedge \\ & BirthYear(?x) \rightarrow sqwrl:select(?x) \end{aligned}$$

This expression examines all records, and selects those for which the expression is true. Note the presence of the privacy requirement ($BirthYear(?x)$). An advantage of using OWL is that it is supported by reasoners. Tools also support the rules extensions for SWRL, which is a proposed W3C standard.

4.4 Summary

From the above, and the analysis of the requirements of sanitization, the most appropriate manner of expression of the requirements is in a language reflecting an open world model. Some form of OWL, combined with SWRL, would be particularly desirable as these are widely used, and there are reasoning engines for both. Further, they are able to express many relationships that enable known desanitization algorithms to succeed, such as the Netflix example shown in Figure 2.

5. DISCUSSION

What language, or languages, are most appropriate for expressing privacy and analysis requirements? Will one language work equally well for all domains, or must each domain have its own language? One possibility is to use a common basic language, with extensions for domain-specific attributes when necessary. Further, privacy requirements are not static, and the analysis requirements are also dynamic. The obvious way to handle this is to redo the composition of privacy and analysis requirements whenever either changes. This works if the changes are infrequent (and part of future, domain-specific, research is to define "infrequent"). It does *not* solve the problem of the use of previously released data.

This last point bears further examination. Suppose one data set is properly sanitized and released. Then a second one is sanitized and released. Separately, they cannot be desanitized. But can the *combination* be desanitized? How, in fact, do we handle such temporally sequential releases? It is infeasible to remove data that has been released, as demonstrated by the incident with the AOL release described in Section 3.2.

Ontologies are key to this approach. Some domains have ontologies that may be useful. What makes ontologies useful must be characterized, as must a method to determine their completeness. This is crucial, because the more relationships an ontology captures, the more effective will be the sanitization. But how does one know if *all* such relationships are captured? Some may not be apparent, especially when

external data provides the linkage. As an example, the relationship between the triple (ZIP code, gender, birthdate) and name is not at all obvious even though studies have now demonstrate it. Data mining approaches may prove very useful in determining such unsuspected relationships.

Another question is whether ontologies can be combined to identify relationships among data in different domains. For example, if we wish to combine medical data with smart grid data (because a resident has a severe medical condition requiring medical equipment be used at home), can we combine the ontologies of those subject domains to sanitize the data from the combined domains? This speaks to both the relationships that need to be captured and the languages of the domains. It again raises the question of whether the languages used to express requirements in the domains are the same—the issue here applies to the expression of relationships within and across the domains.

An additional point is that, as ontological relationships enabling desanitization are defined, one can collect them into a "toolbox" or "reference set" that can then be used by others, thereby allowing for better sanitization methods over time. In essence, the community learns from previous work.

Still another issue is the dynamic nature of most domains. Relationships within (and among) domains change over time, partly as the data gathered within the domain evolves and partly as new relationships are uncovered. How should this be handled? An interesting consequence is whether the ontology itself can be attacked. Suppose an adversary could add bogus relationships to the ontology. These relationships would either suppress or reveal data that would enable the adversary to desanitize the dataset. This is a generalization of the *marker* attack, in which an adversary injects data of a specific form or content that she can recognize in the sanitized data, thus aiding in the attack.

Perhaps the most constructive approach is to provide two sets of relationships. The first lists those relationships that are known to hold in the raw data, and must not hold if desanitization is to be prevented. The second is a set of relationships that, *if they held*, would enable desanitization. The sanitizer can deal with the first set as appropriate. The second enables the sanitizer to perform a simple risk analysis, centered on two questions: (1) What is the probability that the relationships in this set hold; (2) What is the probability that the adversary will be able to determine that the relationships hold, and use that to desanitize the data?

Handling external data is tricky, because the sanitizer has no control over the existence or propagation of that data. Consequently, the best approach is the one described in item 5, above: characterize the external data that enables desanitization, and allow the sanitizer to decide what the risk of an adversary finding that data is. This way, one need not search for information satisfying those relationships—a task that is doomed to miss something, given the amount of information available in this "information age."

We distinguish between *sharing* data and *publishing* data. An entity sharing data can constrain its use, and hence knows or can determine the analysis needs of the other parties. An entity publishing data loses all control over that data, and hence does not know the analysis needs of the other parties. This raises an intriguing question. One can sanitize all information *except* that needed to perform the analysis, or sanitize *only* that information needed to preserve privacy. If one is publishing data, clearly the first choice is impossible (as the sanitizer does not know the analysis requirements). But either approach is possible when sharing data, and in fact the Principle of Least Privilege [79] would require the former

A key part of any research is validation. Here, our contention is that the approach described in this report enables one to sanitize data in such a way that it cannot be desanitized by exploiting relationships that are captured in the ontology. Validating this approach requires two tests. The first is to show that this method encompasses existing sanitization methods. The second is to show that existing *desanitization* methods will be defeated. The large body of work on methods to sanitize and desanitize data provides the basis for this validation.

One interesting question is: how effective is the desanitization? Effectiveness is difficult to measure. Information theoretic measures sound appealing, and indeed may work in some cases. But the problem is the assumptions that underlie those measures. So, measuring those assumptions seems to be another appropriate type of metric. As the assumptions are about relationships, one metric might be the number of relationships that must hold in order to undo the sanitization. A second corresponds to the notion of "work factor" in penetration studies: how hard is it to show that a particular set of relationships either hold or do not hold? These ideas, and others, need to be explored in greater depth.

The human factors aspect of sanitization is critical: how to convince people whose data is being shared that in fact the sanitization is, and will remain, effective? In some sense, this is a form of assurance from the point of view of the people whose records are being sanitized. This aspect has not been explored.

Our work also suggests a different approach to the notion of privacy. Definitions of privacy usually center on some notion of the ability to control who has access to what information about yourself. As noted above, this may be too limiting. Anthea may not be a member of the Nasty Political Party, but she may strongly object to an inference that she is a member. Is it a violation of privacy for someone to be able to conclude, with more certainty than a mere guess, that she is?

Another way to view privacy is the ability to define a threshold of information that may be released safely. For example, Anthea may state that the threshold for determining her correct political party affiliation is $\frac{1}{n}$ (where n is the number of political parties), which means that an adversary may not do better than guessing randomly which party she is a member of. This leads to a definition of *absolute privacy*: the adversary's chance of correctly inferring unsanitized data from the sanitized data is no better than guessing. (This is similar to the definition of a perfect cipher: knowing the ciphertext does not add any information about the plaintext.)

Now generalize this notion using the distance metric defined in section 2. Define the relation $Privacy_{\alpha,\epsilon}$ as providing the following degree of privacy: $Pr[\Delta(f(d), s(d)) < \alpha] < \epsilon$; in other words, the probability is sufficiently small (ϵ) that an adversary cannot determine the data d that one wishes to protect to within a threshold α. This allows Anthea to define different levels of privacy for different data about her. How to integrate these notions with data sanitization rigorously, and indeed how to extend them to other arenas, is a

potentially far-reaching challenge.

6. CONCLUSION

The contribution of this work is twofold. First, we present a methodology of sanitization that embraces an open-world philosophy, in which the relationships to be used by the attacker may be unknown to the sanitizer. Thus, we assume the adversary can draw on information beyond that in the data set. Second, we do not assert that effective sanitization methods always exist. Instead, by capturing those relationships that cause the sanitization to be reversed, we provide information that the sanitizer can use to assess risk. That is, the question of what the likelihood is, in the estimate of the sanitizer (and other interested parties) that an adversary may be able to desanitize the data is reduced to the question of the likelihood of the adversary establishing that specified relationships exist between attributes of the data. The relationships may not exist. They may, but be undiscoverable by an adversary. Or, they may be easily discovered. The sanitizer can use this to determine what data to sanitize. Therein lies the crux of this work.

It is important to determine *what* data to sanitize before deciding *how* to sanitize that data. The goal of sanitization is to prevent an adversary from making unwanted inferences from the sanitized data, such as the ability to extract the original, raw data corresponding to the sanitized data. As shown above, desanitization techniques exploit relationships among data—either within the sanitized data set, or between the sanitized data set and external information. Our research makes these relationships explicit, and integrates them into the decision about what to sanitize. Further, our research provides a mechanism to detect the effects of sanitization on the desired analyses, and in particular to identify conflicts that must be resolved.

One aspect of our work bears emphasizing. We consider "privacy requirements" to include those inferences that the people involved do not want an adversary to be able to draw. *It does not matter whether those inferences are correct. What matters is that they could be drawn.* How to integrate this generality into our model is a challenging topic, and indeed lies at the heart of much interesting work.

7. ACKNOWLEDGEMENTS

We thank Scott Campbell (Lawrence Berkeley National Laboratory), Thomas Edgar, Glenn Fink, J. D. Fluckiger, and Frank Greitzer (Pacific Northwest National Laboratory), and Mary Pat Curry and Davera Gabriel (University of California , Davis Medical Center) for their help.

This material is based in part upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

This work was also supported in part by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Energy.

Matt Bishop was supported in part by grant CCF-0905503 from the National Science Foundation to the University of California , Davis. Sean Peisert was supported in part by the National Science Foundation under Grant Number CNS-0831002. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. ADDITIONAL AUTHORS

Additional authors: Deborah Frincke (Pacific Northwest National Laboratory, Richland, WA 99352 USA, *email*: deborah.frincke@pnl.gov) and Michael Hogarth (Dept. of Pathology and Laboratory Medicine, University of California Davis Health System, Sacramento, CA 95820 USA, *email*: mahogarth@ucdavis.edu).

9. REFERENCES

[1] IDEF5 method report. Technical report, Knowledge Based Systems, Inc., College Station, TX 77840, 1994.

[2] Presidential decision directive/NSC-63: Critical infrastructure protection, May 1998.

[3] Final NIH statement on sharing research data, Feb. 2003.

[4] Homeland security presidential directive 7: Critical infrastructure identification, prioritization, and protection, Dec. 2003.

[5] Protecting personal health information in research: Understanding the HIPAA privacy rule. Publication 03-5388, National Institutes of Health, Bethesda, MD, 2003.

[6] DHS information sharing and access agreements. Publication 2009-01, Department of Homeland Security, May 2009.

[7] A. Acquisti and R. Gross. Predicting social security numbers from public data. *Proceedings of the National Academy of Sciences*, 106(27):10975–10980, July 2009.

[8] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM SIGMOD-SIGAC-SIGART Symposium on Principlkes of Database Systems*, pages 247–255, 2001.

[9] E. Astesiano, M. Bidoit, H. Kirchner, B. Krieg-Brückner, P. Mosses, D. Sannella, and A. Tarlecki. CASL: the common algebraic specification language. *Theoretical Computer Science*, 286(2):153–196, 2002.

[10] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. Blocking anonymity threats raised by frequent itemset mining. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, Nov. 2005.

[11] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. *New York Times*, Aug. 9, 2006.

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

[13] S. Bhansali and B. N. Grosof. Extending the SweetDeal approach for e-procurement using SweetRules and RuleML. In *Proceedings of the 2005 Conference on Rules and Rule Markup Languages for the Semantic Web*, pages 113–129, 2005.

[14] B. Bhumiratana. *Privacy Aware Micro Data Sanitization*. PhD thesis, Dept. of Computer Science, University of California at Davis, Davis, CA 95616-8562, 2009.

[15] B. Bhumiratana and M. Bishop. Privacy aware data sharing: Balancing the usability and privacy of datasets. In *Proceedings of the 2nd International Conference on Pervsive Technologies Related to Assistive Environments (PETRA 2009)*, pages 1–8, New York, NY, USA, June 2009. ACM.

[16] M. Bishop, B. Bhumiratana, R. Crawford, and K. Levitt. How to sanitize data? In *Proceedings of the Thirteenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2004)*, pages 217–222, Los Alamitos, CA, USA, June 2004. IEEE.

[17] M. Bishop, R. Crawford, B. Bhumiratana, L. Clark, and K. Levitt. Some problems in sanitizing network data. In *Proceedings of the Fifteenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2004)*, pages 307–312, June 2006.

[18] A. Blake and R. Nelson. Scalable architecture for prefix preserving anonymization of IP addresses. In *Proceedings of the 8th international Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation*, July 2008.

[19] H. Boley, M. Kifer, P.-L. Patranjan, and A. Polleres. Rule interchange on the web. In *Proceedings of the Third International Summer School on the Reasoning Web*, pages 269–309, Sep. 2007.

[20] H. Boley, S. Tabet, and G. Wagner. Design rationale of RuleML: A markup language for semantic web rules. In *Proceedings of the Semantic Web Working Symposium*, 2001.

[21] R. Boyle. The unsuccessful experiment. In *Certain Physiological Essays*. Henry Herringman, 1661.

[22] D. Brickley and R. Guha. Resource description framework (RDF) schema specification 1.0. Technical report, W3C, Oct. 2000.

[23] M. Burkhart, D. Brauckhoff, and M. May. On the utility of anonymized flow traces for anomaly detection. In *Proceedings of the 19th ITC Specialist Seminar on Network Usage and Traffic*, Oct. 2008.

[24] M. Burkhart, D. Schatzmann, B. Trammell, E. Boschi, and B. Plattner. The role of network trace anonymization under attack. *ACM SIGCOMM Computer Communication Review*, 40(1):5–11, January 2010.

[25] J. Cao, B. Carminati, E. Ferrari, and K. L. Tan. CASTLE: A delay-constrained scheme for k_s-anonymizing data streams. In *Proceedings of the IEEE 24th International Conference on Data Engineering ICDE 2008*, pages 1376–1378, 2008.

[26] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the semantic web recommendations. In *Proceedings of the 13th International World Wide Web Conference*, pages 74–83, 2004.

[27] S. E. Coull, C. V. Wright, A. D. Keromytis, F. Monrose, and M. K. Reiter. Taming the devil: Techniques for evaluating anonymized network data. In *Proceedings of the 15th Network and Distributed System Security Symposium*, 2008.

[28] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, and M. K. Reiter. Playing devil's advocate: Inferring sensitive information from anonymized network traces. In *Proceedings of the 14th Network and Distributed System Security Symposium*, Feb. 2007.

[29] R. Crawford, M. Bishop, B. Bhumiratana, L. Clark, and K. Levitt. Sanitization models and their limitations. In *Proceedings of the 2006 Workshop on New Security Paradigms (NSPW 2006)*, pages 41–56, New York, NY, USA, Sep. 2006. ACM.

[30] M. Dean, G. Schreiber, S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. Stein. OWL web ontology language reference. Technical report, W3C, Feb. 2004.

[31] H. Delugach. Common logic (CL): A framework for a family of logic-based languages. Standard ISO/IEC 24707:2007, International Organization for Standardization, 2007.

[32] J. Dolby, A. Fokoue, A. Kalyanpur, E. Schonberg, and K. Srinivas. Scalable highly expressive reasoner (SHER). *Web Semantics*, 7(4):357–361, Dec. 2009.

[33] J. Fan, J. Xu, M. Ammar, and S. Moon. Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme. *Computer Networks*, 46(2):253–272, 2004.

[34] S. R. Ganta and R. Acharya. Adaptive data anonymization against information fusion based privacy attacks on enterprise data. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 1075–1076, New York, NY, USA, 2008. ACM.

[35] S. R. Ganta and R. Acharya. On breaching enterprise data privacy through adversarial information fusion. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering Workshop*, pages 246–249, Washington, DC, USA, 2008. IEEE Computer Society.

[36] J. Gardner and L. Xiong. An integrated framework for de-identifying unstrutured medical data. *Data and Knowledge Engineering*, 68(12):1441–1451, Dec. 2009.

[37] M. R. Genesereth and R. E. Fikes. Knowledge interchange format version 3.0 reference manual. Technical reportlogic-92-1, Computer Science Department, Stanford University, Stanford, CA, 1992.

[38] P. Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the Fifth ACM Workshop on Privacy in Electronic Society*, pages 77–80, New York, NY, USA, 2006. ACM.

[39] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.

[40] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker.

Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th International Conference on the World Wide Web*, pages 48–57. ACM, 2003.

[41] X. He, J. Vaidya, B. Shafiq, N. Adam, and V. Atluri. Preserving privacy in social networks: A structure-aware approach. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies WI-IAT '09*, volume 1, pages 647–654, Oct. 2009.

[42] I. Horrock, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML. Technical report, W3C, May 2004.

[43] I. Horrocks. DAML+OIL: a description logic for the semantic web. *Bulletin of the Technical Committee on*, 51:4, 2002.

[44] M. Z. Islam and L. Brankovic. A framework for privacy preserving classification in data mining. In *Proceedings of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 163–168, 2004.

[45] M. Jang and J.-C. Sohn. Bossam: An extended rule engine for OWL inferencing. In *Proceedings of the 2004 Conference on Rules and Rule Markup Languages for the Semantic Web*, pages 128–138, 2004.

[46] J. Jin and X. Wang. On the effectiveness of low latency anonymous network in the presence of timing attack. In *Proceedings of the 2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, pages 429–438, 2009.

[47] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[48] E. E. Kenneally and K. Claffy. Dialing privacy and utility: a proposed data-sharing framework to advance internet research. *IEEE Security and Privacy*, 8(2), Mar. 2010.

[49] A. Khoshgozaran, H. Shirani-Mehr, and C. Shahabi. SPIRAL: A scalable private information retrieval approach to location privacy. In *Proc. Ninth International Conference on Mobile Data Management Workshops MDMW 2008*, pages 55–62, 2008.

[50] J. J. Kim and W. E. Winkler. Masking microdata files. Technical report, Bureau of the Census, 1997.

[51] J. J. Kim and W. E. Winkler. Multiplicative noise for masking continuous data. In *Proceedings of the Annual Meeting of the American Statistical Association*, 2001.

[52] D. Koukis, S. Antonatos, D. Antoniades, E. P. Markatos, and P. Trimintzios. A generic anonymization framework for network traffic. In *Proceedings of the 2006 IEEE International Conference on Communications*, volume 5, pages 2302–2309, June 2006.

[53] T. S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, 1962.

[54] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the IEEE 23rd International Conference on Data Engineering*, pages 106–115, June 2007.

[55] T. Li and N. Li. Injector: Mining background knowledge for data anonymization. In *Proceedings of the IEEE 2008 International Conference on Data Engineering*, pages 446–455. IEEE Computer Society, Apr. 2008.

[56] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 93–106, New York, NY, USA, 2008. ACM.

[57] D. Lücke and T. Mossakowski. Heterogeneous model finding with hets. In *Preliminary Proceedings of the 19th International Workshop on Algebraix Development Techniques*, pages 58–61, June 2008.

[58] G. LukÁcsy and P. Szeredi. Efficient description logic reasoning in Prolog: The DLog system. *Theory and Practice of Logic Programming*, 9(3):343–414, 2009.

[59] F. G. M. Atzori, F. Bonchi and D. Pedreschi. k-anonymous patterns. In *Proceedings of the Ninth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)*, volume 3721 of *Lecture Notes in Computer Science, Springer*, Porto, Portugal, October 2005.

[60] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*, Apr. 2006.

[61] M. Moriconi and R. A. Riemenschneider. Introduction to SADL 1.0: A language for specifying software architecture hierarchies. Technical Report SRI-CSL-97-01, SRI International, Mar. 1997.

[62] P. D. Moses. *CASL Reference Manual*, volume 2960 of *Lecture Notes in Computer Science*. Springer, 2004.

[63] B. Motik, P. Patel-Schneider, B. Parsia, C. Bock, A. Fokoue, P. Haase, R. Hoekstra, I. Horrocks, A. Ruttenberg, U. Sattler, et al. OWL 2 web ontology language structural specification and functional-style syntax. Technical report, W3C, Oct. 2009.

[64] B. Motik, U. Sattler, and R. Studer. Query answering for OWL-DL with rules. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(1):41–60, 2005. Rules Systems.

[65] K. Muralidhar, R. Parsa, and R. Sarathy. A general additive data perturbation method for database security. *Management Science*, 45(10):1399–1415, Oct. 1999.

[66] K. Muralidhar, R. Parsa, and R. Sarathy. Security of random data perturbation methods. *ACM Transactions on Database Systems*, 24(4):487–493, Dec. 1999.

[67] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 111–125, May 2008.

[68] A. Narayanan and V. Shmatikov. De-anonymizing social networks. *Proceedings of the 2009 IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[69] National Center for Health Statistics and the Centers for Medicare and Medicaid Services, Hyattsville, MD 20782. *International Classification of Diseases, Ninth Revision, Clinical Modification*, Oct. 2009.

[70] M. E. Nergiz and C. Clifton. δ-presence without complete world knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 22, 2010.

[71] M. E. Nergiz, C. Clifton, and A. E. Nergiz. Multirelational k-anonymity. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1104–1117, Aug. 2009.

[72] A. Panchenko and L. Pimenidis. Cross-layer attack on anonymizing networks. In *Proceedings of the 2008 International Conference on Telecommunications*, pages 1–7, June 2008.

[73] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38, January 2006.

[74] P. Porras and V. Shmatikov. Large-scale collection and sanitization of network security data: Risks and challenges (position paper). In *Proceedings of the 2006 Workshop on New Security Paradigms*, pages 57–64, Sep. 2006.

[75] Racer Systems GmbH & Co. KG, Hamburg, Germany. *RacerPro User's Guide Version 1.9*, Dec. 2005.

[76] S. L. Reed and D. B. Lenat. Mapping ontologies into Cyc. In *Proceedings of the 2002 AAAI Conference Workshop on Ontologies for the Semantic Web*, pages 1–6, July 2002.

[77] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, Nov. 1998.

[78] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-peer based anonymous internet usage with collusion detection. In *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society*, pages 91–102, 2002.

[79] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, Sep. 1975.

[80] N. Singer. When 2+2 equals a privacy question. *New York Times*, Oct. 18, 2009.

[81] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Webervices and agents on the World Wide Web*, 5(2):51–53, 2007.

[82] P. Spyns, R. Meersman, and M. Jarrar. Data modelling versus ontology engineering. *ACM SIGMOD Record*, 31(4):12–17, Dec. 2002.

[83] X. Sun, H. Wang, and J. Li. Injecting purpose and trust into data anonymisation. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pages 1541–1544, New York, NY, USA, 2009. ACM.

[84] L. Sweeney. Uniqueness of simple demographics in the U. S. population. Technical Report LIDAP-WP4, Laboratory for International Data Privacy, Carnegie Mellon University, Pittsburgh, PA, USA, 2000.

[85] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, Oct. 2002.

[86] G. Szarvas, R. Farkas, and R. Busa-Fekete. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580, Sep. 2007.

[87] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. *Automated Reasoning*, pages 292–297, 2006.

[88] A. van Renssen. Gellish: An information representation language, knowledge base, and ontology. In *Proceedings of the 3rd IEEE Conference on Standardization and Innovation In Information Technology*, pages 215–228, Oct. 2003.

[89] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 466–473, Houston, TX, November 2005.

[90] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *Knowledge and Information Systems*, 11(3):345–368, Apr. 2006.

[91] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-up generalization: a data mining solution to privacy protection. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 249–256, Nov. 2004.

[92] S. Warren and L. D. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[93] R. Wong, J. Li, A. Fu, and K. Wang. (α, k)-anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In *Proceedings of the 12th ACM SIGKDD Iinternational Conference on Knowledge Discovery and Data Mining*, pages 754–759, 2006.

[94] M. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communication systems. *ACM Transactions on Information Systems Security*, 7(4):489–522, Nov. 2004.

[95] L. Xiao, Z. Xu, and X. Zhang. Low-cost and reliable mutual anonymity protocols in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 14(9):829–840, Sep. 2003.

[96] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 229–240, 2006.