Privacy in Database Publishing: A Bayesian Perspective

Alin Deutsch*

Department of Computer Science and Engineering
University of California San Diego
9500 Gilman Dr., La Jolla, CA, 92093-0404, USA
deutsch@cs.ucsd.edu

Summary. We present a unifying perspective of privacy guarantees in view-based and generalization-based publishing. This perspective uses a generic Bayesian privacy model which generalizes both types of publishing scenarios and allows us to relate seemingly disparate privacy guarantees found in the literature.

1 Introduction

Database publishing systems export parts of a proprietary database for consumption by client applications. The design of a publishing system is subject to two conflicting requirements. On one hand, the data owner needs to publish appropriate parts of the proprietary data to support various interactions with her clients. On the other hand she must protect certain sensitive data from being disclosed to clients.

In this chapter, we discuss data *privacy* which pertains to defense against attackers who access the data legally. These attackers are regular clients who inspect the published data and potentially combine it with external knowledge to infer information about the secret data. Note that privacy is orthogonal to data *security*, whose goal is defense against unauthorized access to the database using access control mechanisms.

We focus on two classes of publishing systems. In *view-based* publishing, the owner specifies the data to be released by means of views defined in some standard query language. In *generalization-based* publishing, the released data is specified using a formalism of incomparable expressive power, namely anonymization using generalization functions. Examples of anonymization via generalization include replacing a person's actual age by an age range, removing the least significant digits of the zip code, etc.

* Funded by an Alfred P. Sloan fellowship and by NSF CAREER award IIS-0347968.

The two corresponding lines of privacy research have evolved independently, yielding different formalisms for stating privacy guarantees. In this chapter, we show that privacy guarantees in view-based and generalization-based publishing are related, being both particular cases of guarantees in a general privacy model. We call this model the Generic Bayesian Privacy (**GBP**) model as it offers guarantees based on the revision of the attacker's belief about the secret between the state before and after seeing the published data.

We start by developing in Section 2 a generic model for attacks attempting to glean knowledge about the sensitive part of the database starting from the published part thereof, also exploiting external knowledge. In Section 3, we show how privacy guarantees developed for view-based publishing systems can be cast as particular cases in the **GBP** model. Then in Section 4 we connect generalization-based publishing to the **GBP** model. Exploiting the uniform formalization using the **GBP** model, Section 5 compares various privacy guarantees from both view-based and generalization-based publishing. Finally, Section 6 shows how the **GBP** model can be applied to formulate and check meaningful privacy guarantees for publishing in open-world information integration systems.

2 GBP: A Generic Bayesian Privacy Model

The published data. The data owner publishes part of the database D, possibly after some processing such as filtering, aggregation, anonymization, etc. For the purpose of our discussion, this processing can be modeled as a function $\mathcal{V}$, whose result $\mathcal{V}(D)$ is being released.

The secret. The owner wishes to keep sensitive data secret. Since sensitivity depends on the application and is best judged by the data owner, she must be provided with the possibility to declare which data is to be kept secret. The secret may be a subset of the database, possibly altered by processing, which we shall model as another function $\mathcal{S}$, whose result $\mathcal{S}(D)$ is the secret.

We note that in the generic model, $\mathcal{V}$ and $\mathcal{S}$ are arbitrary functions from databases to databases. However, in the running example of this section, we shall express such functions by queries. We shall see in Section 4 examples of functions expressed differently, as anonymization functions.

Example 1. Consider a database whose only relation contains tuples associating the patient with the ailment he suffered from and the doctor who treated him:

$$PDA(patient, doctor, ailment).$$

The secret $\mathcal{S}$ is the association between patients and their ailment, specifiable by the owner for instance using query $\mathcal{S}(p, a) :- PDA(p, d, a)$.

2.1 Attacks

In this model we only consider attackers who access the data legally by inspecting the published data $\mathcal{V}(D)$, using it together with external knowledge to infer information about the secret $\mathcal{S}(D)$. The defense against unauthorized access to the database is beyond the scope of this model.

Possible databases. Ideally, the attacker would like to reverse-engineer D starting from the observed published data $\mathcal{V}(D)$. This would immediately lead to the full disclosure of the secret: the attacker could compute the secret by directly running $\mathcal{S}$ over D. Of course, $\mathcal{V}$ is likely to be a lossy data transformation, thus precluding the unequivocal identification of its arguments from its output. In general there may be (potentially infinitely) many databases which have the same image as D under $\mathcal{V}$. The attacker cannot distinguish among them solely by observing the published data $\mathcal{V}(D)$, regardless of the computational resources at his disposal. Therefore, in the absence of external knowledge about D, all databases with the same image are possible from the attacker's point of view (we will shortly introduce the attacker's external knowledge into the model). We therefore refer to the set $[D]_{\mathcal{V}}$ of databases as the *possible* databases given $\mathcal{V}(D)$:

$$[D]_{\mathcal{V}} := \{D' \mid \mathcal{V}(D') = \mathcal{V}(D)\}.$$

Example 2. Continuing Example 1, assume that the owner publishes a view listing all the patients $V_p(p) :- PDA(p, d, a)$ and one listing all ailments treated by the hospital: $V_a(a) :- PDA(p, d, a)$. Assume that on the actual database D, $V_p(D)$ yields {John, Jane} and $V_a(D)$ yields {flu, pneumonia}. Then some of the possible databases corresponding to the observed views are $D_1 =$ { (John, doc_1, flu), (Jane, doc_2, pneumonia) }, $D_2 =$ { (John, doc_3, flu), (John, doc_3, pneumonia), (Jane, doc_4, flu) }, etc., where doc_i are unknown doctor names.

Clearly the set of possible databases may be very large. For example, consider the case when the published data is a projection of a table. By observing the published table (and using no external knowledge about the data), an attacker must assume any possible completion for the missing columns. This is the case in Example 2 if the attacker does not know the set of all possible doctors.

It is therefore not a priori given that the attacker is even able to enumerate all possible databases. In the following, we assume the worst-case scenario for the owner, namely that the attacker comes up with some finite representation of the set of possible databases which he uses for reasoning about the secret. Note that the more advantage we assume for the attacker, the stronger any privacy guarantees based on these assumptions.

Possible secrets. Since the owner cares about guarding only the secret (rather than the non-sensitive parts of the database), the privacy model focuses on possible secrets. From a reasonable attacker's point of view, a secret

s is *possible* only if it is witnessed by some possible database i.e. if there exists $D' \in [D]_{\mathcal{V}}$ such that $s = \mathcal{S}(D')$. Without worrying yet whether the attacker can even compute all possible secrets, note that they provide an upper bound on the set of candidates for the secret which an attacker needs to consider. Let us denote the set of possible secrets with $\mathcal{S}([D]_{\mathcal{V}})$:

$$\mathcal{S}([D]_{\mathcal{V}}) := \{\mathcal{S}(D') \mid D' \in [D]_{\mathcal{V}}\}.$$

In particular, the actual secret $S(D)$ is a possible secret: $\mathcal{S}(D) \in \mathcal{S}([D]_{\mathcal{V}})$.

Example 3. Continuing Example 2, the possible secrets are obtained by running the $\mathcal{S}$ over each possible database. We obtain $s_1 = \mathcal{S}(D_1) = \{$(John, flu), (Jane, pneumonia)$\}$, $s_2 = \mathcal{S}(D_2) = \{$ (John, flu), (John, pneumonia), (Jane, flu) $\}$, etc.

The optimal attack: compute possible secrets and use external knowledge. In the absence of external knowledge, possible secrets are indistinguishable with respect to the published data $\mathcal{V}(D)$ and even with unlimited computational resources the best an attacker can hope for is to reverse-engineer $\mathcal{S}([D]_{\mathcal{V}})$. Towards a conservative privacy guarantee, let's assume that the attacker is successful at this task, handling the case of infinitely many possible secrets by coming up with a finite representation thereof.[2] If there is only one possible secret, then the actual secret is exposed and the attacker's task accomplished. In the (likely) case of several possible secrets, a sophisticated attacker improves his chances of singling out the actual secret by whittling down $\mathcal{S}([D]_{\mathcal{V}})$ using external knowledge. If several possible secrets remain even now, the attacker is forced to guess the actual secret among them. However, the guess does not have to be uneducated: while the attacker's external knowledge may be insufficient to further rule out any possible secrets, it could still influence the attacker's beliefs about the relative likelihood of the possible secrets. This would enable the attacker to pick the secret he believes likeliest. Finally, if the attacker deemed several possible secrets equally likely but likelier than all others, he would be forced to guess at random among them.

Modeling attacker's belief. The attacker's external knowledge can pertain to the possible databases or exclusively to the possible secrets. Note that any attacker who forms an opinion on how to rank possible databases can infer the ranking of the corresponding secrets and is therefore at least as knowledgeable (and dangerous) as an attacker who does not understand or care about the underlying database, focusing solely on the secret.

To defend against the more dangerous class of attackers, we model the attacker's *a priori* belief (i.e. before observing $\mathcal{V}(D)$) as a probability distribution δ on all databases. This induces a belief (probability distribution) $\mathbf{P}_\delta$ on all secrets as follows: given candidate secret s, the probability $\mathbf{P}_\delta[s]$ that s is the actual secret is the sum of probabilities of all databases witnessing s:

[2] We know such representations exist: (an admittedly crude) one is given by the definition of $\mathcal{V}$ together with $\mathcal{V}(D)$.

$$\mathbf{P}_\delta[s] := \sum_{s=\mathcal{S}(D')} \delta(D'). \quad (1)$$

δ also induces the probability $\mathbf{P}_\delta[\mathcal{V}(D)]$ that the published data is $\mathcal{V}(D)$:

$$\mathbf{P}_\delta[\mathcal{V}(D)] := \sum_{D' \in [D]_\mathcal{V}} \delta(D').$$

The actual release of the published data causes a revision of the attacker's belief about the probability of s being the actual secret. We call this the *a posteriori* probability, and it is the conditional probability $\mathbf{P}_\delta[s|\mathcal{V}(D)]$:

$$\mathbf{P}_\delta[s|\mathcal{V}(D)] = \frac{\mathbf{P}_\delta[s \wedge \mathcal{V}(D)]}{\mathbf{P}_\delta[\mathcal{V}(D)]} = \frac{\sum_{D' \in [D]_\mathcal{V}, \mathcal{S}(D')=s} \delta(D')}{\sum_{D' \in [D]_\mathcal{V}} \delta(D')}. \quad (2)$$

Classes of attackers. For all privacy guarantees we consider next, we conservatively assume that the attacker is able to reverse-engineer the possible databases and secrets from the published data. Attackers are therefore distinguished from each other exclusively by their belief about the likelihood of databases, as induced by the external knowledge they possess. Consequently, in the following we characterize an attacker by the probability distribution δ he associates on all databases. A class of attackers we wish to defend against is then described by a family $\mathcal{P}$ of probability distributions.

2.2 Privacy Guarantees

Privacy guarantees rule out privacy breaches. We list below several alternative guarantees that generalize guarantees considered in the literature. Each one is determined by the definition of what constitutes a "breach".

Extent-Dependent Guarantees. We start with a class of guarantees which depend on the extent of actual database D. Each of them take as argument a publishing function $\mathcal{V}$ and hold if and only if publishing $\mathcal{V}(D)$ does not breach privacy.

No complete database exposure ($\mathbf{NDE}^D$). The worst case of breach consists in complete exposure of the actual database D. That is, the breach is defined as the case when the only possible database is D: $[D]_\mathcal{V} = \{D\}$. In this case, an attacker who successfully reverse-engineers the possible databases retrieves the actual database and can then compute *any* secret function $\mathcal{S}$ on it. The guarantee of no database exposure, denoted $\mathrm{NDE}^D(\mathcal{V})$, requires at least two possible databases:

$$\mathrm{NDE}^D(\mathcal{V}) := |[D]_\mathcal{V}| \geq 2.$$

Example 4. Assume that in the setting of Example 1, the hospital publishes a view revealing which doctors every patient sees: $V_{PD}(p, d) :- PDA(p, d, a)$. An additional view is published as well, listing which ailments every doctor is

treating: $V_{DA} :- PDA(p, d, a)$. If for some database D the view extents are $V_{PD}(D) = \{$ (John, Dr. MacDonald) $\}$ and $V_{DA}(D) = \{$ (Dr. MacDonald, pneumonia) $\}$, then D is exposed since $[D]_{V_{PD},V_{DA}}$ is the PDA table with the single tuple $\{$ (John, Dr. MacDonald, pneumonia) $\}$. If on the other hand the attacker observes $V_{PD}(D) = \{$ (John, Dr. MacDonald), (Jane, Dr. MacDonald) $\}$ and $V_{DA}(D) = \{$ (Dr. MacDonald, flu), (Dr. MacDonald, pneumonia) $\}$, then D is not exposed since there are several possible databases. One in which John has flu and Jane pneumonia, on in which John has both diseases and Jane has flu, etc.

No complete secret exposure ($\mathbf{NSE}^D_{\mathcal{S}}$). Even if the actual database is not exposed, it may be that all possible databases have the same image under $\mathcal{S}$, thus completely exposing the secret. To guard against this case, we define the breach as having a single possible secret: $\mathcal{S}([D]_{\mathcal{V}}) = \{\mathcal{S}(D)\}$. Non-exposure of the secret requires at least two possible secrets:

$$\mathrm{NSE}^D_{\mathcal{S}}(\mathcal{V}) := |\mathcal{S}([D]_{\mathcal{V}})| \geq 2.$$

Example 5. For the schema of Example 1, assume that the hospital publishes the view V_P from Example 1 and view V_{DA} from Example 4. If the attacker observes $V_P(D) = \{$ (John), (Jane) $\}$ and $V_{DA}(D) = \{$ (Dr. MacDonald, pneumonia), (Dr. Zhivago, pneumonia) $\}$, then D is not exposed since there are several possible databases: one in which John sees Dr. MacDonald and Jane Dr. Zhivago, one in which they swap doctors, one in which John sees both doctors and Jane only one of them, etc. And yet, the secret is exposed, since both doctors treat the same disease so no matter whom they see, both John and Jane must suffer from pneumonia.

No belief revision ($\mathbf{NBR}^D_{\mathcal{P},\mathcal{S}}$). The non-exposure guarantees fulfill only the very basic owner expectations. They do not suffice to put her mind at ease since attackers can "learn" something about some candidate secret, thus improving their odds of guessing the actual secret.

For a given attacker described by probability distribution δ, we define "learning something about candidate secret s" in the strongest, information-theoretic sense, as revision of attacker's belief about the secret. The *belief revision* is the change between the δ-induced a priori and a posteriori beliefs that s is the secret. Formally, a belief revision occurs precisely when $\mathbf{P}_\delta[s|\mathcal{V}(D)] \neq \mathbf{P}_\delta[s]$. The guarantee that no attacker from a class $\mathcal{P}$ revises his belief amounts to

$$\mathrm{NBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{V}) := \forall s\ \forall(\delta \in \mathcal{P})\ \mathbf{P}_\delta[s|\mathcal{V}(D)] = \mathbf{P}_\delta[s].$$

This guarantee is preferred by the owner because it makes no assumptions on the attacker's computational resources. When the guarantee holds, the owner can rest assured that nothing can be learned about the secret. The following example however shows that such a guarantee is often unreasonably strong and is violated by most publishing functions, which is why we need to set our sights on more relaxed guarantees.

Example 6. Consider the database from Example 1. Suppose that the owner exports the projection of the PDA relation on its doctor attribute: $V(d) :- PDA(p, d, a)$. Since neither patients nor ailments are exported, this publishing is seemingly safe. However, an attacker can still learn from it some (small amount of) information about the secret. Indeed, if the published list of doctors is empty, then the actual database relation must be empty as well, so no patient can suffer from any ailment. An attacker whose belief assigns non-zero probability to a possible secret containing at least one ailing patient will therefore revise this belief a posteriori. If however there is even one doctor in the published list, then there is a non-zero probability of a certain patient suffering from some disease. An attacker who is a priori certain that there are no ailing patients must revise his belief as well. Clearly, at least these two attackers have learned something about the secret upon observing the list of doctors, and the idealized guarantee $\text{NBR}^D_{\mathcal{P},\mathcal{S}}$ is violated. At the same time, ruling out this publishing amounts to asking the owner to release no data whatsoever, even if she avoids the attributes involved in the secret.

No further belief revision ($\text{NFBR}^D_{\mathcal{P},\mathcal{S}}$). Since the guarantees NDE and $\text{NSE}_{\mathcal{S}}$ are too weak, and the ideal guarantee $\text{NBR}_{\mathcal{P},\mathcal{S}}$ is too strong, we consider a more pragmatic guarantee: it assumes that the owner is willing to live with the current level in attacker's belief as induced by the already published data $\mathcal{V}(D)$, but wants to make sure that publishing any further data will not lead to further belief revision. Formally, denoting with $\mathcal{N}$ the new publishing function which the owner contemplates, a breach occurs when $\mathbf{P}_\delta[s|\mathcal{V}(D)] \neq \mathbf{P}_\delta[s|\mathcal{V}(D) \wedge \mathcal{N}(D)]$. Here, $\mathbf{P}_\delta[s|\mathcal{V}(D) \wedge \mathcal{N}(D)]$ is the belief of the attacker described by distribution δ that s is the secret, provided that both $\mathcal{V}(D)$ and $\mathcal{N}(D)$ are published:

$$\mathbf{P}_\delta[s|\mathcal{V}(D) \wedge \mathcal{N}(D)] = \frac{\mathbf{P}_\delta[s \wedge \mathcal{V}(D) \wedge \mathcal{N}(D)]}{\mathbf{P}_\delta[\mathcal{V}(D) \wedge \mathcal{N}(D)]} = \frac{\sum_{D' \in [D]_\mathcal{V} \cap [D]_\mathcal{N}, \mathcal{S}(D') = s} \delta(D')}{\sum_{D' \in [D]_\mathcal{V} \cap [D]_\mathcal{N}} \delta(D')}. \quad (3)$$

The associated guarantee is the following:

$$\text{NFBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{N}, \mathcal{V}) := \forall s \forall (\delta \in \mathcal{P})\ \mathbf{P}_\delta[s|\mathcal{V}(D)] = \mathbf{P}_\delta[s|\mathcal{V}(D) \wedge \mathcal{N}(D)].$$

Example 7. Assume that on the schema from Example 1, the owner has already published $\mathcal{V} = (V_p, V_a)$ where V_p, V_a are the views from Example 2. The owner is currently contemplating the publishing of the two new views $\mathcal{N} = (V_{PD}, V_{DA})$ from Example 4. Suppose that $V_p(D) =$ {(John),(Jane),(Jack)}, and $V_a(D) =$ {(pneumonia),(flu),(cold)}. From this observation, any attacker can reverse-engineer the set of possible databases. This includes, among others, the database $D_1 =$ {(John,doc_1,pneumonia), (Jane,doc_2,flu), (Jack,doc_3,cold)}, yielding the secret $s_1 = \mathcal{S}(D_1) =$ {(John, pneumonia), (Jane,flu), (Jack,cold)}. Given an attacker described by some distribution δ, assume that his a priori belief that s_1 is the secret is non-zero

$\mathbf{P}_\delta[s_1|\mathcal{V}(D)] > 0$. Now assume that the attacker were to observe the extents of the new views, which are V_{PD} = { (John, Dr. MacDonald), (Jane,Dr. Zhivago), (Jack,Dr. Zhivago) } and V_{DA} = { (Dr. MacDonald, flu), (Dr. Zhivago, pneumonia), (Dr. Zhivago, cold) } The attacker must now revise to 0 his a posteriori belief that s_1 is the secret. Indeed, only Dr. Zhivago treats pneumonia, but John sees Dr. MacDonald, therefore John cannot have pneumonia: $\mathbf{P}_\delta[s_1|\mathcal{V}(D) \wedge \mathcal{N}(D)] = 0$.

An alternative intuition for the no-further-belief-revision guarantee is the following. After observing $\mathcal{V}(D)$, the attacker reverse-engineers the possible databases $[D]_\mathcal{V}$ and uses his background knowledge to assign a likelihood to each of them. After subsequently observing $\mathcal{N}(D)$, the attacker rules out all databases which are possible for $\mathcal{V}(D)$ but not for $\mathcal{N}(D)$, being left with only those in $[D]_\mathcal{V} \cap [D]_\mathcal{N}$. Ruling out even one database results in re-distributing its probability over the remaining ones, thus potentially modifying the attacker's a posteriori belief about the secret. For instance, in an extreme case, the possible databases in $[D]_\mathcal{V}$ may witness two secrets s_1 and s_2. If $[D]_\mathcal{V} \cap [D]_\mathcal{N}$ rules out all witnesses of s_2 (and maybe also some but not all witnesses of s_1), then by (3) the attacker's belief about the secret being s_2 drops to 0 and the belief of s_1 becomes 1, i.e. certainty.

This intuition is formalized by the following result.

Theorem 1 ([8]). *Let $\mathcal{P}$ contain all possible distributions, thus modeling all attackers. Then for every database D and secret $\mathcal{S}$ no attacker's belief is revised upon observing $\mathcal{N}(D)$ if and only if the possible databases do not change:*

$$\forall D \ \forall \mathcal{S} \ \ NFBR^D_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V}) \Leftrightarrow [D]_\mathcal{V} = [D]_\mathcal{V} \cap [D]_\mathcal{N}$$

Note that despite being defined in probabilistic fashion, the no-further-belief-revision guarantee remarkably reduces by Theorem 1 to a purely model-theoretic problem involving reasoning solely about possible databases.

Bounded belief revision ($\mathbf{BBR}^D_{\mathcal{P},\mathcal{S}}$). It is often useful to consider relaxing privacy guarantees to allow desirable publishing functions. We next consider a natural relaxation of the $\mathrm{NBR}^D_{\mathcal{P},\mathcal{S}}$ guarantee of no belief revision, which offers the owner more control over the trade-off between privacy and utility of publishing functions. The idea is to allow revision, but only if bounded by an owner-defined threshold. In this case, a breach is formally defined as $|\mathbf{P}_\delta[s|\mathcal{V}(D)] - \mathbf{P}_\delta[s]| > \epsilon$, where $\epsilon \in [0,1]$ is the threshold. This definition of breach induces a family of privacy guarantees, parameterized by the threshold:

$$\mathrm{BBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{V},\epsilon) := \forall s \forall(\delta \in \mathcal{P}) \ |\mathbf{P}_\delta[s|\mathcal{V}(D)] - \mathbf{P}_\delta[s]| \leq \epsilon.$$

Bounded further belief revision ($\mathbf{BFBR}^D_{\mathcal{P},\mathcal{S}}$). The same idea of allowing bounded belief revision yields a natural relaxation of guarantee $\mathrm{NFBR}^D_{\mathcal{P},\mathcal{S}}$:

$$\mathrm{BFBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V},\epsilon) := \forall s \forall(\delta \in \mathcal{P}) \ |\mathbf{P}_\delta[s|\mathcal{V}(D)] - \mathbf{P}_\delta[s|\mathcal{V}(D) \wedge \mathcal{N}(D)]| \leq \epsilon.$$

Extent-Independent Guarantees. The privacy guarantees we've considered so far depend on the extent of the actual database D. The owner is thus faced with the following dilemma. Checking the guarantee on a given extent D avoids being overly conservative and rejecting those publishing functions that preserve privacy on the actual database but breach it on some other database extent D'. On the other hand, this means re-checking the privacy guarantees upon each update to D. Alternatively, we consider strengthening the above guarantees to hold over all database extents. We obtain the following list of extent-independent privacy guarantees:

$$\begin{aligned}
\mathrm{NDE}(\mathcal{V}) &:= \forall D\ \mathrm{NDE}^D(\mathcal{V}) \\
\mathrm{NSE}_{\mathcal{S}}(\mathcal{V}) &:= \forall D\ \mathrm{NSE}^D_{\mathcal{S}}(\mathcal{V}) \\
\mathrm{NBR}_{\mathcal{P},\mathcal{S}}(\mathcal{V}) &:= \forall D\ \mathrm{NBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{V}) \\
\mathrm{NFBR}_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V}) &:= \forall D\ \mathrm{NFBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V}) \\
\mathrm{BBR}_{\mathcal{P},\mathcal{S}}(\mathcal{V},\epsilon) &:= \forall D\ \mathrm{BBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{V},\epsilon) \\
\mathrm{BFBR}_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V},\epsilon) &:= \forall D\ \mathrm{BFBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V},\epsilon)
\end{aligned}$$

As before, it makes sense to carefully consider the trade-off between strength of the guarantee and utility of the publishing functions it allows. In many situations, the proprietary database is known to satisfy a set of integrity constraints $\mathcal{C}$. By imposing the unrestricted extent-independent guarantees above, the owner risks excluding a perfectly safe publishing function because it breaks the guarantees on some database that will never occur in practice since it violates the constraints. Clearly, the owner does not need the privacy guarantees to hold on all imaginable databases, but only on a subclass thereof: all databases D satisfying the constraints in $\mathcal{C}$ (denoted $D \models \mathcal{C}$). This natural relaxation yields guarantees that are extent-independent as long as the extents satisfy the constraints:

$$\begin{aligned}
\mathrm{NDE}^{\mathcal{C}}(\mathcal{V}) &:= \forall (D \models \mathcal{C})\ \mathrm{NDE}^D(\mathcal{V}) \\
\mathrm{NSE}^{\mathcal{C}}_{\mathcal{S}}(\mathcal{V}) &:= \forall (D \models \mathcal{C})\ \mathrm{NSE}^D_{\mathcal{S}}(\mathcal{V}) \\
\mathrm{NBR}^{\mathcal{C}}_{\mathcal{P},\mathcal{S}}(\mathcal{V}) &:= \forall (D \models \mathcal{C})\ \mathrm{NBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{V}) \\
\mathrm{NFBR}^{\mathcal{C}}_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V}) &:= \forall (D \models \mathcal{C})\ \mathrm{NFBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V}) \\
\mathrm{BBR}^{\mathcal{C}}_{\mathcal{P},\mathcal{S}}(\mathcal{V},\epsilon) &:= \forall (D \models \mathcal{C})\ \mathrm{BBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{V},\epsilon) \\
\mathrm{BFBR}^{\mathcal{C}}_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V},\epsilon) &:= \forall (D \models \mathcal{C})\ \mathrm{BFBR}^D_{\mathcal{P},\mathcal{S}}(\mathcal{N},\mathcal{V},\epsilon)
\end{aligned}$$

A Similar Privacy Model. [5, 6] propose a similar privacy model for relational databases, based on Bayesian belief revision. However the authors do not address the equivalent of the $\mathrm{NFBR}_{\mathcal{P},\mathcal{S}}$, $\mathrm{BBR}_{\mathcal{P},\mathcal{S}}$, and $\mathrm{BFBR}_{\mathcal{P},\mathcal{S}}$ guarantees, nor do they consider guarantees parameterized by classes of probability distributions, or integrity constraints.

3 View-Based Publishing

3.1 Independent-Tuple Attackers

The application of the privacy model from [5] to view-based publishing was pioneered in seminal work by Miklau and Suciu [19, 20].

In the setting of [19, 20], the publishing function $\mathcal{V}$ is given by a list of views. Both $\mathcal{V}$ and the secret $\mathcal{S}$ are specified by conjunctive queries with inequalities.

As in Section 2, an attacker is described by a probability distribution δ on the set of all databases. However, only attackers described by *independent-tuple* distributions are considered. These distributions treat the occurrences of any two tuples t_1, t_2 in a given database as independent events. Formally, given a domain **Dom**, denote the set of all tuples over **Dom** by *tuples(***Dom***)*. Any $D \subseteq$ *tuples*(**Dom**) is a database over domain **Dom**. δ is an independent-tuple distribution on the databases over **Dom** if it is induced by a distribution p on *tuples(***Dom***)*. That is, for any database D over **Dom** we have (by the independent-tuple assumption)

$$\delta(D) := \prod_{t \in D} p(t) \times \prod_{t \in tuples(\mathbf{Dom}) - D} (1 - p(t)).$$

The attacker's a priori and a posteriori beliefs about the secret $\mathcal{S}(R)$ are then induced by p via δ as in (1), respectively (2).

Perfect privacy. Given secret $\mathcal{S}(D)$, the views $\mathcal{V}$ are considered to preserve privacy against an attacker described by distribution δ if there is no change between the attacker's a posteriori belief (after seeing $\mathcal{V}(R)$) and his a priori belief (before seeing $\mathcal{V}(R)$) about secret $s = \mathcal{S}(D)$: $\mathbf{P}_\delta[s] = \mathbf{P}_\delta[s|\mathcal{V}(D)]$.

Given a domain **Dom**, denote with $\mathcal{P}_{\mathbf{Dom}}$ the set of all independent-tuple distributions on databases over **Dom** induced by distributions over *tuples(***Dom***)*.

Then $\mathcal{V}$ is said to maintain *perfect privacy* for secret $\mathcal{S}$, denoted $\mathbf{PerfP}_{\mathcal{S}}(\mathcal{V})$ if for every domain **Dom**, every database D over **Dom**, every secret value s and every distribution $\delta \in \mathcal{P}_{\mathbf{Dom}}$, upon observing $\mathcal{V}(D)$ the attacker does not revise his belief that s is the secret:

$$\begin{aligned}\mathbf{PerfP}_{\mathcal{S}}(\mathcal{V}) := \forall \mathbf{Dom}\ \forall (D \subseteq tuples(\mathbf{Dom}))\ \forall s\ \forall (\delta \in \mathcal{P}_{\mathbf{Dom}}) \\ \mathbf{P}_\delta[s] = \mathbf{P}_\delta[s|\mathcal{V}(D)],\end{aligned}$$

or, equivalently in the notation of the **GBP** model (Section 2.2),

$$\mathbf{PerfP}_{\mathcal{S}}(\mathcal{V}) := \forall \mathbf{Dom}\ \forall (D \subseteq tuples(\mathbf{Dom}))\ \mathrm{NBR}^D_{\mathcal{P}_{\mathbf{Dom}},\mathcal{S}}(\mathcal{V}). \qquad (4)$$

Note that perfect privacy is an extent-independent guarantee. Therefore it need not be re-checked upon every update to the database.

[19] shows that perfect privacy is decidable in Π_2^p in the combined size of the queries defining $\mathcal{V}, \mathcal{S}$. The result follows from a key lemma showing that privacy holds for all domains if it holds for *some* domain of size polynomial in the number of variables and constants appearing in the view and secret queries. Essentially, to check the guarantee on such a domain **Dom**, one simply needs to enumerate the databases over **Dom**. There are only finitely many of them (though their number is exponential in the domain size). In a follow-up paper, Machanavajjhala et al. [15] provide an alternative decision procedure which reduces perfect privacy to checking a number of containments between queries constructed from the views and secret definitions. This allows them to leverage well-known results on the complexity of query containment to identify restrictions leading to a PTIME-checkability of the perfect privacy guarantee.

In addition to a decision procedure for perfect privacy, [19] introduce also a notion equivalent to the bounded belief revision guarantee $\text{BBR}_{\mathcal{P},\mathcal{S}}$ from Section 2.2 (again considering only independent-tuple distributions). Furthermore, Miklau and Suciu consider a limited flavor of the "no further belief revision" guarantee $\text{NFBR}_{\mathcal{P},\mathcal{S}}$, in which the already published views are defined by *boolean* queries.

As recognized in [19, 20], the fact that perfect privacy only defends against attackers described by independent-tuple distributions is a limitation because it ignores attackers whose background knowledge gives them correlations between tuples. For instance, the attacker's background knowledge that reviewers r_1 and r_2 have similar research expertize and taste can be modeled by a distribution in which the probability that r_1 bids for a paper is similar to the probability that r_2 does. In an additional example, the attacker may know that if a patient has a highly contagious disease, then her spouse likely has it, too. Such background information cannot be modeled by independent-tuple distributions.

However, limiting attackers to those characterized by independent-tuple distributions strikes a good balance in the trade-off between guarantee strength and feasibility of checking the guarantee. This conclusion is reinforced by a study (discussed next) of what happens if the limitation is removed.

3.2 More General Classes of Attackers

[8] explores an alternate way to balance the tension between the strength of the guarantee and the feasibility of checking it.

The study starts from the thesis that data owners cannot presume that attacker's beliefs are induced exclusively by the independent-tuple distributions of [19, 20]. However, strengthening the guarantees to consider more general classes of attackers carries the potential danger of rendering them too rigid, i.e. violated by too many desirable publishing scenarios. Therefore, [8] simultaneously considers a relaxation along a different dimension: data owners are assumed willing to accept the privacy breach caused by an already published

set of views $\mathcal{V}$, but want to ensure that a new view $\mathcal{N}$ will cause no further breach. "Breach" is defined as a revision of belief from the a priori of having observed $\mathcal{V}(D)$ to the a posteriori of having also observed $\mathcal{N}(D)$.

In the terminology of Section 2.2, [8] introduces and studies precisely the various flavors of the $\text{NFBR}_{\mathcal{P},\mathcal{S}}$ guarantee: extent-dependent ($\text{NFBR}^{D}_{\mathcal{P},\mathcal{S}}$), and also extent-independent. Moreover, [8] argues that a privacy guarantee that holds for given D, $\mathcal{V}$, $\mathcal{S}$, and $\mathcal{N}$ may be violated if it is also known that D satisfies a set $\mathcal{C}$ of integrity constraints.

Example 8. Assume a hospital database consisting of four tables:

- PW associates patients with the ward they are in;
- WD associates doctors with the wards they are responsible for (several doctors may share responsibility for the same ward, and the same doctor may share responsibility for several wards);
- DA associates doctors with the ailments they treat;
- PA associates patients with the ailments they suffer from.

Assume that PW, WD, DA are published and PA is the secret. If the owner also discloses (or common sense leads the attacker to assume) the following integrity constraints, the attacker's belief can be affected.

- Patients can be treated only by doctors responsible for their ward.
- If a patient p suffers from an ailment a then some doctor treats p for a.

If these constraints do not hold, an attacker may consider a possible database associating a patient p with a doctor d who does not cover p's ward and hold a non-zero belief that p suffers from some ailment a treated only by d. However, under the constraints the secret patient-ailment association PA is a subset of $\Pi_{PA}(PW \bowtie WD \bowtie DA)$, to which (p, a) does not belong. This forces the attacker to revise to 0 his belief about any possible database witnessing (p, a).

[8] takes into account such semantic and integrity constraints when checking privacy.

Maybe the most interesting dimension of the study in [8] stems from proposing a natural way to classify attackers, yielding two groups.

First, we have the class of *all* attackers, described by set $\mathcal{P}_a$ of unrestricted distributions. Ideally, this is whom the owner wishes to defend against. $\mathcal{P}_a$ captures attackers who exploit correlations between tuples, and strictly includes attackers who don't (the ones described by the independent-tuple distributions of [19, 20]).

Second, [8] observes that the attacker is often unaware of (or uninterested in) the details of the possible database D witnessing a secret $\mathcal{S}(D)$, as D may also involve data that are tangential or irrelevant to the secret. For example, the attacker trying to link patients to their ailment does not care about the patient's insurance provider or the hospital's parking facilities, all of which could be also stored in the database.

[8] therefore considers attackers whose background knowledge enables them to form an opinion that discriminates among possible *secrets*, but who cannot (or do not care to) distinguish among the possible *databases* witnessing any given secret. In this survey we call such attackers *secret-focused*, and, given a secret $\mathcal{S}$, we denote with $\mathcal{P}_{\mathcal{S}}$ the set of distributions describing secret-focused attackers with respect to $\mathcal{S}$.

$\mathcal{P}_{\mathcal{S}}$ is defined as follows. Given a distribution $\delta_{\mathcal{S}}$ on possible secrets, we say that $\delta_{\mathcal{S}}$ induces a distribution δ on possible databases if δ satisfies both of the following conditions:

- for every s and every D such that $s = \mathcal{S}(D)$, we have $\sum_{s=\mathcal{S}(D')} \delta(D') = \delta_{\mathcal{S}}(s)$;
- all witnesses of the secret are equi-probable according to δ: $\forall D_1, D_2\ \mathcal{S}(D_1) = \mathcal{S}(D_2) \Rightarrow \delta(D_1) = \delta(D_2)$.

Observing that δ is uniquely determined by $\delta_{\mathcal{S}}$, we have that $\mathcal{P}_{\mathcal{S}}$ is the set of distributions on databases induced by all unrestricted distributions on secrets. Note that $\mathcal{P}_{\mathcal{S}}$ still allows for attackers with arbitrary capacity to discriminate among the secrets, as we start from arbitrary distributions on secrets.

[8] studies the setting in which the already published views $\mathcal{V}$, the secret $\mathcal{S}$, and the new view $\mathcal{N}$ are specified by unions of conjunctive queries with inequalities $\text{UCQ}^{\neq}$. The constraints in $\mathcal{C}$ are equivalent to containment statements between $\text{UCQ}^{\neq}$ queries. Such constraints extend classical embedded dependencies [1] with disjunction and inequalities, and can express such common integrity constraints as keys and foreign keys, functional, inclusion and join dependencies [1], cardinality constraints, and beyond.

For the extent-dependent guarantees, [8] shows that $\text{NFBR}^{D}_{\mathcal{P}_a,\mathcal{S}}(\mathcal{V},\mathcal{N})$ is Π^p_2-complete in the combined size of the queries and database, while $\text{NFBR}^{D}_{\mathcal{P}_{\mathcal{S}},\mathcal{S}}(\mathcal{V},\mathcal{N})$ is in PSPACE. These results hold even when the attacker knows that D satisfies a set $\mathcal{C}$ of constraints, as long as $\mathcal{C}$ is *weakly acyclic* [9, 10]. In addition, both extent-independent guarantees $\text{NFBR}_{\mathcal{P}_a,\mathcal{S}}(\mathcal{V},\mathcal{N})$ and $\text{NFBR}_{\mathcal{P}_{\mathcal{S}},\mathcal{S}}(\mathcal{V},\mathcal{N})$ are undecidable [8], even in the absence of constraints ($\mathcal{C} = \emptyset$).

These results should be viewed in light of the fact that in generalization-based publishing (discussed in Section 4), deciding whether an anonymization is optimal is NP-complete in the size of the database.

While the above results render the proposed privacy guarantees impractical in the current form, the study in [8] is a first step toward identifying restrictions leading to tractability on the views, secret and constraints. Moreover, the study proves that changing the class of attacker distributions yields a novel privacy guarantee, which is qualitatively different from the version in [19, 20], as witnessed by the different complexity and decidability bounds. Finally, the contrast between the various classes of attackers considered in [19, 20] and [8] shows the difficulty of striking the right balance between the strength of the guarantee and the feasibility of checking it.

4 Generalization-Based Publishing

The concept of anonymization by generalization [23, 24] was introduced to enable the publishing of data about individuals for the purpose of studies (e.g. computing statistics and data mining), while making it hard to pinpoint the exact individual associated with each data value. A canonical example pertains to a hospital that publishes seemingly anonymized data by releasing the age, gender and zip code of its patients together with the disease, in the hope that by leaving out the name and social security number attackers cannot infer who suffers from what disease.

Sweeney shows that this hope is unfounded [24], as over 85% of the US population is identified by the combination of age, gender and zip. This data is accessible to attackers either because they know the person, or simply from publicly available databases such as voter registration lists. In a notorious illustration of her point, Sweeney uncovered the medical history of a former governor of Massachusetts by combining the medical data with the registration list.

The attacks based on combining the anonymized data with external public databases are called *linking attacks*. Sweeney argues that in order to defend against linking attacks, the data owner must conservatively assume that the attacker has access to the public database, and that the information in this database uniquely identifies the individual. The upshot of this assumption is that the attacker has access to the identity of each individual, as if the owner had published it. Therefore, the best a defense against linking attacks can accomplish is to hide the *association* between the individual's identity and the sensitive data (such as her disease, salary, etc.).

In detail, work on anonymization by generalization considers a database containing a single relation $R(ID, QI, S)$, where

- the list of attributes ID comprises the person's identifier (e.g. (ssn) or (first name, middle name, last name)),
- the list of attributes QI gives the person's quasi-identifier (e.g. (age,gender,zip)) which can be used to look up the actual identifier in some public database of schema ID, QI, and
- S is the list of *sensitive* attributes (e.g. disease, salary, etc.).

Association between identity and sensitive attributes. We say that identity id is associated in R to sensitive attribute value s if there exists some tuple $r \in R$ with $r[ID] = id$ and $r[S] = s$.

Generalization function. To keep associations private, the owner anonymizes the QI attributes using a *generalization function* g. g hides the actual values of the QI attributes, replacing them with more general values. For instance, an age value is replaced by an age interval, a zip code changed by dropping some of its least significant digits. In the extreme, the generalization function can hide the attribute value completely by replacing it with the wild card "*". This is called attribute *suppression*.

Proprietary data

Name	Age	Gender	Zip	Ailment
John	20	M	92122	flu
Jane	22	F	92121	pneumonia
Jack	26	M	92093	cold
Jill	29	F	92094	bronchitis

Anonymized data

Age	Gender	Zip	Ailment
[20-25)	*	9212*	flu
[20-25)	*	9212*	pneumonia
[25-30)	*	9209*	cold
[25-30)	*	9209*	bronchitis

Fig. 1. Anonymization in Example 9

Anonymization. The generalization function g defines an anonymizing function $\mathcal{A}_g$ on R, which drops the ID attributes of each R-tuple, keeps the sensitive attributes unchanged, and substitutes the QI attributes with the result of g. If duplicates are created in this process, then they are all preserved. We have

$$\mathcal{A}_g(R) := \{\{t : QI, S |\ r \in R, t[QI] = g(r[QI]) \wedge t[S] = r[S]\}\},$$

where $t[X]$ denotes the projection of tuple t on attribute list X, and where $\{\{\}\}$ denote multi-set comprehensions (which preserve duplicates, as opposed to the set comprehensions denoted with $\{\}$).

Example 9. In Figure 1, the proprietary table R on the left has ID attribute *Name*, QI attributes *Age, Gender, Zip*, and S attribute *Ailment*. The table on the right is its anonymization $\mathcal{A}_g(R)$ where g replaces age with the 5-year interval it falls in, suppresses gender and hides the least significant digit of the zip code.

Given a tuple $r \in R$, the owner wishes to preserve the privacy of the association between the identifier $r[ID]$ and the sensitive attribute values $r[S]$. Since the sensitive attributes are published in clear, the attacker needs to guess only $r[ID]$. Intuitively, the anonymization $\mathcal{A}_g$ "hides the identity $r[ID]$ in a crowd" of possible identities, forcing the attacker to guess among them. The larger the crowd, the lower the chance of guessing right.

Equivalence under generalization. This crowd comprises the identities of all tuples whose projection on the quasi-identifiers generalizes under g to the same value. It is easy to see that the property of two tuples having the same image of their QI projection under g is an equivalence relation. Denoting with $[r]_g^R$ the equivalence class of r, we have

$$[r]_g^R := \{r' \in R \mid g(r'[QI]) = g(r[QI])\}.$$

In Example 9, the tuples of table R are partitioned by g into two equivalence classes, one comprising the tuples for John and Jane, the other the tuples for Jack and Jill.

Now consider a tuple $t \in \mathcal{A}_g(R)$ which is the image under $\mathcal{A}_g$ of some tuple $r \in R$. When the attacker observes the *occurrence* of sensitive attribute

value s in t ($t[S] = s$), the identities which could be associated with $t[S]$ in the actual database R are those of the tuples in r's equivalence class: $\{c : ID \mid r \in [r]_g^R, c[ID] = r[ID]\}$. In Example 9, the attacker concludes that either Jack or Jill can have bronchitis.

Assumptions on the attacker's knowledge. As introduced in [23, 24], the defense against linking attacks relies on a few implicit assumptions, also adopted by follow-up work. We explicitly list them below:

A1 For every $r \in R$, the attacker *knows* that $r[ID]$ occurs in the database (e.g. because $r[ID]$ identifies an acquaintance or celebrity whose hospitalization the attacker is aware of).

A2 For every $r \in R$, the attacker knows the value of the quasi-identifier attributes $r[QI]$ (e.g. due to access to some external public database).

A3 The attacker has no additional external knowledge to discriminate among the possible identities, thus treating them as equi-probable.

Util The owner is willing to live with the privacy breach caused by publishing the projection of R on S *in the clear*, since this is a minimal utility requirement for statistical and data mining computations performed by consumers of the released data.

Note that assumptions **A1** and **A2** are conservative, and any guarantee holding under them also defends against less informed attackers. In contrast, assumption **A3** is optimistic and weakens any guarantee, as it ignores attackers who improve their guessing odds by exploiting background knowledge to discriminate among alternatives. We address below versions of anonymity which relax this assumption. Finally, regarding assumption **Util**, note that [23] and most of its follow-up work concerns itself with choosing generalizations of the quasi-identifier attributes so as to minimize information loss, with the understanding that the sensitive data is released in the clear.

Relationship to GBP Model. We show the connection between the **GBP** model and the privacy guarantees offered by an arbitrary anonymization of a table via generalization. This will enable a comparison to the privacy guarantees described in Section 3. Moreover, it will allow us to contrast various anonymization guarantees found in the literature using a uniform framework.

- In typical studies of generalization, the proprietary database D consists of a single relation R of schema (ID, QI, S).
- Assumptions **A1** and **A2** can be modeled by just as well assuming that the owner (or some other authority) has already published the projection of R on ID, QI:
$$V_{id}(R) := \Pi_{ID,QI}(R).$$
- In our modeling, we separate the owner's concerns on releasing the sensitive data (none according to assumption **Util**) and the quasi-identifier data (serious concerns, calling for generalization). To this end, we consider the projection of R on the sensitive attributes S as good as published, by a view

$$V_s(R) := \{\{t : S \mid r \in R, t[S] = r[S]\}\}.$$

Note that V_s is defined under multi-set semantics (it preserves duplicates), thus revealing the distribution of sensitive values in the underlying population for the benefit of statistical studies.

In addition, the owner contemplates a new data release: the table R anonymized using publishing function $\mathcal{A}_g$ which associates anonymized quasi-identifiers with clear sensitive values. [3]

Under assumption **Util**, the owner is not concerned about the attacker's belief revision caused by seeing the sensitive values. The only revision she wishes to bound is caused by considering $\mathcal{A}_g(R)$ *on top of* $V_s(R)$. To this end, we adopt the following convention: *a priori* every attacker has access to views $V_{id}(R)$ and $V_s(R)$. We denote with $\mathcal{V}$ the publishing function given by the pair of views V_{id}, V_s. *A posteriori* refers to having released $\mathcal{A}_g(R)$ on top of $\mathcal{V}(R)$.

- For each proprietary tuple $r \in R$, both the identity value $r[ID]$ and the sensitive value $r[S]$ are known a priori to the attacker via views V_{id}, respectively V_s. The attacker is uncertain only about whether the two are associated in R. To hide this association from the attacker, the owner declares as secret the boolean query that checks the existence of some tuple $r' \in R$ which witnesses the association:

$$\mathcal{S}_r := \exists(r' \in R)\ r'[ID] = r[ID] \wedge r'[S] = r[S].$$

  Note that the secret does not include the quasi-identifier attributes, as by assumption **A2**, these are known for every identifier anyway (via V_{id}).
- Under assumption **A3**, the owner guards only against a single type of attackers, namely those who for lack of additional external knowledge deem all possible databases equally likely. We model these attackers by the *uniform probability distribution* u on possible databases.

Denote the multiplicity of sensitive value s in table X with $\mathrm{mult}(s, X)$. Then it is easy to verify that, under assumptions **A1**,**A2**, and **A3**, the probability that $id = r[ID]$ is associated to $s = r[S]$ in R (i.e. that secret $\mathcal{S}_r$ holds) is a priori (i.e. after seeing $\mathcal{V}(R)$) given by $\frac{\mathrm{mult}(s,R)}{|R|}$. The a posteriori probability (after seeing $\mathcal{A}_g(R)$) equals $\frac{\mathrm{mult}(s,[r]_g^R)}{|[r]_g^R|}$. It follows that g offers the following guarantee of bounded belief revision for secret $\mathcal{S}_r$:

$$\mathrm{BFBR}^R_{\{u\},\mathcal{S}_r}(\mathcal{V}, \mathcal{A}_g, |\frac{\mathrm{mult}(r[S],[r]_g^R)}{|[r]_g^R|} - \frac{\mathrm{mult}(r[S],R)}{|R|}|).$$

This immediately yields that the anonymization of R via g satisfies the following privacy guarantee:

[3] In practice, view $V_s(R)$ is released simultaneously with anonymized table $\mathcal{A}_g(R)$ (as its projection on S), not prior to it. Our modeling is merely a means to capture assumption **Util**.

$$\bigwedge_{r\in R} \mathrm{BFBR}^{R}_{\{u\},\mathcal{S}_r}(\mathcal{V},\mathcal{A}_g,|\frac{\mathrm{mult}(r[S],[r]^R_g)}{|[r]^R_g|}-\frac{\mathrm{mult}(r[S],R)}{|R|}|). \tag{5}$$

Note that the frequency of a sensitive value s in the entire table R can diverge widely from the frequency of s in the equivalence class of some $r \in R$. In a worst-case scenario when s is predominant in R (its frequency in R is close to 1) but very infrequent in r's equivalence class, the belief revision for secret $\mathcal{S}_r$ is considerably close to 1, which is the maximum possible.

4.1 K-Anonymity

In this section, we expose the connection between the original work on k-anonymity and the attacker's Bayesian belief revision. Casting the terminology of [23, 24] in terms of the **GBP** model, we find that [23, 24] bounds the attacker's belief revision by requiring the generalization function g to induce only equivalence classes of cardinality at least k. In that case, g is called *k-anonymous*, which we shall denote $\mathrm{anon}^R_k(g)$:

$$\mathrm{anon}^R_k(g) := \forall(r \in R)\ |[r]^R_g| \geq k.$$

For instance, function g in Example 9 is 2-anonymous.

By the above discussion, k-anonymity immediately implies that for a given *occurrence* of sensitive attribute value s in some tuple t of the anonymized data, there are at least k distinct identities which could be associated with s in the actual database R. Under assumptions **A1**,**A2**, and **A3**, the attacker's odds of guessing that indeed $r[ID]$ is the correct identity are at most $1/k$.

Previous work has interpreted this fact as implying that the probability of correctly guessing that identity *id* is associated in R to sensitive data *value* s is at most $1/k$. As pointed out in [16] and detailed below, this conclusion is unjustified: it is caused by the confusion between the *value* of the sensitive attributes and their *occurrence*. Specifically, if sensitive value s occurs l times in r's equivalence class, then the probability that $r[ID]$ is associated with value s is the sum over all occurrences of s of the probability that $r[ID]$ is associated with that occurrence, yielding $\frac{l}{|[r]^R_g|}$. This quantity can be arbitrarily larger than $\frac{1}{k}$, reaching 1 in the extreme case when all tuples in r's equivalence class have the same sensitive value. This observation gives an alternative explanation why k-anonymity provides no meaningful privacy guarantees in general.

Before discussing in the following sections refinements of k-anonymity which address this problem, we first articulate an implicit assumption under which k-anonymity does bound by $\frac{1}{k}$ the probability of guessing secret $\mathcal{S}_r$.

A4 For every $r \in R$, sensitive value $r[S]$ occurs only once in $[r]^R_g$.

We are now ready to relate the definition of k-anonymity with the **GBP** model. Under additional assumption **A4**, if g yields a k-anonymization of R then the a priori probability of $\mathcal{S}_r$ is $\frac{1}{|R|}$ and the a posteriori probability is $\frac{1}{|[r]_g^R|} \leq \frac{1}{k}$:

$$(\text{anon}_k^R(g) \wedge \mathbf{A4}) \Leftrightarrow \bigwedge_{r \in R} \text{BFBR}_{\{u\},\mathcal{S}_r}^R(\mathcal{V}, \mathcal{A}_g, \frac{1}{k} - \frac{1}{|R|}) \quad (6)$$

$$\Rightarrow \bigwedge_{r \in R} \text{BFBR}_{\{u\},\mathcal{S}_r}^R(\mathcal{V}, \mathcal{A}_g, \frac{1}{k}). \quad (7)$$

(7) states that under assumption **A4** the amount of belief revision for each secret $\mathcal{S}_r$ is bounded by a constant rather than the size of the database.

We discuss next a widely applicable guarantee that lifts restriction **A4**, relaxes restriction **A3**, and still bounds the amount of belief revision by an owner-defined constant.

4.2 L-Diversity

Machanavajjhala et al. [16] point out two key deficiencies of the k-anonymity guarantee: it does not withstand so-called *homogeneity* and *background* attacks.

In the general case when sensitive attribute values may occur more than once in R, vulnerability to homogeneity attacks arises whenever few sensitive values occur with high multiplicity in an equivalence class. In particular, when all tuples in r's equivalence class share the same sensitive value s, any attacker can infer with certainty that $r[ID]$ is associated with s. In this case, the attacker learns the maximum possible amount of information about the secret $\mathcal{S}_r$ since its a posteriori probability is 1.

In background attacks, the attacker exploits external background information to rule out a number of sensitive values as being definitely *not* associated to $r[ID]$. The remaining alternatives are considered equi-probable. This class of attackers is not covered by k-anonymity, which considers the single attacker who a priori deems all associations equi-probable.

[16] proposes the concept of *l-diversity* to remedy these deficiencies of k-anonymity. The intuition behind this concept is to defend against attackers who are able to rule out at most $l-1$ sensitive values from the equivalence class of each $r \in R$, by ensuring that the frequency of each sensitive value in the remaining set of tuples is upper bounded by an owner-defined threshold. [16] introduces the notion of *recursive* (c,l)*-diversity* as a sufficient condition for l-diversity.

For every $r \in R$, let o be the number of distinct sensitive values occurring in r's equivalence class. Let their list be $s_1, \ldots, s_o$, and let m_i be the multiplicity of s_i in r's equivalence class. Assuming w.l.o.g. that $m_1 \geq m_2 \geq \ldots \geq m_o$, we say that the equivalence class of r satisfies recursive (c,l)-diversity if

$$m_1 \leq c(m_l + m_{l+1} + \ldots + m_o)$$

for some constant c. We say that g satisfies recursive (c,l)-diversity for R, denoted r-div$_{c,l}(g,R)$, if for every $r \in R$, r's equivalence class satisfies recursive (c,l)-diversity.

Example 10. The anonymized table in Figure 1 satisfies recursive (1,2)-diversity.

Recursive (c,l)-diversity has two immediate implications.

First, it enables owners to drop assumption **A4**, thus extending applicability of the guarantee to tables with duplicate sensitive values. Indeed, it is easy to check that under assumptions **A1**, **A2** and **A3**, (c,l)-diversity imposes an upper bound of $\frac{c}{1+c}$ on the attacker's a posteriori and a priori belief, and hence on the belief revision that $\mathcal{S}_r$ holds. Recursive (c,l)-diversity thus provides defense even when assumption **A4** is violated.

Second, recursive (c,l)-diversity allows to relax assumption **A3** to accommodate defense against background attacks. [16] shows that this guarantee implies that regardless of which (at most) $l-1$ sensitive values are pruned from r's equivalence class as being unassociated to $r[ID]$ (according to background information), the frequency of each remaining sensitive value in the pruned equivalence class is at most $\frac{c}{1+c}$. This is the upper bound on the a posteriori belief about secret $\mathcal{S}_r$.

[17] discusses additional refinements of (c,l)-diversity, relaxing the definition to allow for the disclosure of attributes for certain individuals with less stringent privacy concerns. The authors also show that l-diversity is a practical notion, not only because it defends against more realistic attacks than k-anonymity, but also because finding an optimal l-diverse generalization of a table can be done no less efficiently than finding an optimal k-anonymization. Machanavajjhala et al. show how to exploit the structural similarity of the two privacy notions to easily adapt to l-diversity the state-of-the-art techniques developed for k-anonymity, such as the Incognito algorithm [12].

In the remainder of this section, we connect l-diversity to the **GBP** model.

Relationship to the GBP Model. The insight that when assumption **A4** does not hold K-anonymity provides no guarantees, is also reflected in the **GBP** model. Specifically, in the pathological case when all tuples in r's equivalence class share the same sensitive value, the posterior probability of $\mathcal{S}_r$ is given by

$$\mathbf{P}_u[\mathcal{S}_r | \mathcal{V}(R) \wedge \mathcal{A}_g(R)] = \frac{\text{mult}(r[S], [r]_g^R)}{|[r]_g^R|} = 1$$

so from (5) we obtain that the only guarantee possible for $\mathcal{S}_r$ is

$$\text{BFBR}^R_{\{u\},\mathcal{S}_r}(\mathcal{V}, \mathcal{A}_g, 1 - \frac{\text{mult}(r[S], R)}{|R|}).$$

This is a trivial guarantee, satisfied by any anonymization, including those in which the secret $\mathcal{S}_r$ is completely exposed.

In contrast, it is easily verified that, even after dropping assumption **A4**, recursive (c, l)-diversity guarantees that

$$\frac{\text{mult}(r[S], R)}{|R|} \leq \frac{\text{mult}(r[S], [r]_g^R)}{|[r]_g^R|} \leq \frac{c}{1+c}$$

which implies that the further belief revision is bounded by $\frac{c}{1+c}$. Plugging this bound into (5), we obtain

$$\text{r-div}_{c,l}^R(g) \Rightarrow \bigwedge_{r \in R} \text{BFBR}_{\{u\},\mathcal{S}_r}^R(\mathcal{V}, \mathcal{A}_g, \frac{c}{1+c}).$$

A remarkable fact about recursive (c, l)-diversity is that it represents the first anonymity flavor that looks beyond the uninformed attacker described by the uniform probability distribution. The class of attackers it considers can be described by the following family of probability distributions. We say that a probability distribution δ is *l-pruning* if it satisfies both conditions below:

- for every $r \in R$, there is a set V_r of sensitive values occurring in $[r]_g^R$, such that
  - $|V_r| < l$ and
  - for every database R', $\delta(R') = 0$ if and only if there are $r' \in R$ and $v \in V_{r'}$ such that R' contains the association of $r'[ID]$ with v;
- all databases with non-zero probability are equi-probable.

Intuitively, V_r is the set of alternatives which the attacker rules out as unassociated to $r[ID]$. Denoting with $\mathcal{LP}$ all l-pruning distributions given by R and g, we have

$$\text{r-div}_{c,l}^R(g) \Rightarrow \bigwedge_{r \in R} \text{BFBR}_{\mathcal{LP},\mathcal{S}_r}^R(\mathcal{V}, \mathcal{A}_g, \frac{c}{1+c}).$$

Since $\mathcal{LP}$ is generated by all possible choices of V_r, the guarantee defends against all attackers able to rule out at most $l-1$ alternatives, no matter which these alternatives are, as dictated by the various attackers' backgrounds.

We conclude this section with a few remarks.

4.3 Additional Remarks on Anonymization Techniques

Complexity of Finding Optimal Anonymizations. Clearly one extreme way to ensure k-anonymity is to generalize tuples into a single equivalence class. This would of course minimize the utility of the released data. [18] studies the problem of finding the k-anonymization which incurs the least amount of data loss due to generalization (for various metric for data loss), showing that the problem of optimal k-anonymization is NP-complete. Several follow-up papers propose practical k-anonymization algorithms based on approximations and heuristics [12, 3, 7, 4]. While Machanavajjhala et al. do

not provide a lower bound for finding optimal l-diverse anonymizations, they conjecture NP-hardness as well, and show how to adapt the Incognito Algorithm [12].

Sensitive Data Generalization. There are slight exceptions from assumption **Util**: an example occurs in [22]. In this work, sensitive data is not published in the clear, but generalized itself using a function f. The generalization function f exploits a hierarchy among concepts in the sensitive domain, treating ancestor concepts as more general than descendant concepts. For instance, instead of displaying "pneumonia", the owner may release a more general concept such as "respiratory tract problems" which in turn is generalized by "antibiotic-curable ailment". Evidently the objective in [22] is to minimize the information loss resulting from generalization of both quasi-identifiers and sensitive attributes. We can capture this scenario as well in the **GBP** model, by simply adjusting assumption **Util** to state that the owner is willing to live with the attacker's belief after seeing the generalized sensitive values described by view $V_s(R) := f(\Pi_S(R))$.

T-Closeness. One paper that explicitly states and exploits assumption **Util** is [14]. It considers the probability distribution p on the secrets $\{\mathcal{S}_r\}_{r\in R}$ after seeing the entire anonymized table $\mathcal{A}_g(R)$, and the probability distribution q of the sensitive values in R, i.e. in $V_s(R)$. The authors introduce the privacy guarantee of *t-closeness*, which holds if the *distribution distance* between p and q is smaller than a parameter threshold t. The authors show shortcomings of standard metrics for comparing distributions and propose their own. They also show that the search for a t-close anonymization that maximizes utility (under a standard measure) can be performed by adapting efficient algorithms developed for k-anonymity. However, t-closeness does not subsume k-anonymity and the authors suggest combining the two before releasing an anonymized table.

An Alternative Bayesian Modeling. [17] compares the notion of l-diversity to a model called *Bayesian Optimal Privacy (BOP)* model. Just like the **GBP** model, the BOP model is based on belief revision. However, the authors conclude a mismatch between l-diversity and the BOP model. As demonstrated in this section, the reason is not due to any fundamental mismatch between Bayesian privacy models and l-diversity. Rather, it stems from the particular modeling choice in [17] which ignores assumption **Util**: [17] considers that a priori the attacker sees $V_{id}(R)$ but not $V_s(R)$. The difficulty with this modeling (identified in [17] as well) is that to estimate the attacker's a priori belief revision about $\mathcal{S}_r$, we require knowledge of the attacker's probability distribution on the domain of all sensitive values, which is an unrealistic expectation. The modeling we describe in this section surmounts this obstacle, as under assumption **Util**, it needn't care about this distribution; it only considers belief revision starting from the attacker's adjusted belief *after seeing* $V_s(R)$. We can estimate this belief (as in (5)), regardless of the belief before seeing $V_s(R)$.

work	attacker classes considered
[8]	all $\mathcal{P}_a$; secret-focused $\mathcal{P}_{\mathcal{S}}$
[19, 20]	independent-tuple $\mathcal{P}_{it}$
[16, 17]	l-pruning $\mathcal{LP}$
[23, 24]	uniform distribution $\mathcal{P}_u = \{u\}$

$$\mathcal{P}_u \subset \mathcal{LP} \subset \begin{matrix}\mathcal{P}_{\mathcal{S}} \\ \mathcal{P}_{it}\end{matrix} \subset \mathcal{P}_a$$

Fig. 2. Classes of attackers considered by privacy guarantees in various works

k-Anonymous Views. An intriguing idea introduced by Jajodia et al in [25] is to apply the notion of k-anonymity to view-based publishing. The setting is similar to generalization-based publishing: we have a single table R with identity attributes ID and sensitive attributes S. The owner publishes data from R via views expressed as conjunctive queries. It is assumed that releasing all identifiers $\Pi_{ID}(R)$ and all sensitive attributes $\Pi_S(R)$ is acceptable to the owner, but releasing the *association* between them is not.

A view V is said to satisfy k-anonymity if for every identifier $id \in \Pi_{ID}(R)$, there are k distinct possible databases $\{R_1, \ldots, R_k\} \subseteq [R]_V$, each associating id with a distinct sensitive value $s_1, \ldots, s_k$.

This guarantee can be connected to the **GBP** model as follows. Say that an attacker is *uniform secret-focused* if he is described by a distribution on databases which is generated by a uniform distribution on secrets. Given secret $\mathcal{S}$, there is only one such uniform secret-focused distribution, $\delta_{\mathcal{S}}$. Then view V's k-anonymity implies

$$\bigwedge_{r \in R} \mathrm{BFBR}^R_{\{\delta_{\mathcal{S}_r}\},\mathcal{S}_r}(\mathcal{V}, V, \frac{1}{k}).$$

where $\mathcal{V}$ are the views (considered a priori known to the attacker) $\Pi_{ID}(R)$ and $\Pi_S(R)$, and $\mathcal{S}_r$ is the secret association for tuple r, as defined in Section 4.1.

5 View-Based Versus Generalization-Based Publishing

The formalization of various privacy guarantees in terms of the **GBP** model allows us to qualitatively compare view-based and generalization-based privacy guarantees.

Abstracting from the different expressive powers of the publishing functions $\mathcal{V}$ and $\mathcal{N}$ (views versus generalizations), the fundamental difference between these guarantees remains the class of probability distributions used to model attackers.

The guarantee in [8] is the most conservative one, considering all types of attackers (with the drawback of high complexity for deciding the extent-dependent guarantees, and undecidability in the extent-independent case).

Miklau and Suciu's guarantee of perfect privacy considers a subclass of attackers described by independent-tuple distributions, with the benefit of featuring better decision complexity. Recursive (c, l)-diversity requires l-pruning distributions, which are a subclass of the distributions of [8]. L-pruning distributions are also particular cases of independent-tuple distributions. Finally, the uniform distribution u implicitly used to model attackers in k-anonymity is a particular case of l-pruning distributions (for $l = 1$). Figure 2 summarizes the relationship between the various classes of attackers.

Note that the classes $\mathcal{P}_a, \mathcal{P}_{\mathcal{S}}, \mathcal{P}_{it}$ were introduced for view-based privacy, while $\mathcal{LP}$ and $\mathcal{P}_u$ for generalization-based privacy. There is no reason why the various classes of attackers should not be considered uniformly, across both publishing paradigms.

6 Privacy in Open-World Integration

So far we have only considered publishing settings in which $\mathcal{V}$ is a function. However, this modeling leaves out an important publishing paradigm, namely open-world integration [11, 13].

In open-world integration, a collection L of data sources (also known as local databases) is registered into an integrated database G (also known as the global database). Each data source is registered by stating the inclusion of a publishable data subset into G. The publishable subset is typically specified by a query against the local database, and the global dataset containing it is specified by a query against the global database. This allows for instance a Toyota car dealer to register the classified deals in her database as a subset of the Toyota deals from the global database of a portal covering many dealerships. If the portal offers several brands, specifying its Toyota deals requires a selection query.

Such inclusion statements do not uniquely determine the global database, since whenever a global database G satisfies them, so does any other database strictly containing the tuples in G. Consequently, the relation $\mathcal{V}$ between local (proprietary) and global (public) database is not functional: $\mathcal{V}$ associates any extent of local databases L to an infinite family of global databases. Towards a well-defined semantics of answering application queries Q against the global schema, the notion of *certain answers* was introduced [11, 13]. Given a set L of local databases, the certain answer of Q against the global schema is the set of all tuples appearing in the answer of Q on all global databases G related to L: $cert_Q(L) = \cap_{(L,G)\in\mathcal{V}} Q(G)$.

Clients (and therefore attackers) can interact with the integration system only by posing queries against the global schema and receiving their certain answer. In such a setting, it still makes sense to allow the owner of an individual local database to specify the sensitive data using a query $\mathcal{S}$ against the local database. Privacy of the secret can still be defined in terms of no

(or bounded) belief revision, which depends on the possible local databases, analogous to the **GBP** model.

However, the possible local databases now represent precisely those which are indistinguishable from the actual local database by an arbitrary interaction with the integration system. That is, they cannot be distinguished by posing arbitrary-length sequences of arbitrary queries against the global schema and observing their certain answer.

The problem is that the space of possible interactions between attacker and integration system is infinite, so this definition does not immediately lead to an algorithm for identifying the set of possible local databases, which in turn hinders the development of an algorithm for checking privacy guarantees.

[21] solves the problem in a setting where $\mathcal{V}$ is given by containment statements between a union of conjunctive queries with inequalities ($\text{UCQ}^{\neq}$) against the local data and a $\text{UCQ}^{\neq}$ query against the global data (such statements are also known as GLAV [11, 13] or source-target constraints [10]). The secret $\mathcal{S}$ is also given by a $\text{UCQ}^{\neq}$ query against the local database. [21] shows that, instead of considering the infinitely many possible interactions of an attacker with the integration system, it suffices to focus on a single, canonically built interaction. This canonical interaction is optimal in the sense that it poses a finite set of queries against the integration system, such that no further queries an attacker could conceive give additional information. More precisely, the certain answers of the canonical queries suffice to reverse-engineer precisely the set of possible local databases. This in turn enables formulating and checking all extent-dependent **GBP** privacy guarantees (Section 2).

7 Conclusions

In this chapter, we reduced various instantiations of the view-based and generalization-based publishing to the **GBP** model, also showing how to apply it to publishing in open-world integration. This reduction offers a unifying perspective on various seemingly disparate privacy guarantees developed independently for the various publishing paradigms.

We have applied the **GBP** model to settings in which the publishing transformation is deterministically defined as either a function or a relation. This assumption leaves out the mature line of research on preserving privacy by randomizing the data (see for instance [2] and references within).

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. Charu C. Aggarwal. On randomization, public information and the curse of dimensionality. In *International Conference on Data Engineering (ICDE)*, pages 136–145, 2007.

3. G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *International Conference on Database Theory (ICDT)*, pages 246–258, 2005.
4. C. C. Aggrawal. On k-anonymity and the curse of dimensionality. In *International Conference on Very Large Data Bases (VLDB)*, pages 901–909, 2005.
5. Francois Bancilhon and Nicolas Spyratos. Protection of information in relational data bases. In *International Conference on Very Large Data Bases (VLDB)*, pages 494–500, 1977.
6. Francois Bancilhon and Nicolas Spyratos. Algebraic versus probabilistic independence in data bases. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 149–153, 1985.
7. R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *International Conference on Data Engineering (ICDE)*, pages 217–228, 2005.
8. Alin Deutsch and Yannis Papakonstantinou. Privacy in database publishing. In *International Conference on Database Theory (ICDT)*, pages 230–245, 2005.
9. Alin Deutsch and Val Tannen. Reformulation of XML queries and constraints. In *International Conference on Database Theory (ICDT)*, 2003.
10. R. Fagin, P. Kolaitis, R. Miller, and L. Popa. Data exchange: Semantics and query answering. In *International Conference on Database Theory (ICDT)*, 2003.
11. Alon Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.
12. K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM Conference on Management of Data (SIGMOD)*, pages 49–60, 2005.
13. Maurizio Lenzerini. Data integration: A theoretical perspective. In *ACM Symposium on Principles of Database Systems (PODS)*, 2002.
14. Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *International Conference on Data Engineering (ICDE)*, 2007.
15. Ashwin Machanavajjhala and Johannes Gehrke. On the efficiency of checking perfect privacy. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 163–172, 2006.
16. Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkatasubramaniam. l-diversity: Privacy beyond k-anonymity. In *International Conference on Data Engineering (ICDE)*, page 24, 2006.
17. Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *To appear in IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
18. A. Meyerson and R.Williams. On the complexity of optimal k-anonymity. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 223–228, 2004.
19. Gerome Miklau and Dan Suciu. A formal analysis of information disclosure in data exchange. In *ACM Conference on Management of Data (SIGMOD)*, pages 575–586, 2004.
20. Gerome Miklau and Dan Suciu. A formal analysis of information disclosure in data exchange. *Journal of Computer and Systems Sciences*, 73(3):507–534, 2007.
21. Alan Nash and Alin Deutsch. Privacy in GLAV information integration. In *International Conference on Database Theory (ICDT)*, pages 89–103, 2007.

22. Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.
23. Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In *ACM Symposium on Principles of Database Systems (PODS)*, page 188, 1998.
24. Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness, and Knowlege-Based Systems*, 10(5):557–570, 2002.
25. Chao Yao, Xiaoyang Sean Wang, and Sushil Jajodia. Checking for k-anonymity violation by views. In *International Conference on Very Large Data Bases (VLDB)*, pages 910–921, 2005.