Secure Anonymization for Incremental Datasets

Ji-Won Byun[1], Yonglak Sohn[2], Elisa Bertino[1], and Ninghui Li[1]

[1] CERIAS and Computer Science, Purdue University, USA
`{byunj,bertino,ninghui}@cs.purdue.edu`,
[2] Computer Engineering, Seokyeong University, Korea
`syl@skuniv.ac.kr`

Abstract. Data anonymization techniques based on the k-anonymity model have been the focus of intense research in the last few years. Although the k-anonymity model and the related techniques provide valuable solutions to data privacy, current solutions are limited only to the static data release (i.e., the entire dataset is assumed to be available at the time of release). While this may be acceptable in some applications, today we see databases continuously growing everyday and even every hour. In such dynamic environments, the current techniques may suffer from poor data quality and/or vulnerability to inference. In this paper, we analyze various inference channels that may exist in multiple anonymized datasets and discuss how to avoid such inferences. We then present an approach to securely anonymizing a continuously growing dataset in an efficient manner while assuring high data quality.

1 Introduction

A model on which recent privacy-protecting techniques often rely is the *k-anonymity* model [17]. In the k-anonymity model, privacy is guaranteed by ensuring that any record in a released dataset be indistinguishable (with respect to a set of attributes, called *quasi-identifier*) from at least $(k-1)$ other records in the dataset. Thus, in the k-anonymity model the risk of re-identification is maintained under an acceptable probability (i.e., $1/k$). Another interesting protection model addressing data privacy is the ℓ-diversity model [12]. The ℓ-diversity model assumes that a private dataset contains some sensitive attribute(s) which cannot be modified. Such a sensitive attribute is then considered disclosed when the association between a sensitive attribute value and a particular individual can be inferred with a significant probability. In order to prevent such inferences, the ℓ-diversity model requires that every group of indistinguishable records contains at least ℓ distinct sensitive attribute values; thereby the risk of attribute disclosure is kept under $1/\ell$.

Although the k-anonymity and ℓ-diversity models have led to a number of valuable privacy-protecting techniques [2, 6, 7, 9, 10, 16], the existing solutions are limited only to static data release. That is, in such solutions it is assumed that the entire dataset is available at the time of release. This assumption implies a significant shortcoming, as data today are continuously collected (thus continuously grow) and there is a strong demand for up-to-date data at all times. For instance, suppose that a hospital wants to publish its patient records for medical researchers. Surely, all the published records

AGE	Gender	Diagnosis
21	Male	Asthma
23	Male	Flu
52	Male	Alzheimer
57	Female	Diabetes

Fig. 1. Initial patient records

AGE	Gender	Diagnosis
$[21-25]$	Male	Asthma
$[21-25]$	Male	Flu
$[50-60]$	Person	Alzheimer
$[50-60]$	Person	Diabetes

Fig. 2. 2-diverse patient records

AGE	Gender	Diagnosis
21	Male	Asthma
23	Male	Flu
52	Male	Alzheimer
57	Female	Diabetes
27	*Female*	*Cancer*
53	*Male*	*Heart Disease*
59	*Female*	*Flu*

Fig. 3. New patient records

AGE	Gender	Diagnosis
$[21-30]$	Person	Asthma
$[21-30]$	Person	Flu
$[21-30]$	Person	Cancer
$[51-55]$	Male	Alzheimer
$[51-55]$	Male	Heart Disease
$[56-60]$	Female	Flu
$[56-60]$	Female	Diabetes

Fig. 4. New 2-diverse patient records

must be properly anonymized in order to protect patients' privacy. At first glance, the task seems reasonably straightforward, as any of the existing anonymization techniques can anonymize the records before they are published. The challenge is, however, that as new records are frequently created (e.g., whenever new patients are admitted), the hospital needs a way to provide up-to-date information to researchers in timely manner.

One possible approach is to anonymize and publish new records periodically. Then researchers can either study each released dataset independently or merge multiple datasets together for more comprehensive analysis. Although straightforward, this approach may suffer from severely low data quality. The key problem is that small sets of records are anonymized independently; thus, records may have to be modified much more than when they are anonymized all together. Thus, in terms of data quality, this approach is highly undesirable.

A better approach is to anonymize and publish the entire dataset whenever the dataset is augmented with new records. In this way, researchers are always provided with up-to-date information. Although this can be easily accomplished using existing techniques (i.e., by anonymizing the entire dataset every time), there are two significant drawbacks. First, it requires redundant computation, as the entire dataset has to be anonymized even if only a few records are newly inserted. Another, much more critical, drawback is that even though published datasets are securely anonymous independently (i.e., each dataset is k-anonymous or ℓ-diverse), they could be vulnerable to inferences when observed collectively. In the following section, we illustrate such inferences.

1.1 Examples of Inferences

A hospital initially has a dataset in Fig. 1 and publishes its 2-diverse version shown in Fig. 2. As previously discussed, in an ℓ-diverse dataset the probability of attribute disclosure is kept under $1/\ell$. For example, even if an attacker knows that the record of

Tom, who is a 21-year-old male, is in the published dataset, he cannot be sure about Tom's disease with greater than $1/2$ probability (although he learns that Tom has either asthma or flu). At a later time, three more patient records (shown in *Italic*) are inserted into the dataset, resulting the dataset in Fig. 3. The hospital then publishes a new 2-diverse version in Fig. 4. Observe that Tom's privacy is still protected in the newly published dataset. However, not every patient is protected from the attacker.

Example 1. Suppose the attacker knows that Alice, who is in her late twenties, has recently been admitted to the hospital. Thus, he knows that Alice is not in the old dataset in Fig. 2, but in the new dataset in Fig. 4. From the new dataset, he learns only that Alice has one of $\{$Asthma, Flu, Cancer$\}$. However, by consulting the previous dataset, he can easily infer that Alice has neither asthma nor flu. He concludes that Alice has cancer.

Example 2. The attacker knows that Bob is 52 years old and has long been treated in the hospital. Thus, he is sure that Bob's record is in both datasets. First, by studying the old dataset, he learns that Bob suffers from either alzheimer or diabetes. Now the attacker checks the new dataset and learns that Bob has either alzheimer or heart disease. He thus concludes that Bob suffers from alzheimer. Note that three other records in the new dataset are also vulnerable to similar inferences.

1.2 Contributions and paper outline

As shown in the previous section, anonymizing datasets statically (i.e., without considering previously released datasets) may lead to various inferences. In this paper, we present an approach to securely anonymizing a continuously growing dataset in an efficient manner while assuring high data quality. The key idea underlying our approach is that one can efficiently anonymize a current dataset by directly inserting new records to the previously anonymized dataset. This implies, of course, that both new records and anonymized records may have to be modified, as the resulting dataset must satisfy the imposed privacy requirements (e.g., k-anonymity or ℓ-diversity). Moreover, such modifications must be cautiously made as they may lead to poor data quality and/or enable undesirable inferences. We thus describe several inference attacks where attacker tries to undermine the imposed privacy protection by comparing a multiple number of anonymized datasets. We analyze various inference channels that attacker may exploit and discuss how to avoid such inferences. In order to address the issue of data quality, we introduce a data quality metric, called *Information Loss* (IL) metric, which measures the amount of data distortion caused by generalization. Based on our analysis on inference channels and IL metric, we develop an algorithm that securely and efficiently inserts new records into an anonymized dataset while assuring high data quality.

The remainder of this paper is organized as follows. We review the basic concepts of the k-anonymity and ℓ-diversity models in Section 2. In Section 3, we describe several inference attacks and discuss possible inference channels and how to prevent such inferences. Then we describe our algorithm that securely and efficiently anonymizes datasets in Section 4 and evaluate our techniques in Section 5. We review some related work in Section 6 and conclude our discussion in Section 7.

2 k-Anonymity and ℓ-diversity

The k-anonymity model assumes that person-specific data are stored in a table (or a relation) of columns (or attributes) and rows (or records). The process of anonymizing such a table starts with removing all the explicit identifiers, such as name and SSN, from it. However, even though a table is free of explicit identifiers, some of the remaining attributes in combination could be specific enough to identify individuals. For example, as shown by Sweeney [17], 87% of individuals in the United States can be uniquely identified by a set of attributes such as $\{$ZIP, gender, date of birth$\}$. This implies that each attribute alone may not be specific enough to identify individuals, but a particular group of attributes could be. Thus, disclosing such attributes, called *quasi-identifier*, may enable potential adversaries to link records with the corresponding individuals.

Definition 1. (**Quasi-identifier**) A quasi-identifier of table T, denoted as Q_T, is a set of attributes in T that can be potentially used to link a record in T to a real-world identity with a significant probability. $\square$

The main objective of the k-anonymity problem is thus to transform a table so that no one can make high-probability associations between records in the table and the corresponding entity instances by using quasi-identifier.

Definition 2. (**k-anonymity requirement**) Table T is said to be k-*anonymous* with respect to quasi-identifier Q_T if and only if for every record r in T there exist at least $(k-1)$ other records in T that are indistinguishable from r with respect to Q_T. $\square$

By enforcing the k-anonymity requirement, it is guaranteed that even though an adversary knows that a k-anonymous table T contains the record of a particular individual and also knows the quasi-identifier value of the individual, he cannot determine which record in T corresponds to the individual with a probability greater than $1/k$. The k-anonymity requirement is typically enforced through generalization, where real values are replaced with "less specific but semantically consistent values" [17]. Given a domain, there are various ways to generalize the values in the domain. Commonly, numeric values are generalized into intervals (e.g., [12$-$19]), and categorical values into a set of distinct values (e.g., $\{$USA, Canada$\}$) or a single value that represents such a set (e.g., North-America). A group of records that are indistinguishable from each other is often referred to as an *equivalence class*.

Although often ignored in most k-anonymity techniques, a private dataset typically contains some sensitive attribute(s) that are not quasi-identifier attributes. For instance, in the patient records in Fig. 3, *Diagnosis* is considered a sensitive attribute. For such datasets, the key consideration of anonymization is the protection of individuals' sensitive attributes. Observe, however, that the k-anonymity model does not provide sufficient security in this particular setting, as it is possible to infer certain individuals' attributes without precisely re-identifying their records. For instance, consider a k-anonymized table where all records in an equivalence class have the same sensitive attribute value. Although none of these records can be matched with the corresponding individuals, their sensitive attribute value can be inferred with a probability of 1. Recently, Machanavajjhala et al. [12] pointed out such inference issues in the k-anonymity model and proposed the notion of ℓ-*diversity*.

Definition 3. (**ℓ-diversity requirement**) Table T is said to be *ℓ-diverse* if records in each equivalence class have at least ℓ distinct sensitive attribute values. □

As the ℓ-diversity requirement ensures that every equivalence class contains at least ℓ distinct sensitive attribute values, the risk of attribute disclosure is kept under $1/\ell$. Note that the ℓ-diversity requirement also ensures ℓ-anonymity, as the size of every equivalence class must be greater than equal to ℓ.

3 Incremental Data Release and Inferences

In this section, we first describe our assumptions on datasets and their releases. We then discuss possible inference channels that may exist among multiple data releases and present requirements for preventing such inferences.

3.1 Incremental data release

We assume that a private table T, which contains a set of quasi-identifier attributes Q_T and a sensitive attribute S_T, stores person-specific records, and that only its ℓ-diverse[3] version $\widehat{T}$ is released to public. As more data are collected, new records are inserted into T, and $\widehat{T}$ is updated and released periodically to reflect the changes of T. Thus, users, including potential attackers, are allowed to read a sequence of ℓ-diverse tables, $\widehat{T}_0, \widehat{T}_1, \ldots$, where $|\widehat{T}_i| < |\widehat{T}_j|$ for $i < j$. As previously discussed, this type of data release is necessary to ensure high data quality in anonymized datasets.

As every released table is ℓ-diverse, by observing each table independently, one cannot gain more information than what is allowed. That is, the risk of attribute disclosure in each table is at most $1/\ell$. However, as shown in Section 1, it is possible that one can increase the probability of attribute disclosure by observing changes made to the released tables. For instance, if one can be sure that two (anonymized) records in two different versions indeed correspond to the same individual, then he may be able to use this knowledge to infer more information than what is allowed by the ℓ-diversity protection.

Definition 4. (**Inference channel**) Let $\widehat{T}_i$ and $\widehat{T}_j$ be two ℓ-diverse versions of a private table T. We say that there exists an inference channel between $\widehat{T}_i$ and $\widehat{T}_j$, denoted as $\widehat{T}_i \rightleftharpoons \widehat{T}_j$, if observing $\widehat{T}_i$ and $\widehat{T}_j$ together increases the probability of attribute disclosure in either $\widehat{T}_i$ or $\widehat{T}_j$ to a probability greater than $1/\ell$. □

Thus, for a data provider, it is critical to ensure that there is no inference channel among the released tables. In other words, the data provider must make sure that a new anonymized table to be released does not create any inference channel with respect to the previously released tables.

Definition 5. (**Inference-free data release**) Let $\widehat{T}_0, \ldots, \widehat{T}_n$ be a sequence of previously released tables, each of which is ℓ-diverse. A new ℓ-diverse table $\widehat{T}_{n+1}$ is said to be *inference-free* if and only if $\nexists\, \widehat{T}_i, i = 1, \ldots, n,$ s.t. $\widehat{T}_i \rightleftharpoons \widehat{T}_{n+1}$. □

---

[3] Although we focus on ℓ-diverse data in this paper, one can easily extend our discussion to k-anonymous data.

3.2 Inference attacks

We first describe a potential attacker and illustrate how the attacker may discover inference channels among multiple anonymized tables. We then describe various inference channels and discuss how to prevent them.

Attacker's knowledge. Before discussing possible inference channels, we first describe a potential attacker. We assume that the attacker has been keeping track of all the released tables; he thus possesses a set of released tables $\{\widehat{T}_0, \ldots, \widehat{T}_n\}$, where $\widehat{T}_i$ is a table released at time i. We also assume that the attacker has the knowledge of who is and who is not contained in each table. This may seem to be too farfetched at first glance, but such knowledge is not always hard to acquire. For instance, consider medical records released by a hospital. Although the attacker may not be aware of all the patients, he may know when target individuals (in whom he is interested) are admitted to the hospital. Based on this knowledge, the attacker can easily deduce which tables include such individuals and which tables do not. Another, perhaps the worst, possibility is that the attacker may collude with an insider who has access to detailed information about the patients; e.g., the attacker could obtains a list of patients from a registration staff. Thus, it is reasonable to assume that the attacker's knowledge includes the list of individuals contained in each table as well as their quasi-identifier values. However, as all the released tables are ℓ-diverse, the attacker cannot infer the individuals' sensitive attribute values with a significant probability. That is, the probability that an individual with a certain quasi-identifier has a particular sensitive attribute is bound to $1/\ell$; $P(S_T = s | Q_T = q) \leq 1/\ell$. Therefore, the goal of the attacker is to increase this probability of attribute disclosure (i.e., above $1/\ell$) by comparing the released tables all together.

Comparing anonymized tables. Let us suppose that the attacker wants to know the sensitive attribute of a particular individual, say Tom, whose quasi-identifier value is q. There are two types of comparisons that may help the attacker: 1) comparison of table $\widehat{T}_i$ that does not contain Tom and table $\widehat{T}_j$ that does, and 2) comparison of $\widehat{T}_i$ and $\widehat{T}_j$, that both contain Tom. In both cases, $i < j$. Let us call these types $\delta(\neg\widehat{T}_i, \widehat{T}_j)$ and $\delta(\widehat{T}_i, \widehat{T}_j)$, respectively. Note that in either case the attacker only needs to look at the records that may relate to Tom. For instance, if Tom is a 57 years old, then records such as $\langle [10-20], Female, Flu \rangle$ would not help the attacker much. In order to find records that may help, the attacker first finds from $\widehat{T}_i$ an equivalence class e_i, where $q \subseteq e_i[Q_T]$. In the case of $\delta(\neg\widehat{T}_i, \widehat{T}_j)$, the attacker knows that Tom's record is not in e_i; thus, none of the records in e_i corresponds to Tom. Although such information may not seem useful, it could help the attacker as he may be able to eliminate such records when he looks for Tom's record from $\widehat{T}_j$. In the case of $\delta(\widehat{T}_i, \widehat{T}_j)$, however, the attacker knows that one of the records in e_i must be Tom's. Although he cannot identify Tom's record or infer his sensitive attribute at this point (as e_i must contain at least ℓ number of records that are all indistinguishable to each other and also at least ℓ number of distinct sensitive attribute values), this could be useful information when he examines $\widehat{T}_j$.

After obtaining e_i, the attacker needs to identify in $\widehat{T}_j$ the records that possibly correspond to the records in e_i, that is, equivalence class(es) that are *compatible* to e_i.

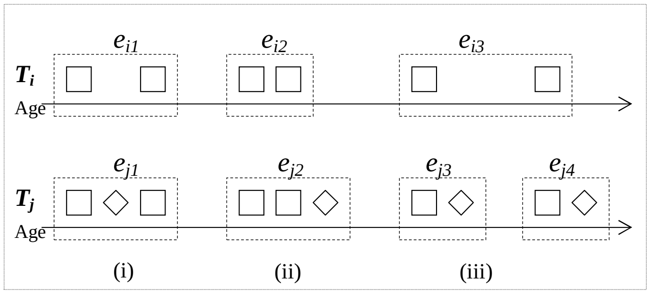


Fig. 5. Compatible equivalence classes

Definition 6. (**Compatibility**) Let $Q = \{q_1, \ldots, q_m\}$ be a set of quasi-identifier attributes. Let $e[q_i]$ be the q_i-value of an equivalence class e, where $q_i \in Q$. We say that two equivalence classes e_a and e_b are *compatible* with respect to Q if and only if any of the following conditions holds.

1. $\forall q_i \in Q,\ e_a[q_i] = e_b[q_i]$: the quasi-identifer values of e_a and e_b are identical to each other; we denote it as $e_a \cong e_b$.
2. $e_a \ncong e_b$ and $\forall q_i \in Q,\ e_a[q_i] \subseteq e_b[q_i]$: the quasi-identifer value of e_b is a more generalized form of the quasi-identifier of e_a; we denote it as $e_a \prec e_b$.
3. $e_a \ncong e_b$, $e_a \not\prec e_b$, and $\forall q_i \in Q,\ e_a[q_i] \cap e_b[q_i] \neq \emptyset$: the quasi-identifier values of e_a and e_b overlap with each other; we denote it as $e_a \Bumpeq e_b$. $\Box$

Example 3. Consider Fig. 5, where the records of two tables T_i and T_j are spatially represented along the dimension of the quasi-identifier, Age. For simplicity, we do not show their sensitive attribute values. Table T_i contains six records (shown as '$\Box$'), and its 2-diverse version, $\widehat{T_i}$, consists of three equivalence classes, e_{i1}, e_{i2}, and e_{i3}. On the other hand, table T_j contains four additional records (shown as '$\Diamond$'), and its 2-diverse version, $\widehat{T_j}$, consists of four equivalence classes, e_{j1}, e_{j2}, e_{j3}, and e_{j4}. Given $\widehat{T_i}$ and $\widehat{T_j}$, the following compatible equivalences can be found.

1. $e_{i1} \cong e_{j1}$ (Fig. 5 (i))
2. $e_{i2} \prec e_{j2}$ (Fig. 5 (ii))
3. $e_{i3} \Bumpeq e_{j3}$ and $e_{i3} \Bumpeq e_{j4}$ (Fig. 5 (iii))

The fact that two equivalence classes are compatible implies that there exist some records present in both equivalence classes, although their quasi-identifiers may have been modified differently. In what follows, we show how matching such records between compatible equivalence classes could enable the attacker to make high probability inferences.

Inference channels between compatible equivalence classes. As previously discussed, there are three cases of compatible equivalence classes. We now examine these cases in conjunction with each of $\delta(\neg\widehat{T_i}, \widehat{T_j})$ and $\delta(\widehat{T_i}, \widehat{T_j})$, illustrating how the attacker may infer Tom's sensitive attribute, s_T.

	$e_i \cong e_j$	$e_i \prec e_j$	$e_i \Bumpeq e_{j1}$ and $e_i \Bumpeq e_{j2}$
$\delta(\neg\widehat{T}_i, \widehat{T}_j)$	$e_j[S] \setminus e_i[S]$	$e_j[S] \setminus e_i[S]$	$((e_{j1}[S] \cup e_{j2}[S]) \setminus e_i[S]) \cap e_{jk}[S],\ k = 1, 2$
$\delta(\widehat{T}_i, \widehat{T}_j)$	$\emptyset$	$\emptyset$	$e_i[S] \cap e_{jk}[S],\ k = 1, 2$

Fig. 6. Summary of inference-enabling sets

1. $e_i \cong e_j$ or $e_i \prec e_j$: In these cases, the attacker can reason that all the records in e_i must also appear in e_j, and the attacker only needs to look at the sensitive attribute values. Let $e_i[S]$ and $e_j[S]$ be the multisets (i.e., duplicate-preserving sets[4]) of sensitive attribute values in e_i and e_j, respectively.
   (a) In the case of $\delta(\neg\widehat{T}_i, \widehat{T}_j)$, the attacker knows that Tom's sensitive attribute value is not in $e_i[S]$, but in $e_j[S]$; i.e., $s_T \notin e_i[S]$ and $s_T \in e_j[S]$. As he knows that all the values in $e_i[S]$ must also appear in $e_j[S]$, he can conclude that $s_T \in (e_j[S] \setminus e_i[S])$. Therefore, the attacker can infer s_T with a probability greater than $1/\ell$ if $(e_j[S] \setminus e_i[S])$ contains less than ℓ number of distinct values.
   (b) In the case of $\delta(\widehat{T}_i, \widehat{T}_j)$, $s_T \in e_i[S]$ and $s_T \in e_j[S]$. However, as both sets are ℓ-diverse, the attacker does not gain any additional information on s_T.
2. $e_i \Bumpeq e_{j1}$ and $e_i \Bumpeq e_{j2}$[5]: In this case, the attacker reasons that the records in e_i must appear in either e_{j1} or e_{j2}. Moreover, as the attacker knows Tom's quasi-identifier is q, he can easily determine which of e_{j1} and e_{j2} contains Tom's record. Let us suppose e_{j1} contains Tom's record; i.e., $q \subseteq e_{j1}[Q_T]$. Let $e_i[S]$, $e_{j1}[S]$, and $e_{j2}[S]$ be the multisets of sensitive attribute values in e_i, e_{j1}, and e_{j2}, respectively.
   (a) In the case of $\delta(\neg\widehat{T}_i, \widehat{T}_j)$, the attacker knows that Tom's sensitive attribute value is included in neither $e_i[S]$ nor $e_{j2}[S]$, but in $e_{j1}[S]$; i.e., $s_T \notin e_i[S]$, $s_T \notin e_{j2}[S]$, and $s_T \in e_{j1}[S]$. Note that unlike the previous cases, he cannot simply conclude that $s_T \in (e_{j1}[S] \setminus e_i[S])$, as not all the records in e_i are in e_{j1}. However, it is true that Tom's record is in $e_{j1} \cup e_{j2}$, but not in e_i; thus $s_T \in (e_{j1}[S] \cup e_{j2}[S]) \setminus e_i[S]$. As Tom's record must be in e_{j1}, the attacker can finally conclude that $s_T \in ((e_{j1}[S] \cup e_{j2}[S]) \setminus e_i[S]) \cap e_{j1}[S]$. Therefore, if this set does not contain at least ℓ distinct values, the attacker can infer s_T with a probability greater than $1/\ell$.
   (b) In the case of $\delta(\widehat{T}_i, \widehat{T}_j)$, the attacker knows that Tom's sensitive attribute value appears in both $e_i[S]$ and $e_{j1}[S]$. Based on this knowledge, he can conclude that $(s_T \in e_i[S] \cap e_{j1}[S])$. Thus, attacker can infer s_T with a probability greater than $1/\ell$ if $(e_i[S] \cap e_{j1}[S])$ contains less than ℓ distinct values.

We summarize our discussion on possible inference-enabling sets in Fig. 6. Intuitively, a simple strategy that prevents any inference is to ensure that such sets are all ℓ-diverse. Note that with current static anonymization techniques, this could be a daunting task as inference channels may exist in every equivalence class and also with respect to every previously released dataset. In the following section, we address this issue by developing an efficient approach to preventing inferences during data anonymization.

---

[4] Therefore, set operations (e.g., $\cap$, $\cup$, and $\setminus$) used in our discussion are also multiset operations.

[5] It is possible that $\widehat{T}_j$ contains more than two equivalence classes that are compatible to e_i. However, we consider two compatible equivalence classes here for simplicity.

4 Secure Anonymization

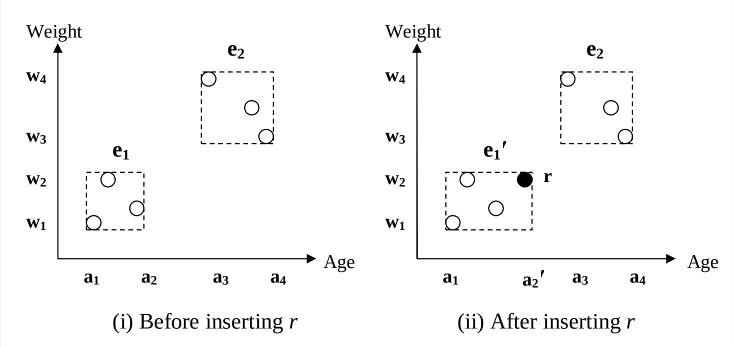
In this section, we present an approach to securely anonymizing a dataset based on previously released datasets. We first describe a simple ℓ-diversity algorithm and propose a noble quality metric that measures the amount of data distortion in generalized data. Based on the algorithm and the quality metric, we then develop an approach where new records are selectively inserted to a previously anonymized dataset while preventing any inference.

4.1 ℓ-diversity algorithm and data quality

Data anonymization can be considered a special type of optimization problem where the cost of data modification must be minimized (i.e., the quality of data must be maximized) while respecting anonymity constraints (e.g., k-anonymity or ℓ-diversity). Thus, the key components of anonymization technique include generalization strategy and data quality metric.

ℓ-diversity algorithm. In [12], Machanavajjhala et al. propose an ℓ-diversity algorithm by extending the k-anonymity algorithm in [9] to ensure that every equivalence class is ℓ-diverse. In this paper, we present a slightly different algorithm which extends the *multidimensional* approach described in [10]. The advantage of the multidimensional approach is that generalizations are not restricted by pre-defined generalization hierarchies (DGH) and thus more flexible. Specifically, the algorithm consists of the following two steps. The first step is to find a partitioning scheme of the d-dimensional space, where d is the number of attributes in the quasi-identifier, such that each partition contains a group of records with at least ℓ number of distinct sensitive attribute values. In order to find such a partitioning, the algorithm recursively splits a partition at the median value (of a selected dimension) until no more split is allowed with respect to the ℓ-diversity requirement. Then the records in each partition are generalized so that they all share the same quasi-identifier value, thereby forming an equivalence class. Compared to the technique based on DGH in [12], this multidimensional approach allows finer-grained search and thus often leads to better data quality.

Data quality metric. The other key issue is how to measure the quality of anonymized datasets. To date, several data quality metrics have been proposed for k-anonymous datasets [2, 6, 10, 7, 16]. Among them, Discernibility Metric (DM) [2] and Average Equivalence Class Size Metric [10] there are two data quality metrics that do not depend on generalization hierarchies. Intuitively, DM measures the effect of k-anonymization process by measuring how much records are indistinguishable from each other. However, DM does not consider the actual transformation of data values. For instance, suppose that there are more than k records that already have the identical quasi-identifer value and that they are all in the same equivalence class. Even though these records are not generalized at all, DM penalizes each of these un-generalized records. The same issue arises for the average equivalence class size metric, which measures the quality of anonymization directly based on the size of the equivalence classes.



(i) Before inserting *r* (ii) After inserting *r*

Fig. 7. Generalization and data distortion

To address this shortcoming, we propose a data quality metric that captures the amount of data distortion by measuring the expansion of each equivalence class (i.e., the geometrical size of each partition). For instance, consider Fig. 7 (i), where the records are spatially represented in 2-dimensional space for quasi-identifier, $\{$*Age*, *Weight*$\}$. In the figure, the dotted regions group the records into two 3-diverse equivalence classes, e_1 and e_2. Note that as all the records in an equivalence class are modified to share the same quasi-identifer, each region indeed represents the generalized quasi-identifier of the records contained in the region. For instance, the generalized records in e_1 may share the identical quasi-identifier $\langle [a_1 - a_2], [w_1 - w_2] \rangle$. Thus, data distortion can be measured naturally by the size of the regions covered by equivalence classes. Based on this idea, we now define a new data quality metric, referred to as *Information Loss* metric (IL) as follows.

Definition 7. (**Information loss**) Let $e = \{r_1, \ldots, r_n\}$ be an equivalence class where $Q_T = \{a_1, \ldots, a_m\}$. Then the amount of data distortion occurred by generalizing e, denoted by $IL(e)$, is defined as:

$$\text{IL}(e) = |e| \times \sum_{j=1,\ldots,m} \frac{|G_j|}{|D_j|}$$

where $|e|$ is the number of records in e, and $|D_j|$ represents the domain size of attribute a_j. $|G_j|$ represents the amount of generalization in attribute a_j (e.g., the length of an interval which contains all the attribute values existing in e. ☐

4.2 Updates of anonymized datasets

As previously described, our goal is to produce an up-to-date anonymized dataset by inserting new records into a previously anonymized dataset. Note that in our discussion below, we assume that all ℓ-diverse tables are maintained internally as partitioned, but unmodified tables. That is, each ℓ-diverse table consists of a set of equivalent groups, $\{e_1, \ldots, e_m\}$, which contain un-generalized records. This is a practical assumption as generating actual ℓ-diverse records for publication from a partitioned table is a relatively simple task. Consequently, given such a partitioned table and a new set of records, our insertion algorithm produces a new partitioned table which includes the new records.

Suppose that an anonymized table $\widehat{T}$, which is an ℓ-diverse version of a private table T, has been published. Suppose that at a later time, a new set of records $R = \{r_1, \ldots, r_n\}$ has been inserted into T. Let us denote the updated T as T'. Intuitively, a new ℓ-diverse version $\widehat{T}'$ can be generated by inserting R into $\widehat{T}$. The key requirements for such insertions are: 1) $\widehat{T}'$ must be ℓ-diverse, 2) the data quality of $\widehat{T}'$ should be maintained as high as possible, and 3) $\widehat{T}'$ must be inference-free.

We now briefly describe such an insertion algorithm which ensures the first two requirements. A key idea is to insert a record into a "closest" equivalence class so that the necessary generalization is minimized. For instance, let us revisit Fig. 7, which (i) depicts six records partitioned into two 3-diverse equivalence classes, and (ii) shows revised equivalence classes after record r is inserted. Observe that as r is inserted into e_1 resulting in e'_1, the information loss of the dataset is increased by $IL(e'_i) - IL(e_i)$. However, if r were inserted into e_2, then the increase of the information would have been much greater. Based on this idea, we devise an insertion algorithm that ensures high data quality as follows.

1. **(Add)** If a group of records in R forms an ℓ-diverse equivalence class which does not overlap with any of existing equivalence classes, then we can simply *add* such records to $\widehat{T}$ as a new equivalence class.
2. **(Insert)** The records which cannot be added as a new equivalence class must be *inserted* into some existing equivalence classes. In order to minimize the data distortion in $\widehat{T}'$, each record r_i is inserted into equivalent group e_j in $\widehat{T}$ which minimizes $IL(e_j \cup \{r_i\}) - IL(e_j)$.
3. **(Split)** After adding or inserting all the records in R into $\widehat{T}$, it is possible that the number of distinct values in some equivalence class exceeds 2ℓ. If such an equivalence class exists, then we may be able to *split* it into two separate equivalence classes for better data quality. Note that splitting an equivalence class may or may not be possible, depending on how the records are distributed in the equivalence class.

Clearly, the algorithm above do not consider the possibility of inference channels at all. In the following section, we enhance this algorithm further to ensure that an updated dataset does not create any inference channel.

4.3 Preventing inference channels

In Section 3, we discuss that in order to prevent any inference channel, all the inference-enabling sets (see Fig. 6) must be ℓ-diverse. We now discuss how to enhance our unsecure insertion algorithm to ensure such sets are all ℓ-diverse. Specifically, we examine each of three major operations, *add*, *insert*, and *split*, and describe necessary techniques to achieve inference-free updates.

Clearly, the add operation does not introduce any inference channel, as it only adds new ℓ-diverse equivalence classes that are not compatible to any previously released equivalence class. However, the insert operation may introduce inference channels (i.e., $e_j[S] \setminus e_i[S]$). That is, if the new records inserted into an equivalence class contain less than ℓ number of distinct sensitive values, then the equivalence class becomes vulnerable to inference attacks through $\delta(\neg\widehat{T}_i, \widehat{T}_j)$. Thus, such insertions must not be allowed. In order to address this issue, we modify the insertion operation as follows. During the insertion phase, instead of inserting records directly to equivalence classes, we insert records into the waiting-lists of equivalence classes. Apparently, the records in a waiting-list can be actually inserted into the corresponding equivalence class if they are ℓ-diverse by themselves; until then, they are suppressed from the anonymized dataset. Note that as more records are continuously inserted into the table (and into the waiting lists), for most records, the waiting period is not significant. However, to expedite the waiting period, we also check if the records in the waiting lists can be added as an independent equivalence class which does not overlap with any other existing equivalence class.

There are two inference channels that may be introduced when an equivalence class e_i is split into e_{ji} and e_{j2}. The first possibility is: $((e_{j1}[S] \cup e_{j2}[S]) \setminus e_i[S]) \cap e_{jk}[S],\ k = 1, 2$. Clearly, if such sets are not ℓ-diverse, then they become vulnerable to inference attacks through $\delta(\neg\widehat{T}_i, \widehat{T}_j)$. Thus, the condition must be checked before splitting e_i. The other possible inference channel is: $e_i[S] \cap e_{jk}[S],\ k = 1, 2$. This implies that if there are not enough overlapping sensitive values between the original equivalence class and each of the split equivalence classes, then split equivalence classes become vulnerable to inference attacks through $\delta(\widehat{T}_i, \widehat{T}_j)$. Thus, unless such condition is satisfied, e_i must not be split. The tricky issue in this case is, however, that inference channels may exist between any of the compatible equivalence classes that were previously released. For instance, if there exists equivalence class e'_i that was released before e_i, then the splitting condition must be satisfied with respect to e'_i as well. This means that the system needs to maintain the information about all the previous releases. In order to facilitate this, we store such information for each equivalence class; that is, each equivalence class keeps the information about its previous states. Note that it does not require a huge storage overhead, as we need to keep only the information about the sensitive attribute (not all the records). We also purge such information when any previous equivalence class becomes no longer compatible to the current equivalence class.

Clearly, inference preventing mechanisms may decrease the quality of anonymized data. Although it is a drawback, it is also the price to pay for better data privacy.

5 Experimental Results

In this section, we describe our experimental settings and report the results in details.

Experimental setup. The experiments were performed on a 2.66 GHz Intel IV processor machine with 1 GB of RAM. The operating system on the machine was Microsoft Windows XP Professional Edition, and the implementation was built and run in Java 2 Platform, Standard Edition 5.0. For our experiments, we used the Adult dataset from the UC Irvine Machine Learning Repository [15], which is considered a de facto benchmark for evaluating the performance of anonymization algorithms. Before the experiments, the Adult data set was prepared as described in [2, 7, 10]. We removed records with missing values and retained only nine of the original attributes. In our experiments,

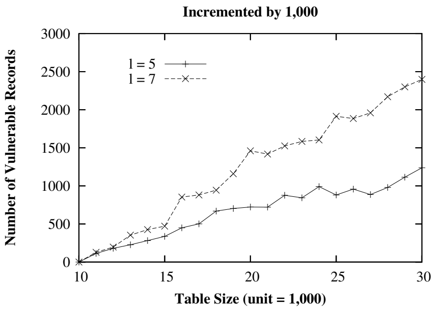


Fig. 8. Vulnerabilities

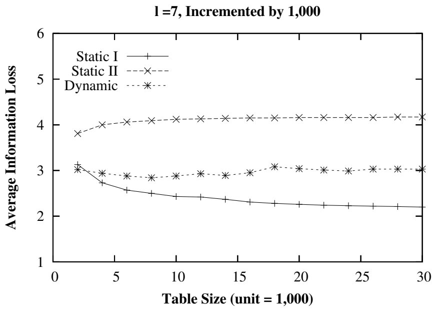


Fig. 9. Data quality

we considered $\{$*age*, *work class*, *marital status*, *occupation*, *race*, *gender*, *native country*, *salary*$\}$ as the quasi-identifier, and *education* attribute as the sensitive attribute.

For the experiments, we implemented three different ℓ-diversity approaches: *Static I*, *Static II*, and *Dynamic*. Static I is an approach where the entire dataset is anonymized whenever new records are inserted, while Static II anonymizes new records independently and merges the result with the previously anonymized dataset. Dynamic implements our approach, where new records are directly inserted into the previously anonymized dataset while preventing inference channels.

Vulnerability. The first question we investigated was how vulnerable datasets were to inferences when they were statically anonymized (i.e., Static I). In the experiment, we first anonymized 10K records and generated the first "published" dataset. We then generated twenty more subsequent datasets by anonymizing 1,000 more records each time. Thus, we had the total of twenty-one ℓ-diverse datasets with different sizes ranging from 10K to 30K. After obtaining the datasets, we examined the inference-enabling sets existing between the datasets. For instance, we examined the inference-enabling sets of the 12K-sized dataset with respect to the 10K- and 11K-sized datasets. Whenever we found an inference channel, we counted how many records were vulnerable by it. Fig. 8 shows the results where $\ell = 5, 7$. As expected, more records become vulnerable to inferences as the size of dataset gets larger; for the 30K-sized dataset, about 8.3% of records are vulnerable to inferences. Note that there were no vulnerable records in datasets generated by Static II and Dynamic.

Data Quality. Next, we compared the data quality resulted by Static I, Static II, and Dynamic. For each approach, we generated different sizes of ℓ-diverse datasets, ranging from 1K to 30K, as previously described. For the data quality measure, we used the average cost of IL metric (described in Section 4.1). That is, the quality of an anonymized dataset $\widehat{T}$ was computed as: $\sum_{e \in \mathcal{E}} IL(e) \;/\; |\widehat{T}|$, where $\mathcal{E}$ is a set of all equivalence classes in $\widehat{T}$. Intuitively, this measure indicates the degree to which each of record is generalized. Our experiment results are shown in Fig. 9. Although Dynamic results in lower data quality when compared to Static I, it produces much higher quality data than Static II. Moreover, the quality is maintained regardless of the data size. Fig. 10 shows the number of suppressed records in Dynamic approach. Note that each number shows

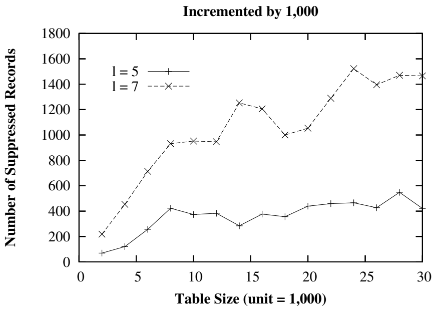


Fig. 10. Suppression

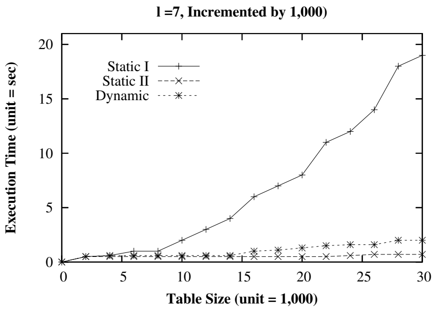


Fig. 11. Execution Time

the total number of suppressed records with respect to the entire dataset. For instance, when $\ell = 5$, only 421 records needed to be suppressed for the 30K-sized dataset.

Execution Time. Fig. 11 shows the execution times of anonymizing various sizes of datasets. As shown, the execution time of Static I increases linearly with respect to the size of the dataset, while Static II and Dynamic produce anonymized datasets almost instantly.

6 Related Work

In this section, we briefly survey existing literature that addresses data privacy. Instead of providing a comprehensive survey, we discuss various aspects of data privacy. Note that we do not include the k-anonymity or ℓ-diversity work here as detailed discussion can be found in Section 2.

Ensuring privacy in published data has been a difficult problem for a long time, and this problem has been studied in various aspects. In [8], Lambert provides informative discussion on the risk and harm of undesirable disclosures and discusses how to evaluate a dataset in terms of these risk and harm. In [3], Dalenius poses the problem of re-identification in (supposedly) anonymous census records and firstly introduces the notion of “quasi-identifier”. He also suggests some ideas such as suppression or encryption of data as possible solutions.

Data privacy has been extensively addressed in statistical databases, which primarily aim at preventing various inference channels. One of the common techniques is data perturbation [11, 14, 18], which mostly involves swapping data values or introducing noise to the dataset. While the perturbation is applied in a manner which preserves statistical characteristics of the original data, the transformed dataset is useful only for statistical research. Another important technique is query restriction [4, 5], which restricts queries that may result in inference. In this approach, queries are restricted by various criteria such as query-set-size, query-history, and partitions. Although this approach can be effective, it requires the protected data to remain in a dedicated database at all time.

7 Conclusions

In this paper, we presented an approach to securely anonymizing a continuously growing dataset in an efficient manner while assuring high data quality. In particular, we described several inference attacks where attacker tries to undermine the imposed privacy protection by comparing a multiple number of anonymized datasets. We analyzed various inference channels and discussed how to avoid such inferences. We also introduced Information Loss (IL) metric, which measures the amount of data distortion caused by generalization. Based on the discussion on inference channels and IL metric, we then developed an algorithm that securely and efficiently inserts new records into an anonymized dataset while assuring high data quality.

References

1. G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Anonymizing tables. In *Proceedings of the 10th International Conference on Database Theory*, pages 246–258, 2005.
2. R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *the 21st International Conference on Data Engineering*, 2005.
3. T. Dalenius. Finding a needle in a haystack. *Journal of Official Statistics*, 2, 1986.
4. D. Dobkin, A. K. Jones, and R. J. Lipton. Secure databases: Protection against user influence. *ACM Transactions on Database systems*, 4, 1979.
5. I. P. Fellegi. On the question of statistical confidentiality. *Journal of the American Statistical Association*, 1972.
6. B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *the 21st International Conference on Data Engineering*, 2005.
7. V. S. Iyengar. Transforming data to satisfy privacy constraints. In *ACM Conference on Knowledge Discovery and Data mining*, 2002.
8. D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9, 1993.
9. K. LeFevre, D. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM International Conference on Management of Data*, 2005.
10. K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *the 22nd International Conference on Data Engineering*, 2006.
11. C. K. Liew, U. J. Choi, and C. J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10, 1985.
12. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *the 22nd International Conference on Data Engineering*, 2006.
13. A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2004.
14. S. P. Reiss. Practical data-swapping: The first steps. *ACM Transactions on Database Systems*, 9, 1980.
15. C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.
16. L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.
17. L. Sweeney. K-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.
18. J. F. Traub and Y. Y. H. Wozniakowski. The statistical security of statistical database. *ACM Transactions on Database Systems*, 9, 1984.