Journal of Biomedical Informatics xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Small sum privacy and large sum utility in data publishing

Ada Wai-Chee Fu [a,*], Ke Wang [b], Raymond Chi-Wing Wong [c], Jia Wang [d], Minhao Jiang [c]

[a] Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong
[b] Department of Computer Science, Simon Fraser University, Canada
[c] Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong
[d] Department of Computer Science, University of Illinois at Urbana-Champaign, USA

ARTICLE INFO

Article history:
Received 4 August 2013
Accepted 1 April 2014
Available online xxxx

Keywords:
Privacy preserving data publishing
Inference attacks
Privacy versus utility

ABSTRACT

While the study of privacy preserving data publishing has drawn a lot of interest, some recent work has shown that existing mechanisms do not limit all inferences about individuals. This paper is a positive note in response to this finding. We point out that not all inference attacks should be countered, in contrast to all existing works known to us, and based on this we propose a model called *SPLU*. This model protects sensitive information, by which we refer to answers for aggregate queries with small sums, while queries with large sums are answered with higher accuracy. Using SPLU, we introduce a sanitization algorithm to protect data while maintaining high data utility for queries with large sums. Empirical results show that our method behaves as desired.

1. Introduction

In a recent work by Cormode [7], it is shown that despite much progress in two main branches of privacy models for data publishing, namely differential privacy [13], and various *syntactic methods* such as k-anonymity [26] and ℓ-diversity [20], inference-based attacks can still be successful. The study is based on the ability of an attacker to construct accurate classifiers on top of releases protected by state-of-the-art privacy preserving data publishing techniques.

The empirical study result above is in fact consistent with the result from [10]. Following the model in [10], given a dataset $d = (d_1, \ldots, d_n) \in \{0,1\}^n$, a query q is a subset of $\{1, 2, \ldots, n\}$, and its true answer $a_q = \sum_{i \in q} d_i$. Hence, the query q determines a subset of d, and the answer for q is the number of entries in the subset. Given algorithm $\mathcal{A}$ for query response, we say that $\mathcal{A}(q)$ is within ϵ perturbation if it deviates from the true answer by no more than ϵ. $\mathcal{A}$ is within ϵ perturbation if $\mathcal{A}(q)$ is within ϵ perturbation for all q. If an adversary can reconstruct with time complexity $t(n)$ the entire database very accurately, then the database $\mathcal{D} = (d, \mathcal{A})$ is said to be $t(n)$-non-private. The following theorem from [10] says that any privacy preserving algorithm renders the database useless, and conversely utility in the published data implies privacy breach.

Theorem 1 [10]. *Let $\mathcal{D} = (d, \mathcal{A})$ be a database where $\mathcal{A}$ is within $o(\sqrt{n})$ perturbation then $\mathcal{D}$ is poly (n)-non-private.*

The above findings are based on the assumption that all inference attacks are to be defended, and any relatively accurate information derivable from the published data is considered privacy breaching. This is quite inconsistent with the simultaneous requirement of utility whereby minimum distortion is to be introduced so that the published data are as close to the original data as possible. Here we show that the dilemma can be resolved by a segregation of utility and privacy.

The key point as observed by Cormode is that privacy and utility are closely related. As stated in the conclusion in [7], "release of (anonymized) data may reveal hitherto unknown population parameters which compromise individual privacy. ... in some settings, these population statistics may represent exactly the desired utility of the data collection and publication." This remark highlights the issue to be resolved. The key is how to differentiate between utility and privacy. Once we identify the utility of the data and once users agree that this utility has no conflict with their privacy, the proper solution is not to insist on protection for the information related to the utility. We provide a way to differentiate what concepts may be reasonable to be disclosed for utility. If users indeed have concerns about the disclosure of such concepts there is always the option of not releasing any data. This provides for a better alternative for the status quo of releasing the data knowing that certain inference attacks are possible.

To our knowledge, there is no known model for the separation of concepts that need protection and those that need to be

* Corresponding author. Fax: +852 2603 5024.
E-mail addresses: adafu@cse.cuhk.edu.hk (A.W.-C. Fu), wangk@cs.sfu.ca (K. Wang), raywong@cse.ust.hk (R.C.-W. Wong), jwang@cse.cuhk.edu.hk (J. Wang), minhaojiang@gmail.com (M. Jiang).

http://dx.doi.org/10.1016/j.jbi.2014.04.002
1532-0464/

Fig. 1. The *SPLU*-Gen system for data sanitization and query processing.

maintained for utility purposes, in privacy preserving data publishing. Some previous works [11,19] study the adjustment of parameters in the anonymization process for the trade-off between privacy and utility. The problem studied in such works is very different and in their model, all concepts are treated equally in terms of utility and privacy. We assume that aggregate queries of large sums should be answered relatively accurately for utility, while those with very small sums should not. Consider an example from [13] where a dataset D' tells us that almost everyone involved in a dataset is two footed. Knowing with high certainty that an individual is two footed from D' is not considered a privacy issue since it is true for almost everyone in the dataset.[1] Large sum concepts are statistical and of value for utility. In contrast, small count concepts are non-statistical and the protection of small counts has been well-studied in the topic of security in statistical databases [1].

Our main contributions are summarized as follows.

(1) We propose a framework, called SPLU, which allows releasing data for answering large sum queries with high accuracy to provide utility, while offering high inaccuracy for small sum queries in order to ensure privacy. We point out that not all inference attacks should be defended.
(2) To demonstrate the feasibility of the concept of SPLU, we propose a data sanitization mechanism, called *SPLU*-Gen, for achieving this goal. *SPLU*-Gen is based on randomized perturbation on the sensitive values.
(3) We introduce a sophisticated reconstruction algorithm which takes into account the global data distribution. This improves on the known reconstruction approach in syntactic methods and leads to higher data utility.
(4) We have conducted experiments on two real datasets to show that *SPLU*-Gen provides protection for small sums and high utility for large sum queries. We note that existing mechanisms may readily support SPLU, which is an encouraging result.

In Fig. 1, we outline the *SPLU*-Gen mechanism for data sanitization and the query processing based on the sanitized data. The dataset on the left of the figure is passed as input to *SPLU*-Gen. The input is processed and as a result, a sanitized dataset is published. Querying is applied on the sanitized data, and the query result is generated by a reconstruction algorithm. The user will receive relatively accurate results for large sum queries and inaccurate results for queries of small sums.

The rest of the paper is organized as follows. Section 2 presents the SPLU model. Section 3 describes the mechanism *SPLU*-Gen. Section 4 is about count reconstruction and properties of *SPLU*-Gen. Section 5 considers multiple attribute aggregations. Section 6 is on empirical study, and Section 7 is on related works. Section 8 concludes this work.

2. SPLU model

We consider the data model in previous works on k-anonymity [26] and ℓ-diversity [20]. This data model assumes that a set of attributes form a quasi-identifier, the values of which for a target individual can be known to the adversary from other sources, and also one or more sensitive attributes which need to be protected. Hence, there are two kinds of attributes in the dataset, the non-sensitive attributes (*NSA*) and the sensitive attributes (*SA*). In Fig. 2(a) we show a given dataset D. In table D, the attribute *id* is for the tuple id. The attributes Age and Zip-Code are considered non-sensitive attributes and they form a quasi-identifer. The term quasi-identifier indicates that it may be possible to identify an individual based on the respective attribute values. For example, it is possible that Age 90 and Zip-Code [12–17 k] uniquely determine an individual, if there is only one resident aged 90 in the area of Zip-Code [12–17 k]. Such attributes are considered not sensitive. In table D, Disease is a sensitive attribute.

In this model we do not perturb the non-sensitive values but may alter the sensitive values to ensure privacy. This is a commonly used data model and it corresponds to the initial problem settings with real world applications [25,22].

We are given a dataset (table) D which is a set of N tuples that follow the above data model. A concept c in D is a predicator formed by the conjunction of value assignments to a set of attributes in D. Our problem is how to generate and replace the sensitive values for the tuples in D to be published in the output dataset D'. D' should satisfy both utility for large sum querying and privacy protection for small sum queries.

In Fig. 1(b), we show a possible published dataset D', which is a sanitized counterpart of dataset D. We shall discuss in Section 3 about how D' is generated from D.

We define the requirements of our model in the following.

Given a dataset D, an anonymized data set D' generated by sanitization mechanism $\mathcal{A}$, and a concept c involving $s \in SA$, let f_c be the true frequency of c in D and f'_c be the estimated frequency of c from D'.

Definition 1 (*large sum utility*). Concept c has a $(\epsilon, \mathcal{T}_E, \mathcal{T}_f)$ utility guarantee if

$$Pr\left[|f'_c - f_c| \geqslant \varepsilon f_c\right] \leqslant \mathcal{T}_E \text{ for } f_c \geqslant \mathcal{T}_f \quad (1)$$

The above definition says that a concept c has a $(\epsilon, \mathcal{T}_E, \mathcal{T}_f)$ guarantee if whenever the frequency f_c of c is above $\mathcal{T}_f$ in D, then the probability of a relative error of more than ε is at most $\mathcal{T}_E$.

[1] There may be scenarios where our assumption does not hold. That is, even if something is true for most tuples in D', the information is still sensitive. An example would be a dataset containing only information about patients with a certain cancer disease. In such a case knowing that a person is in the dataset is already considered sensitive, and all attributes will be sensitive. Hence, our proposed model becomes irrelevant and does not apply.

Please cite this article in press as: Fu A.W.-C et al. Small sum privacy and large sum utility in data publishing. J Biomed Inform (2014), http://dx.doi.org/10.1016/j.jbi.2014.04.002

id	Age	Zip-Code	Disease
1	45	[12k-17k]	Hiv
2	33	[20k-33k]	Flu
3	24	[12k-17k]	Flu
4	76	[20k-33k]	Fever
5	61	[34k-35k]	Hiv
6	32	[12k-17k]	Flu
7	55	[34k-35k]	Fever
8	30	[20k-33k]	Fever
9	22	[12k-17k]	H5N1

(a) Given dataset D

Age	Zip-Code	Disease
32	[12k-17k]	Hiv
22	[12k-17k]	H5N1
55	[34k-35k]	Fever
30	[20k-33k]	Flu
61	[34k-35k]	Flu
33	[20k-33k]	Flu
24	[12k-17k]	Fever
76	[20k-33k]	Hiv
45	[12k-17k]	Flu

(b) Published dataset

Fig. 2. An example.

Definition 2 (*small sum privacy*). We say that $\mathcal{A}$ satisfies SSP-privacy requirement w.r.t. $(\varepsilon,\ \mathcal{T}_P,\ \alpha)$ if

$$Pr[|f'_c - f_c| \geqslant \varepsilon f_c] \geqslant \mathcal{T}_P \text{ for } f_c \leqslant \alpha \quad (2)$$

Definition 3 (*Problem Definition*). Our problem is to design a database sanitization mechanism $\mathcal{A}$ which conforms to SPLU by supporting large sum utility as well as small sum privacy.

The threshold for small sum privacy should be set by the application based on the level of security that is desired. Similarly for the threshold for large sum utility. Note that the two thresholds may be different, so that there can be some frequencies that we guarantee neither privacy nor utility. These thresholds are specifications to be given by the users. Also note that a user needs to first decide if the attributes with large sums are sensitive. As we have discussed in Footnote 1 in Section 1, there are scenarios where they can be sensitive. If large sum data are sensitive, our model will be irrelevant.

We shall also make use of the following definition for privacy guarantee.

Definition 4 (*privacy guarantee*). Concept c has a $(\varepsilon, \mathcal{T}_P)$ privacy guarantee if $Pr[|f'_c - f_c| \geqslant \varepsilon f_c] \geqslant \mathcal{T}_P$.

The above definition of small sum privacy resembles the definition of differential privacy in [12] in that probabilistic bounds are adopted. However, the exact formulations are quite different. A randomized algorithm $\mathcal{A}$ is said to give *ϵ-differential privacy* if for all datasets D and D' differing on at most one row, for all $S \in Range(\mathcal{A})$, $Pr[\mathcal{A}(D) \in S] \leqslant e^{\epsilon} \times Pr[\mathcal{A}(D') \in S]$, where the probability space is over the coin flips of $\mathcal{A}$, and $Range(\mathcal{A})$ denotes the output range of $\mathcal{A}$.

The goal of small sum privacy is to ensure that information that applies to a small number of individuals would not be released without introducing a significant amount of error probabilistically. SPLU is defined based on the estimated frequency. It will be up to the anonymization process to ensure that the estimated frequency is not accurate. We shall propose a mechanism *SPLU*-Gen that has a good guarantee in Section 3.

We do not exclude the possibility that existing methods may also be shown to satisfy SPLU. In our empirical studies we show that Anatomy may achieve high enough relative errors for small sums and good utility for large sums. Also we show that ϵ-differential privacy may achieve high relative error for small sums with a proper choice of ϵ. However, these are empirical results and not theoretical guarantees. There may be future works to show that both techniques conform to SPLU under certain settings.

3. A mechanism for SPLU

In this section, we make use of a randomization technique to guarantee a tapering accuracy for the estimated values from large counts to small counts. In [4], a retention replacement perturbation (RRP) scheme for categorical sensitive values is proposed. This scheme keeps the original sensitive value in a tuple with a probability of p and randomly picks any other value to substitute for the true value with a probability of $(1-p)$. Not all randomization techniques are equally effective. For example, if we simply generate sensitive values in the published records based on the original distribution of *SA* (let us refer to this distribution based randomization technique as *DBR*), the correlation with the *NSA* values will be lost since the generation of *SA* value for each tuple is based on the same p.d.f. In what follows, we propose a mechanism, called *SPLU*-Gen, which introduces uniform probability for replacement over a subset of the domain. We will show that this mechanism is a solution to our problem in Definition 3.

3.1. Randomization by SPLU-Gen

SPLU-Gen generates a dataset D' given the dataset D. We assume that there is a single sensitive attribute (*SA*) S in D. Later we shall discuss the more general case of multiple sensitive attributes. We make the same assumption as in previous works [20,30] that the dataset is eligible, so that the highest frequency of any sensitive attribute value does not exceed N/γ. Furthermore we assume that N is a multiple of γ (it is easy to ensure this by deleting no more than $\gamma - 1$ tuples from the dataset). There are four main steps for *SPLU*-Gen:

[Step 1] Include the tuple id as an attribute *id* in D. The first step of *SPLU*-Gen is an initialization step, whereby the dataset D goes through a projection operation on *id* and the *SA* attribute S. Let the resulting table be D_s. That is, $D_s = \Pi_{id,S}(D)$. Note that the non-sensitive values have no influence on the generation of D_s. Fig. 3(a) shows a given dataset D, which is the same as the dataset in Fig. 1(a). After Step 1, the projected table of D_s is shown in Fig. 3(b).

[Step 2] The set of tuples in D_s is partitioned into groups of size γ each in such a way that in each partitioned group, the sensitive value of each tuple is unique. Let there be r partitioned groups, $P_1, \ldots, P_r$; in each group P_i, there are γ tuples, and γ different sensitive values. We call each partitioned group a *decoy group*. If tuple t is in P_j, we say that the elements in P_j are the decoys for t. We also refer to P_j as $P(t)$. With a little abuse of terminology, we also refer to the set of records in D with the same *id*'s as the tuples in this decoy group as $P(t)$. One can adopt some existing partitioning method in the literature of ℓ-diversity (e.g. [30]). We require that the method be deterministic. That is, given a D_s (*which involves only id and S*), there is a unique partitioning from this step.

Consider dataset D in Fig. 3(a). Let $\gamma = 3$. Fig. 3(c) shows a possible partitioning of the tuples in D into 3 groups. In each group, there are 3 tuples with distinct Disease values.

[Step 3] For each given tuple t in D_s, we determine the partition $P(t)$. Let the sensitive values in $P(t)$ be $\{s'_1, \ldots, s'_\gamma\}$. For each of these decoy values, there is a certain probability that the value is selected for publication as the sensitive value for t. For a value not in $\{s'_1, \ldots, s'_\gamma\}$, the probability of being published as the value for t is zero. In the following we shall also refer to the set $\{s'_1, \ldots, s'_\gamma\}$ as $decoys(t)$. Suppose that a tuple t has sensitive value $t.s$ in D. We create the tuple t' and initialize it to t. Next we generate a value to replace the S value in t' by selecting s_i with probability p_i, so that

$$p_i = p \quad \text{for } s_i = t.s$$
$$p_i = q = (1-p)\frac{1}{\gamma - 1} \quad \text{for } s_i \neq t.s,\ s_i \in decoys(t)$$
$$p_i = 0 \quad \text{for } s_i \notin decoys(t)$$

id	Age	Zip-Code	Disease
1	45	[12k-17k]	Hiv
2	33	[20k-33k]	Flu
3	24	[12k-17k]	Flu
4	76	[20k-33k]	Fever
5	61	[34k-35k]	Hiv
6	32	[12k-17k]	Flu
7	55	[34k-35k]	Fever
8	30	[20k-33k]	Fever
9	22	[12k-17k]	H5N1

(a) Given dataset D

id	Disease
1	Hiv
2	Flu
3	Flu
4	Fever
5	Hiv
6	Flu
7	Fever
8	Fever
9	H5N1

(b) Step 1: D_s

id	Disease
1	Hiv
2	Flu
4	Fever
5	Hiv
6	Flu
7	Fever
3	Flu
8	Fever
9	H5N1

(c) Step 2

id	Disease
1	Flu
2	Flu
4	Hiv
5	Flu
6	Hiv
7	Fever
3	Fever
8	Flu
9	H5N1

(d) Step 3 : Ds'

id	Age	Zip-Code	Disease
1	45	[12k-17k]	Flu
2	33	[20k-33k]	Flu
3	24	[12k-17k]	Fever
4	76	[20k-33k]	Hiv
5	61	[34k-35k]	Flu
6	32	[12k-17k]	Hiv
7	55	[34k-35k]	Fever
8	30	[20k-33k]	Flu
9	22	[12k-17k]	H5N1

(e) Step 4 : after joining

Age	Zip-Code	Disease
32	[12k-17k]	Hiv
22	[12k-17k]	H5N1
55	[34k-35k]	Fever
30	[20k-33k]	Flu
61	[34k-35k]	Flu
33	[20k-33k]	Flu
24	[12k-17k]	Fever
76	[20k-33k]	Hiv
45	[12k-17k]	Flu

(f) Final published data

Fig. 3. An example showing the steps of *SPLU*-Gen.

We set $p = q = 1/\gamma$ in our mechanism, which has the nice property that the cases for $s_i = t.s$ and $s_i \neq t.s$ are identical.

For table D_s in Fig. 3(b), with the partitioned groups in Fig. 3(c) and (d) shows a possible resulting table Ds' after the randomization process in Step 3. Since $\gamma = 3$, $p = q = 1/3$. For example, in the first group of tuples 1, 2 and 4, each of the diseases of Hiv, Flu and Fever has a probability of 1/3 to be assigned as the sensitive value for each tuple in this group.

[Step 4] The set of tuples t' created in the previous step forms a table Ds'. Remove the s column from D, resulting in D_N. Form a new table D' by joining Ds' and D_N and retaining only *NSA* and S in the join result. The tuples in D' are shuffled randomly. Finally D' and γ are published.

Given D in Fig. 3(a). In Fig. 3(e), we show the table after joining Ds' and D_N. Removing the *id* attribute from this table gives us D'. Next, the tuples in D' are randomly shuffled and Fig. 3(f) shows the published dataset. □

Our method can be seen as a variation of the Retention Replacement Perturbation (RRP) scheme which is known to preserve good utility [4]. In particular the correlation of *SA* with *NSA* is maintained by the retention of the *SA* value for each tuple with probability p.[2]

In comparison with partitioning based methods for ℓ-diversity [20] such as Anatomy [30], there are some major observations. Firstly, Anatomy can be seen as introducing a random permutation in each group, whereas our method draws values in each group with replacement rather than without replacement. Hence, both methods can preserve the distribution of *SA* well. However, random permutation has a known problem of privacy leak for infrequent values, which need suppression for privacy reason [17]. The honest recording of *SA* values resulting from the permutations in Anatomy preserves the exact counts of the sensitive values for all frequencies, and there is no protection for the small counts. *SPLU*-Gen is based on a probabilistic assignment of values to sensitive attributes in the tuples. Due to the randomization, as we shall see in later analysis, the small counts will be protected by large expected error in the results of queries for such counts.

Secondly, the partitioning information is not released by *SPLU*-Gen, in contrast to Anatomy and related approaches, in which the anonymized groups or buckets are made known in the data publishing. When the partitioning is known, each tuple has a limited set of ℓ possible values, hence Anatomy is known to suffer from background knowledge attack where the adversary has $\ell - 1$ pieces of information for eliminating the possibilities. For *SPLU*-Gen, by withholding the partitioning information, and with the possibility that a value existing in D may not exist in D', the possible values for the sensitive attribute is the entire domain. Another advantage of not releasing the partitioning information is that it makes the de Finetti attack [18] and the foreground attack in [28] mechanisms inapplicable, since both attacks are based on the knowledge of the partitioned groups. Both attacks are possible for Anatomy.

4. Aggregate estimation

In this section we examine how to answer count queries for the sensitive attribute based on the published dataset D'. Let $|D| = N$, so that there are N tuples in D. Consider a sensitive value s. Let the true frequency of s in D be f_s. By Algorithm *SPLU*-Gen, there will be f_s decoy groups which contain s in the decoy value sets. Each tuple in these groups has a probability of $p = \frac{1}{\gamma}$ to be assigned s in D'. The probability that it is assigned other values $\bar{s}$ is $1 - p$. There are $f_s\gamma$ such tuples.

Let N'_s denote the number of times that s is published in D'. The random variable N'_s follows the binomial distribution with parameters $f_s\gamma$ and p.

$$P[N'_s = x] = \binom{f_s\gamma}{x} p^x (1-p)^{f_s\gamma - x}$$

The expected value is $f_s\gamma p$. Since we set $p = q = 1/\gamma$, the expected count of s in D' is given by $e_s = p\ \gamma f_s = f_s$. That is, to estimate the true count of an *SA* value s, we simply take the count of s in D', f'_s.

Theorem 2. *The estimation of f_s by f'_s is a maximum likelihood estimation (MLE).*

Proof. Let $L(D)$ be the likelihood of the observation f'_s in D', given the original dataset D. $L(D) = Pr(f'_s|D)$

[2] Since the sensitive value has some stickiness with the original tuple with a probability of p, an adversary with background knowledge that two individuals have the same *SA* value can launch an attack if both individuals happen to retain the *SA* value and the value is extremely rare. To handle such attacks we may adopt suppression of rare values. This, however, is beyond the scope of this work.

From Mechanism A', given f_s occurrences of s in D, there will be exactly γf_s tuples that generate s in D' with a probability of p. The remaining tuples have zero probability of generating a s value. The probability that f'_s occurrences of s is generated in D' is given by

$$L(D) = Pr(f'_s|D) = \binom{\gamma f_s}{f'_s} p^{f'_s}(1-p)^{\gamma f_s - f'_s}$$

where $p = 1/\gamma$. This is a binomial distribution function which is maximized when f'_s is at the mean value of $\gamma f_s p = f_s$. In particular, $lnL(D) = c + f'_s lnp + (\gamma f_s - f'_s) ln(1-p)$ for a constant c. Setting $\frac{d}{dp}(lnL(D)) = \frac{f'_s}{p} - \frac{\gamma f_s - f'_s}{1-p} = 0$ gives $f_s = f'_s$. □

To examine the utility of the dataset D', we ask how likely it is for f'_s to be close to f_s. We also need to provide protection for small counts. Next, we show that our method simultaneously discloses useful information where the sum is large and hence safe, and withholds accurate information when the sum is small.

4.1. Large sum utility

To answer the question about the utility for large sums, we derive a bound for the relative error. If there are f_s tuples with s value, then $n = \gamma f_s$ tuples in D will have a probability of p to be assigned s in D'. The setting of value s to the tuples in D' corresponds to a sequence of γf_s independent Bernoulli random variables, $X_1, \ldots, X_{\gamma f_s}$, each with parameter p. Here $X_i = 1$ corresponds to the event that s is chosen for the i-th tuple, while $X_i = 0$ corresponds to the case where s is not chosen. Since $p = 1/\gamma$ and $n = \gamma f_s$, $f_s = np$.

Theorem 3. *For $\varepsilon > 0$,*

$$Pr[|f'_s - f_s| \geqslant \varepsilon f_s] \leqslant \frac{1}{\gamma \varepsilon^2 f_s^2} \tag{3}$$

Proof. The proof of the utility of the published data for large sums is based on Chebychev's Theorem: If X is a random variable with mean μ and standard deviation σ, then for any positive k, $Pr(|X-\mu| < k\sigma) \geqslant 1 - \frac{1}{k^2}$ and $Pr(|X-\mu| \geqslant k\sigma) \leqslant \frac{1}{k^2}$.

Let $X_1, X_2, \ldots, X_n, \ldots$ be a sequence of independent, identically distributed random variables, each with mean μ and variance σ^2. Define the new sequence of $\overline{X}_i$ values by $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, $n = 1, 2, 3, \ldots$.

From Chebychev's inequality, $P\left[|\overline{X}_n - \mu_{\overline{X}_n}| \geqslant k\sigma_{\overline{X}_n}\right] \leqslant \frac{1}{k^2}$ where $\mu_{\overline{X}_n} = E[\overline{X}_n] = \mu$, $\sigma_{\overline{X}_n} = E[(\overline{X}_n - \mu)^2] = \frac{\sigma^2}{n}$ and k is any positive real number. Choose $k = \frac{\epsilon\sqrt{n}}{\sigma}$ for some $\epsilon > 0$, we get

$$Pr[|\overline{X}_n - \mu| \geqslant \epsilon] \leqslant \frac{\sigma^2}{\epsilon^2 n}. \tag{4}$$

We use the above reasoning to derive the utility of our published data for large sums. If there are f_s tuples with s value, then $n = \gamma f_s$ tuples in D will have a probability of p to be assigned s in D'. The setting of value s to the tuples in D' corresponds to a sequence of γf_s independent Bernoulli random variables, $X_1, \ldots, X_{\gamma f_s}$, each with parameter p. Here $X_i = 1$ corresponds to the event that s is chosen for the i-th tuple, while $X_i = 0$ corresponds to the case where s is not chosen.

The mean value $\mu_{\overline{X}_n} = p$. Also, $\sigma^2_{\overline{X}_n} = p(1-p)/n$. From Inequality (4), $Pr[|\overline{X}_n - \mu| \geqslant \epsilon] \leqslant \frac{p(1-p)}{\epsilon^2 n^2} = \frac{p(1-p)}{\epsilon^2 \gamma^2 f_s^2}$. We set $p = \frac{1}{\gamma}$, hence

$$Pr[|\overline{X}_n - \mu| \geqslant \epsilon] \leqslant \frac{1}{\gamma^3 \epsilon^2 f_s^2} \tag{5}$$

Note that $\overline{X}_n$ is the count of s in D' divided by n, and $n = \gamma f_s$. Hence, the occurrence of s in D' is $f'_s = \gamma f_s \overline{X}_n$.

Rewriting Inequality (5), we get $Pr[|\gamma f_s \overline{X}_n - \gamma f_s \mu| \geqslant \gamma f_s \epsilon] \leqslant \frac{1}{\gamma^3 \epsilon^2 f_s^2}$ Since $\mu = p = 1/\gamma$, $Pr[|f'_s - f_s| \geqslant \gamma\epsilon f_s] \leqslant \frac{1}{\gamma^3 \epsilon^2 f_s^2}$.

With the above inequality, we are interested in how different f'_s is from f_s. Since the deviation is bounded by $\gamma\epsilon f_s$, it is better to use another variable $\varepsilon = \gamma\epsilon$ to quantify the difference.

$$Pr[|f'_s - f_s| \geqslant \varepsilon f_s] \leqslant \frac{1}{\gamma \varepsilon^2 f_s^2} \quad \square$$

Our estimation is $e_s = f'_s$, hence the above gives a bound on the probability of error in our estimation. If f_s is small, then the bound is large. In other words the utility is not guaranteed, which means better privacy protection.

Given a desired ε and a desired γ, we may find a frequency threshold $\mathcal{T}_f$ so that for f_s above this threshold, the probability of error in Inequality (3) is below another threshold $\mathcal{T}_E$ for utility. We can set the RHS in the above inequality to be this threshold.

Lemma 1. *SPLU-Gen provides a $(\epsilon, \mathcal{T}_E, \mathcal{T}_f)$ utility guarantee for each sensitive value, where $\mathcal{T}_f = \left(\frac{1}{\gamma\varepsilon^2\mathcal{T}_E}\right)^{\frac{1}{2}}$*

Hence, given ε and $\mathcal{T}_E$, we can determine the smallest count which can provide the utility guarantee.

Fig. 4 shows the relationship between the possible values of $\mathcal{T}_f$ and $\mathcal{T}_E$. The utility is better for small $\mathcal{T}_E$, and the value of $\mathcal{T}_E$ becomes very small when the count is increasing towards 900. Note that it also means that for concepts with large counts, privacy protection is not guaranteed, since the accuracy in the count will be high.

4.2. Small sum privacy

Next, we show how our mechanism can inherently provide protection for small counts. From Inequality (3), small values of f_s will weaken the guarantee of utility. We can in fact give a probability for relative errors based on the following analysis.

The number of s in D' is the total number of successes in $f_s\gamma$ repeated independent Bernoulli trials with probability $\frac{1}{\gamma}$ of success on a given trial. It is the binomial random variable with parameters $n = \gamma f_s$ $p = \frac{1}{\gamma}$, and $q = 1 - p$. The probability that this number is x is given by $\binom{n}{x} p^x q^{n-x} = \binom{\gamma f_s}{x} \left(\frac{1}{\gamma}\right)^x \left(1 - \frac{1}{\gamma}\right)^{\gamma f_s - x}$.

Example 1. If $f_s = 5$, $\gamma = 10$, for an $\varepsilon = 0.3$ bound on the relative error, we are interested to know how likely it is for f'_s to be close to 5 within a deviation of 1. The probability that f'_s is between 4 and 6 is given by $\sum_{x=4}^{6} \binom{50}{x} 0.1^x 0.9^{50-x} = 0.52$. Hence, the probability that f'_s deviates from f_s by more than $0.3 f_s$ is 0.48.

From Definition 4, a sensitive value s has a $(\varepsilon, \mathcal{T}_P)$ privacy guarantee if the probability that the estimated count of s, f'_s, has

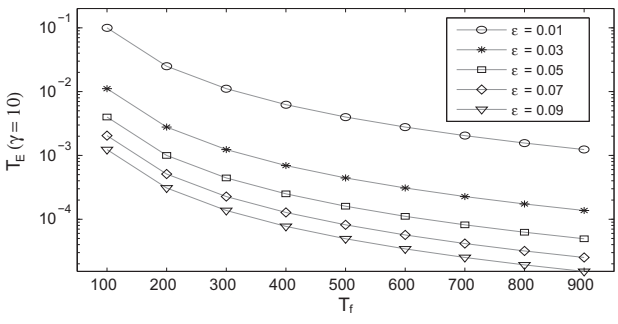


Fig. 4. Relationship between $\mathcal{T}_E$ and $\mathcal{T}_f$.

a relative error of more than ε is at least $\mathcal{T}_P$. In Example 1, the value s has a $(0.3, 0.48)$ privacy guarantee. We can derive the following.

Lemma 2. *Considering only single SA value concepts, SPLU-Gen with parameter γ satisfies SSP-privacy with respect to $(\varepsilon, \mathcal{T}_P, \alpha)$, where*

$$\mathcal{T}_P = \min_{f_s \in [1..\alpha]} \left(1 - \sum_{x=\lceil (1-\varepsilon) f_s \rceil}^{\lfloor (1+\varepsilon) f_s \rfloor} \binom{\gamma f_s}{x} \frac{1}{\gamma^x} \left(1 - \frac{1}{\gamma}\right)^{\gamma f_s - x}\right)$$

Note that this guarantee is independent of the dataset size and independent of the data distribution. Also note that a closed form approximation by replacing the binomial with a normal distribution is not applicable here since the value of f_s of interest is very small.

Example 2. A graph is plotted in Fig. 5 for the expected error for small values of f_s. Here the summation in the above probability is taken from $f_s' = \lceil 0.7 f_s \rceil$ to $f_s' = \lfloor 1.3 f_s \rfloor$. We have plotted for different f_s values the probability given by $1 - \sum_{x=\lceil 0.7 f_s \rceil}^{\lfloor 1.3 f_s \rfloor} \binom{\gamma f_s}{x} \frac{1}{\gamma^x} \left(1 - \frac{1}{\gamma}\right)^{\gamma f_s - x}$. Due to the rounding effects of the summation over integer values of x, the graph has a sawtooth shape. This graph shows that the relative error in the count estimation is expected to be large for sensitive values with small counts. From this graph, we can derive that for single *SA* value concepts, given $\alpha = 3$, $\varepsilon = 0.3$, $\mathcal{T}_P = 0.6$, *SPLU*-Gen with $\gamma = 10$ satisfies *SPLU*-privacy with respect to $(\varepsilon, \mathcal{T}_P, \alpha)$. The graph also shows that the choice of γ has little impact on the guarantee.

5. Multiple attribute predicates

In this section we consider the reconstruction of counts for sets of values. For example, we may want to estimate the count of tuples with both lung cancer and smoking, or the count of tuples with *gender = female, Age = 60* and *disease = allergy*. First we shall consider the estimation of counts for predicates involving a single sensitive attribute, then we extend our discussion to predicates involving multiple sensitive attributes.

5.1. Predicates involving a single SA

Assume that we have a set of non-sensitive attributes *NSA* and a single sensitive attribute *SA*. Let us consider queries involving both *NSA* and *SA*. We may divide such a query into two components: P and s, where $P \in domain(NA)$ ($NA \subseteq NSA$), and $s \in domain(SA)$. For example $P = (female, 60)$ and $s = (allergy)$. Note that the non-sensitive attributes are not distorted in the published dataset. This can be seen as a special case of generating a non-sensitive value for the individual t by selecting s_i with probability p_i, so that $p_i = 1$ for $s_i = t.s$; and $p_i = 0$ for $s_i \neq t.s$.

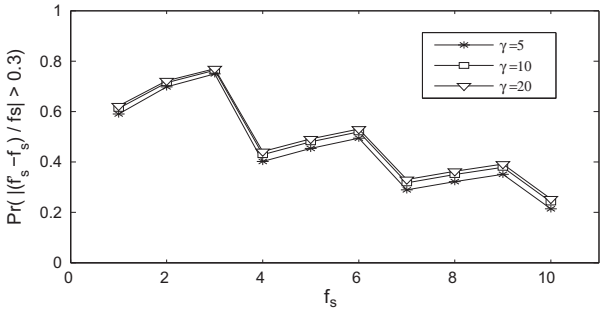


Fig. 5. Expected error for small sums.

Suppose we are interested in the count of the co-occurrences of non-sensitive values P and sensitive value s.

Definition 5 (*state i*). There are 4 conjunctive predicates concerning P and s, namely, $\phi_0 = \overline{P} \wedge \overline{s}$, $\phi_1 = \overline{P} \wedge s$, $\phi_2 = P \wedge \overline{s}$, and $\phi_3 = P \wedge s$. If a tuple satisfies ϕ_i, we say that it is at state *i*.

The distributions of the predicates in D and D' are given by $cnt(\phi_i)$ and $cnt'(\phi_i)$, respectively. Here $cnt(\phi_i)(cnt'(\phi_i))$ is the number of tuples satisfying ϕ_i in D (D').

For simplicity we let $x_i = cnt(\phi_i)$ and $y_i = cnt'(\phi_i)$, hence the a priori distribution concerning the states in D is given by $x = \{x_0, x_1, x_2, x_3\}$, and the distribution in D' is given by $y = \{y_0, y_1, y_2, y_3\}$. Hence, y contains the observed frequencies.

Definition 6 (*Transition matrix M*). The probability of transition for a tuple from an initial state i in D to a state j in D' is given by a_{ij}. Values a_{ij} form a transition matrix M.

Let $Pr(r_i|x)$ be the probability that a tuple is at state i in D' given vector x for the initial state distribution. The following can be derived.

$$Pr(r_0|x) = \frac{1}{N}\left(\left(1 - \frac{x_1 + x_3}{N}\right)x_0 + \frac{\gamma - 1}{\gamma}x_1\right) \quad (6)$$

$$Pr(r_1|x) = \frac{1}{N}\left(\left(\frac{x_1 + x_3}{N}\right)x_0 + \frac{1}{\gamma}x_1\right) \quad (7)$$

$$Pr(r_2|x) = \frac{1}{N}\left(\left(1 - \frac{x_1 + x_3}{N}\right)x_2 + \frac{\gamma - 1}{\gamma}x_3\right) \quad (8)$$

$$Pr(r_3|x) = \frac{1}{N}\left(\left(\frac{x_1 + x_3}{N}\right)x_2 + \frac{1}{\gamma}x_3\right) \quad (9)$$

The above equations are based on the mechanism generating D' from D. Let us consider the last equation, the other equations are derived in a similar manner. For each true occurrence of (P, s), there is a $\frac{1}{\gamma}$ probability that it will generate such an occurrence in D'. If there are x_3 such tuples, then the expected number of generated instances will be x_3/γ.

Other occurrences of (P, s) in D' may be generated by the x_2 tuples satisfying P but with $t.s \neq s$ $(P, \bar{s})$. Each such tuple t satisfies P for the non-sensitive values and it is possible that $s \in decoys(t)$. We are interested to know how likely it is that $s \in decoys(t)$.

There are in total $\frac{N}{\gamma}$ partitions. There can be at most one s tuple in each partition. Hence, f_s of the partitions contain s in the decoy set, and if a tuple t is in such a partition, then $s \in decoys(t)$. The probability of having s in $decoys(t)$ for a tuple t with $t.s \neq s$ is the probability that t is in one of the f_s partitions above given that $t.s \neq s$. Note that the condition in this probability is $t.s \neq s$ and not on more detailed information about t. This probability is given by $f_s/\frac{N}{\gamma} = f_s\frac{\gamma}{N}$. Since $f_s = x_1 + x_3$, this probability is $\frac{x_1+x_3}{N}\gamma$. The total

	$y_0(\bar{P}\bar{s})$	$y_1(\bar{P}s)$	$y_2(P\bar{s})$	$y_3(Ps)$
$x_0(\overline{P}\overline{s})$	$a_{00} = 1 - a_{01}$	$a_{01} = \frac{x_1+x_3}{N}$	$a_{02} = 0$	$a_{03} = 0$
$x_1(\overline{P}s)$	$a_{10} = \frac{\gamma-1}{\gamma}$	$a_{11} = \frac{1}{\gamma}$	$a_{12} = 0$	$a_{13} = 0$
$x_2(P\overline{s})$	$a_{20} = 0$	$a_{21} = 0$	$a_{22} = 1 - a_{23}$	$a_{23} = \frac{x_1+x_3}{N}$
$x_3(Ps)$	$a_{30} = 0$	$a_{31} = 0$	$a_{32} = \frac{\gamma-1}{\gamma}$	$a_{33} = \frac{1}{\gamma}$

Fig. 6. State transition probabilities.

Please cite this article in press as: Fu A.W.-C et al. Small sum privacy and large sum utility in data publishing. J Biomed Inform (2014), http://dx.doi.org/10.1016/j.jbi.2014.04.002

expected occurrence of (P,s) is given by $\frac{x_3}{\gamma}+(\frac{x_1+x_3}{N}\gamma)\frac{x_2}{\gamma}$. We can convert this into a conditional probability that a tuple in D' satisfies (P,s) given x, denoted by $Pr(r_3|x)$. This gives Eq. (9).

Rewriting Eqs. (6)–(9) with the transition probabilities in Fig. 6 gives the following:

$$Pr(r_i|x)=\sum_{j=0}^{3}a_{ji}\frac{x_j}{N} \tag{10}$$

Eq. (10) shows that a_{ji} is the probability of transition for a tuple from an initial state j in D to a state i in D'.

We adopt the iterative Bayesian technique for the estimation of the counts of $x_0,\ldots,x_3$. This method is similar to the technique in [4] for reconstructing multiple column aggregates.

Let the original states of tuples $t_1,\ldots,t_N$ in D be $U_1,\ldots,U_N$, respectively. Let the states of the corresponding tuples in D' be $V_1,\ldots,V_N$. From Bayes rule, we have

$$Pr(U_k=i|V_k=j)=\frac{P(V_k=j|U_k=i)P(U_k=i)}{P(V_k=j)}$$

Since $Pr(U_k=i)=x_i/N$, and $Pr(V_k=j|U_k=i)=a_{ij}$,

$$Pr(U_k=i|V_k=j)=\frac{a_{ij}\frac{x_i}{N}}{\sum_{r=0}^{3}a_{rj}\frac{x_r}{N}} \tag{11}$$

$$Pr(U_k=i)=\sum_{j=0}^{3}Pr(V_k=j)Pr(U_k=i|V_k=j)$$

Hence, since $Pr(V_k=j)=y_j/N, Pr(U_k=j)=x_j/N$ and from Eq. (11), we have

$$\frac{x_i}{N}=\sum_{j=0}^{3}\frac{y_j}{N}\frac{a_{ij}\frac{x_i}{N}}{\sum_{r=0}^{3}a_{rj}\frac{x_r}{N}} \tag{12}$$

We iteratively update x using the following equation

$$x_i^{t+1}=\sum_{j=0}^{3}y_j\frac{a_{ij}^t x_i^t}{\sum_{r=0}^{3}a_{rj}^t x_r^t} \tag{13}$$

We initialize $x^0=y$, and x^t is the value of x at iteration t. In Eq. (13), a_{ij}^t refers to the value of a_{ij} at iteration t, meaning that the value of a_{ij}^t depends on setting the values of $x=x^t$. We iterate until x^{t+1} does not differ much from x^t. In our experiments, the stopping criterion is that for all $0\leqslant i\leqslant 3, |x_i^{t+1}-x_i^t|/x_i^t\leqslant 0.01$. The value of x at this fixed point is taken as the estimated x value. In particular, x_3 is the estimated count of (P,s).

5.2. Multiple sensitive attributes

So far we have considered that there is a single sensitive attribute in the given dataset. Suppose instead of a single sensitive attribute (SA), there are multiple SAs. Let the sensitive attributes be $S_1,S_2,\ldots S_w$. We can generalize the randomization process by treating each SA independently, building decoy sets for each S_i.

For predicates involving $\{P,s_1,s_2,\ldots,s_w\}$, where P is a set of values for a set of non-sensitive attributes, $s_i\in domain(S_i)$, there will be $K=2^{w+1}$ different possible states for each tuple. We let $(P,s_1,s_2,\ldots,s_w)$ stand for $(P\wedge s_1\wedge s_2\ldots\wedge s_w)$. For reconstruction of the count for $(P,s_1,s_2,\ldots,s_w)$, we form a transition matrix for all the $K=2^{w+1}$ possible states. It is easy to see that the case of a single SA in Section 5.1 is a special case where the transition matrix M is the tensor product of two matrices M_0 and $M_1, A=M_0\otimes M_1$, where M_0 is for the set of non-sensitive values and M_1 is for s_1, and they are defined as follows:

$$M_0=\begin{pmatrix}1&0\\0&1\end{pmatrix}M_i=\begin{pmatrix}1-f_{s_i}/N & f_{s_i}/N\\(\gamma-1)/\gamma & 1/\gamma\end{pmatrix}$$

In general, with sensitive attributes $S_1,\ldots,S_w$, the transition matrix is given by $M=M_0\otimes M_1\ldots\otimes M_w$.

Let the entries in matrix M be given by m_{ij}. We initialize $x^0=y$ and iteratively update x using the following equation

$$x_i^{t+1}=\sum_{j=0}^{K-1}y_j\frac{m_{ij}^t x_i^t}{\sum_{r=0}^{K-1}m_{rj}^t x_r^t} \tag{14}$$

In Eq. (14), x^t is the value of x at iteration t. a_{ij}^t refers to the value of m_{ij} at iteration t, meaning that the value of m_{ij}^t depends on setting the values of $x=x^t$. We iterate until x^{t+1} does not differ much from x^t. In our experiments the stopping criterion is that for all $0\leqslant i\leqslant 3, |x_i^{t+1}-x_i^t|/x_i^t\leqslant 0.01$. The value of x at this fixed point is taken as the estimated x value. In particular x_{K-1} is the estimated count of $(P,s_1,\ldots,s_w)$.

5.3. Small sum privacy

For the multiple attribute predicate counts, we also guarantee that privacy for small sums will not be jeopardized.

Lemma 3. *Let s be a sensitive value with a $(\varepsilon,\mathcal{T}_p)$ privacy guarantee, then the count for a multiple column aggregate involving s also has the same privacy guarantee.*

Proof. Without loss of generality, consider a multiple attribute aggregate of (P,s), where $P\in domain(NSA)$. Since the randomization of s is independent of the NSA attributes, the expected relative error introduced for $(\overline{P},s)$ is the same as that for (P,s). The total expected error for $(\overline{P},s)$ and (P,s) must not be less than that dictated by the $(\varepsilon,\mathcal{T}_P)$ guarantee since otherwise the sum of the two counts will generate a better estimate for the count of s, violating the $(\varepsilon,\mathcal{T}_P)$ privacy for s. Hence for (P,s) the privacy guarantee is at least $(\varepsilon,\mathcal{T}_P)$. □

Let $\phi=(P,s)$ be a predicate with sensitive value s. and $P\in domain(NA)$ $(NA\subseteq NSA)$. There are two possible cases as follows. CASE 1: There exists f_ϕ occurrences of ϕ in D and there is no other occurrence of s in D. CASE 2: There exists f_ϕ occurrences of ϕ in D and there exists other occurrence(s) of s in D.

Lemma 4. *In CASE 1, the estimation of f_ϕ by f_s' is a maximum likelihood estimation (MLE).*

Proof. There are in total f_ϕ occurrences of s in D. Hence estimating the number of s in D also estimates the number of ϕ in D. With $SPLU$-Gen, given f_ϕ occurrences of s in D, there will be exactly γf_s tuples that generates s in D' with a probability of p, where $p=1/\gamma$. The remaining tuples have zero probability of generating a s value. The maximum likelihood of the estimation by f_s' follows the same argument as that in the proof of Theorem 2. □

For CASE 2, suppose there are f_ϕ occurrences of ϕ in D and $f_{\overline{P}s}$ other occurrences of s. Let $\phi_1=(\overline{P}\wedge s)$. Estimation of f_ϕ by f_s' will be affected by other occurrences of s in D' that may be generated from groups containing records satisfying ϕ_1. Suppose f_s' is taken to be the estimation of f_ϕ, then we can show that the probability of the estimation error exceeding a given relative error threshold is always at least the same as that in CASE 1: if there is no occurrence of ϕ_1, the probability that f_s' is within ε of the true f_ϕ value is

given by $Prob_A = \sum_{x=\lceil(1-\varepsilon)f_\phi\rceil}^{\lfloor(1+\varepsilon)f_\phi\rfloor} \binom{\gamma f_\phi}{x} \left(\frac{1}{\gamma}\right)^x (1-\frac{1}{\gamma})^{\gamma f_\phi - x}$. If there are x' instances of s generated by instances of ϕ_1, the probability will become $Prob_B = \sum_{x=\lceil(1-\varepsilon)f_\phi\rceil - x'}^{\lfloor(1+\varepsilon)f_\phi\rfloor - x'} \binom{\gamma f_\phi}{x} \left(\frac{1}{\gamma}\right)^x (1-\frac{1}{\gamma})^{\gamma f_\phi - x}$. It is easy to show that $Prob_B \leqslant Prob_A$.

We next consider how to derive the *SPLU*-privacy property of *SPLU*-Gen. For CASE 1, the number of ϕ in D' is the total number of successes in $f_\phi\gamma$ repeated independent Bernoulli trials, with a probability of $\frac{1}{\gamma}$ for success on a given trial. Hence, we can derive the $(\varepsilon, \mathcal{T}_P)$ privacy guarantee for ϕ as in Section 4.2.

For CASE 2, let us consider the scenario that x', the number of occurrences of s due to occurrences of ϕ_1, has also been given for the estimation of f_ϕ. In this case the estimation of f_ϕ by $f'_s - x'$ will be the maximum likelihood estimation (MLE) for f_ϕ, with the argument given for Lemma 4, and the privacy guarantee is the same as CASE 1. If the value of x' is not given, the estimation of f_ϕ will no longer have the MLE guarantee, and therefore we use the CASE 1 guarantee as a bound for the overall guarantee. Hence, we have the following result.

Theorem 4. *SPLU-Gen with parameter γ satisfies SSP-privacy with respect to $(\varepsilon, \mathcal{T}_P, \alpha)$, where*

$$\mathcal{T}_P = \min_{f_\phi \in [1..\alpha]} \left(1 - \sum_{x=\lceil(1-\varepsilon)f_\phi\rceil}^{\lfloor(1+\varepsilon)f_\phi\rfloor} \binom{\gamma f_\phi}{x} \left(\frac{1}{\gamma}\right)^x \left(1-\frac{1}{\gamma}\right)^{\gamma f_\phi - x}\right) \quad \square$$

6. Empirical study

We have implemented our mechanism *SPLU*-Gen and compared it with some existing techniques that are related in some way to our method. The objectives of our empirical study are the following: (1) While we have shown in Section 4 that *SPLU*-Gen satisfies small sum privacy and large sum utility, we would like to see the actual results in a real dataset. Hence, in our experiments we adopt the measure of relative error to illustrate the privacy and utility levels, where a higher relative error corresponds to more privacy and less utility. (2) Demonstrate the effectiveness of our sophisticated reconstruction mechanism, which makes the querying results more accurate for large sum queries compared to previous approach. We shall compare with Anatomy. (3) Show that ϵ-differential privacy may not preserve small sum privacy with some known parameter settings, and therefore careful parameter setting is needed. (4) Show that the choice of retention based randomization is sound by comparing with a randomization algorithm (DBR) that does not have a retention probability for the original *SA* value in each tuple. (5) Show that correlation for the case of multiple *SA* attributes can be preserved. (6) Show that the computations involved are not costly.

6.1. Experimental setup

All algorithms are implemented in C++ and tested on a machine with Intel (R) Core (TM) i3 3.10 GHz CPU and 4.0 GB RAM. The program for Anatomy is provided by the first author of [30].

For step 2 of mechanism *SPLU*-Gen, we need to partition tuples in D_s into sets of size γ each and each partition contains γ different sensitive values. We have adopted the group creation step in the algorithm for Anatomy [30]. In this algorithm, all tuples of the given table are hashed into buckets by the sensitive values, so that each bucket contains tuples with the same *SA* value. The group creation step consists of multiple iterations. In each iteration a partition (group) with γ tuples is created. Each iteration has two substeps: (1) find the set L with the γ hash buckets that currently have the largest number of tuples. (2) From each bucket in L, randomly select a tuple to be included in the newly formed partition. Note that the random selection in step (2) can be made deterministic by picking the tuple with the smallest tuple id.

We use two real datasets. The first dataset is CENSUS,[3] which contains the information for American adults. The second dataset, CADRMP, is from a publicly available[4] real hospital database.

The CENSUS dataset consists of 8 categorical dimensions: Gender (cardinality 2), Education (17), Marital (6), Race (9), Work-class (10), Country (83), Age (78), and Occupation (50). From the CENSUS dataset, we form datasets with increasing cardinalities. First, we randomly sampled 500 k tuples from the real dataset, then we further randomly sampled five datasets from the 500 k tuples, with cardinalities ranging from 100 k to 500 k, the 100 k dataset is used as the default. *Occupation* is chosen as the sensitive attribute, while it is combined with *Age* for experiments with multiple sensitive values.

For the dataset CADRMP, there are 8 tables: *Reports*, *Reactions, Drugs, ReportDrug, Ingredients, Outcome*, and *Druginvolve*. *Reports* consists of patients' basic personal information. *Reactions* has a foreign key *PID* that references the attribute *ID* in *Reports* and an attribute to indicate the patient's disease. *Reports* contains records for 42,264 different individuals. A patient may have multiple diseases. We first pre-process the dataset so that we retain only the most frequent disease for each patient. Next we join the tables *Reports* and *Reactions* in order to link the patient information with the disease information. There are 42,264 tuples in the joined table with 17 attributes each. The attributes are Report-id (cardinality 42264), Report-no (6998), Gender-English (5), Age (107), Age-unit (1), Weight (316), Height (245), Manufacture-id (3241), Date-of-Last-Follow-up (1200), Serious (3), Feature-of-Report (12), Report-Type (11), Notifier-Type (12), Notifier-Location (13), ADR-id (42264) and Disease (1346). Disease will be used as the sensitive attribute.

In the experiment we consider count queries, which have been used for utility studies for partition-based methods [30] and randomization-based methods [21]. Count queries are generated according to the method described in Appendix 10.9 in [5]. Specifically, we generate random predicates with up to 3 of the non-sensitive attributes, each of which is combined with a randomly selected value in the domain of the sensitive attribute to form a query. We count the tuples satisfying such a query, which is of the form $A_1 = v_1 \wedge \ldots \wedge A_d = v_d \wedge SA = v_s$, where each A_i is a distinct non-sensitive attribute, *SA* is the sensitive attribute, and the v_i and v_s are values from the domains of A_i and *SA*, respectively. The selectivity of a query is defined as the percentage of tuples that satisfy the conditions in the query. In most of our analysis, we group queries according to their distinct selectivity ranges. We consider the queries with selectivity no more than 10 as small counts. Large count queries are those with $[0.5\%, 5\%)$ selectivity in the CENSUS dataset, and with $[1\%, 9\%)$ selectivity in the CADRMP dataset. A pool of 5000 small count queries and a pool of 5000 large count queries are generated for each dataset. When a selectivity s is considered without a range, we report on the average relative error of the estimated count for all queries that pass the selectivity threshold s. For a single query, if the correct answer is a and the returned answer is a', then the relative error is given by $|a - a'|/a$. For a set of Q queries, if the sum of relative error for all queries in the set is E, then the average relative error is E/Q.

Given queries in the pool, we calculate the average relative error between the actual count (from the original dataset) and the estimated count (from the published dataset) as the metric for utility. As discussed earlier, we differentiate between small

[3] Downloadable at http://www.ipums.org.

[4] Downloadable at http://www.cse.cuhk.edu.hk/adafu/medical-data.html, originally from http://www.hc-sc.gc.ca/dhp-mps/medeff/databasdon/structure-eng.php.

Please cite this article in press as: Fu A.W.-C et al. Small sum privacy and large sum utility in data publishing. J Biomed Inform (2014), http://dx.doi.org/10.1016/j.jbi.2014.04.002

counts and large counts. Specifically, we vary the selectivity (denoted by s, which is the ratio of the actual count to the cardinality of dataset) for large counts. For small counts, we require the actual count to be no more than 10. We evaluate the influence of various γ values, and also the cardinalities of dataset on the utility. To assess the efficiency, we record the running time of our data publishing algorithm.

6.2. Results for the CENSUS dataset

In this subsection we report on the results for the CENSUS dataset. We shall consider the utility for large sums, the privacy for small sums, and the comparison with other algorithms.

6.2.1. Utility for large sums

We first consider the utility of the anonymized data for queries with sufficient sums. The results are shown in Fig. 9(a). For selectivity (i.e., large counts) between 2% and 5%, the relative error is around 10%. The relative errors are bounded by 20% for other selectivities between 0.5% and 5%. Another observation is that the relative error is the smallest when the selectivity is the greatest at 4–5%. This shows the desired effect that the result is more accurate for large sums. In Fig. 8, we plot the relative errors for increasing selectivity from [0.5%, 1%), [1%, 2%), ..., to [4%, 5%), and we notice a drop of the relative error as selectivity reaches 5%, which is the desired behavior.

We have also run some special queries on the dataset by generalizing the domain of attribute age into intervals so that each age interval spans about 8% of the tuples, this allows us to generate queries with selectivities between 5% and 8%. With such queries, and with $\gamma = 5$, dataset of size 100 K, the average relative error for *SPLU*-Gen is 0.01, which is very small, and it shows that the utility for larger sums is very high.

6.2.2. Privacy for small sums

We plot the relative errors of queries with small counts in Fig. 7, where the counts are smaller than 10. We see that the error is sufficiently high to ensure privacy, consistent with our requirement that answer for small count should be inaccurate enough to prevent privacy leakage. The relative error also displays a positive linear correlation with γ. As γ becomes bigger, higher uncertainty is introduced and privacy for small counts is ensured at a higher level. It also indicates that a small value of γ such as 5 is sufficient for a big relative error.

6.2.3. Results from other methods

We have implemented three existing methods: ϵ-differential privacy, Anatomy, and the distribution based randomization technique (DBR). DBR is described in Section 3, it generates *SA* values based on the distributions in the original dataset.

6.2.3.1. ϵ-differential privacy. The first other method we have implemented is querying with ϵ-differential privacy. We add independently generated noise with distribution $Lap(\Delta f/\epsilon)$ to the true query result, and this ensures ϵ-differential privacy [14]. The parameter ϵ is public and as stated in [13], it may be set to values such as 0.01, 0.1, or in some cases, $\ln 2$ or $\ln 3$, and in particular, an example of $\Delta f = 1$ and $\epsilon = \ln 2$ is used for illustration. For this set of experiments, we are not comparing ϵ-differential privacy with other methods. We aim to study the effects of different values of ϵ on small sum privacy. We have plotted the result of relative error for small count queries (count between 1 and 10) in Fig. 7 for the cases of $\Delta f = 1$ and $\epsilon = \ln 3$ and also $\epsilon = \ln 2$. We can see that the relative error is quite small compared to the other methods, and can lead to privacy concerns. If ϵ is set smaller, the relative error will become bigger. This result shows that differential privacy mechanism may violate the small sum privacy requirement if the parameters are not set properly. Therefore, careful selection of parameters is needed.

6.2.3.2. Anatomy. To compare with the Anatomy method, we set both γ and ℓ in Anatomy to the same value, $\gamma = \ell = 5$. The answers for Anatomy are estimated using the method in [30]. We then choose different settings of s (selectivity) and N (sizes of dataset) to evaluate their performance. We first examine the relative error for small counts (see Fig. 7). In comparison it is noted that *SPLU*-Gen achieves a highest error rate for larger values of γ, which shows that *SPLU*-Gen is more sensitive to the group sizes. In Fig. 8, we compare the methods in terms of their achieved utility by setting $N = 300$ K and varying the selectivity. *SPLU*-Gen has a decreasing pattern for the relative error with decreasing selectivity, which is consistent with the SPLU model. Anatomy does not exhibit a similar trend. These results can be explained by the use of a randomization and reconstruction technique in *SPLU*-Gen which directly corresponds to the law of large numbers. For large count queries, the average relative errors for Anatomy and *SPLU*-Gen are shown in Fig. 9(b) and (a), respectively. The overall error of our method is smaller than that of Anatomy and also there is better correspondence of utility with selectivity. This helps to show the superiority of our reconstruction mechanism.

6.2.3.3. DBR. Next we consider the comparison with the distribution based randomization algorithm DBR. For DBR, the maximum likelihood estimations for sums of *SA* or predicates involves both *NSA* and *SA* are the corresponding counts in the published data. While DBR makes use of the original *SA* distribution in the randomization, and hence can preserve it well, it does not try to retain the original *SA* value in a given tuple. Hence, the correlation of the *NSA* and *SA* is not preserved. This affects the utility of DBR for predicates involving both *NSA* and *SA*. From Fig. 9(c), we see that DBR leads to higher relative errors for large sum queries. In Fig. 8, for DBR, we observe larger errors for smaller selectivities compared

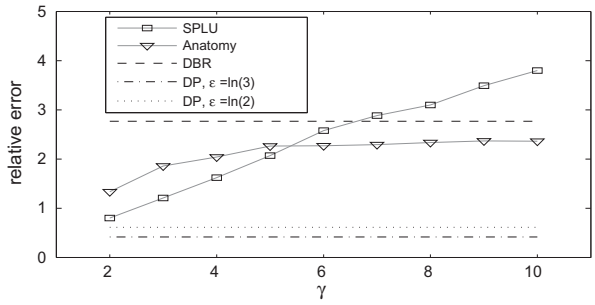


Fig. 7. Comparison of errors for *SPLU*-Gen, Anatomy, *DBR*, and differential privacy (small counts) for CENSUS.

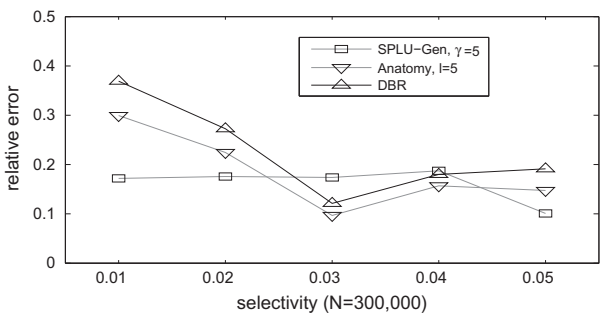
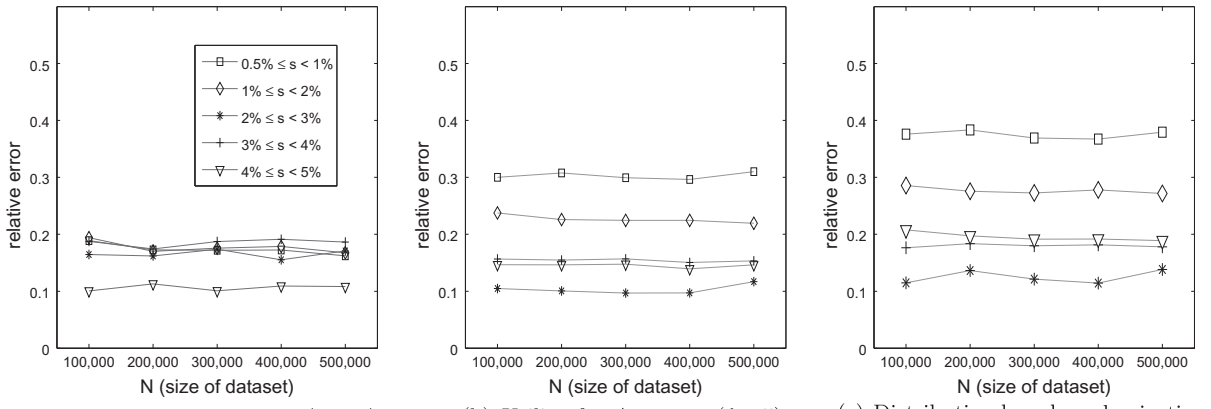


Fig. 8. Comparison of utilities for *SPLU*-Gen, Anatomy, and *DBR* for CENSUS.

Please cite this article in press as: Fu A.W.-C et al. Small sum privacy and large sum utility in data publishing. J Biomed Inform (2014), http://dx.doi.org/10.1016/j.jbi.2014.04.002



(a) Utility for $SPLU$-Gen ($\gamma = 5$) (b) Utility for Anatomy ($\ell = 5$) (c) Distribution-based randomization

Fig. 9. Comparison of utilities for CENSUS.

to *SPLU*-Gen, and an increase in error for larger selectivity. There is no obvious trend as the utility fluctuates with the correlations in the original datasets, where higher inaccuracy results if the original dataset exhibits more correlations for the chosen query.

6.3. Results from the CADRMP dataset

Next we report our results on the medical dataset CADRMP. This dataset is comparatively smaller than CENSUS, with only 42,264 data records. The smaller size makes the results less stable compared to CENSUS. We shall again examine the utility for large sums, the privacy for small sums and the comparison with other methods. Since the relative errors in general for CADRMP are lower for small counts, we have attempted to set the default value of γ to 10 as is set in [30]. However, there exists some sensitive value with a frequency above 10%, hence CADRMP violates the eligibility condition [30,20] and Anatomy becomes infeasible. Hence, we set γ to the nearest feasible value of 9.

6.3.1. Utility for large sums

The results for utility are shown in Fig. 11. We plot the relative errors for increasing selectivity from [1%, 2%), [2%, 3%), ..., to [8%, 9%). The relative errors range from below 10% to below 30%. The trend is not clear as the greatest errors occur in the middle at selectivity $[5\%, 6\%)$, and is the lowest at selectivity $[6\%, 7\%)$. The errors from the three methods, *SPLU*-Gen, Anatomy, and *DBR* are quite similar in trend. One explanation to this fluctuation is that the given dataset has a smaller size and the cardinality of the sensitive value is large, which introduces more uncertainty for the errors. The cardinality of Disease in CADRMP is 1346. Compared to the cardinality of 50 for Occupation in CENSUS, Disease is much more diverse. This makes the results more unstable across the different

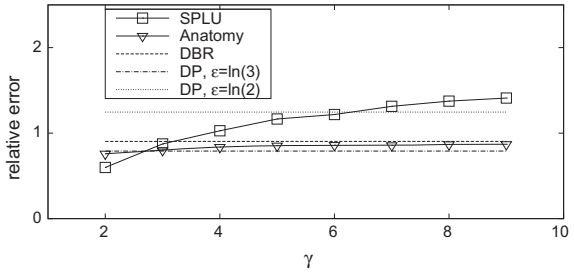


Fig. 10. Comparison of errors for *SPLU*-Gen, Anatomy, *DBR*, and differential privacy (small counts) for CADRMP.

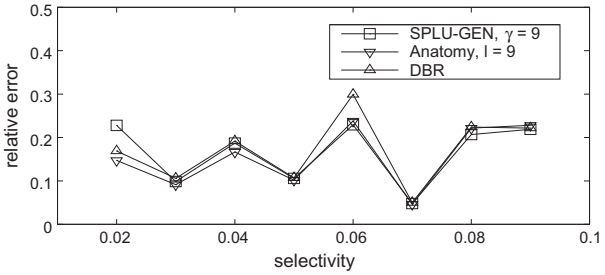


Fig. 11. Comparison of utilities for *SPLU*-Gen, Anatomy, and *DBR* for CADRMP.

methods. Hence, users may first analyze the diversity of their data attributes and take note of its possible impact on the results. Overall, the relative errors for *SPLU*-Gen and Anatomy are both small for the large sum queries and both methods provide utility for such queries.

6.3.2. Privacy for small sums

We vary γ from 2 to 9 in Fig. 10 to measure the relative error for small count queries. The relative errors from *SPLU*-Gen, Anatomy, DBR and DP are plotted. The resulting trend is similar to the results from CENSUS. *SPLU*-Gen has a clearly higher relative error rate compared to the other methods at $\gamma = 4$ or above, and hence provides better small sum privacy.

In Fig. 10 we also show the results of querying with ϵ-differential privacy [14]. As stated in [13], ϵ may be set to values such as 0.01, 0.1, or in some cases, $\ln 2$ or $\ln 3$. We have plotted the result of relative errors for small count queries (count between 1 and 10) in Fig. 10 for the cases of $\Delta f = 1$ and $\epsilon = \ln 3$ and also $\epsilon = \ln 2$. We can see that for $\epsilon = \ln 3$, the relative error is quite small compared to the other methods, and can lead to privacy concerns. If ϵ is set smaller the relative error will become bigger. This result is similar to that for CENSUS.

6.4. Multiple sensitive values

We also consider the utility in scenarios where a query involves more than one sensitive value. To this end, we choose Age and Occupation as the sensitive attributes in CENSUS. The two sensitive attributes are randomized independently and then combined for data publishing To allow queries of large selectivities, we first generalize the domain of Age into ten intervals; without this step, most of the resulting counts are too small and the range of selectivity is limited. The relative error for multiple-dimension aggregates

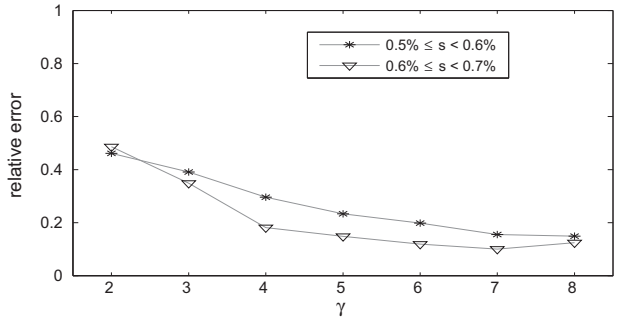


Fig. 12. Utility for *SPLU*-Gen (two *SA*'s).

involving two sensitive attributes is shown in Fig. 12, where γ ranges from 2 to 8. Although the selectivity is very low (0.1–0.7% for this case), the overall accuracy can match that of the single-sensitive-attribute scenario.

6.5. Computational overhead

The computational overhead mainly comes from the partitioning process. We have adopted the partitioning method of Anatomy. This algorithm can be implemented with a time complexity of $O(N(1+\frac{V}{\gamma}))$, where N is the cardinality of the table, and V is the number of distinct values of the sensitive attribute. We measure the running time for the case of single sensitive attribute on the largest 500 K dataset, varying γ from 2 to 10. For all chosen γ values, our algorithm can finish within 10 s for a 500 K dataset, which is practical to be deployed in real applications.

We also consider the querying efficiency at the user side. To estimate the answer, a user will compute each component of the vector y, and do matrix multiplications to iteratively converge to the answer x. When each component of y changes by no more than 1%, we terminate the iteration and measure the querying time and number of iterations. In our experiments, SQLITE3[5] serves for querying y, and we consider the case with two sensitive attributes which involves the most number of components in y, implying the largest computational cost. The result shows that the Bayesian iterative process takes negligible time, while the major cost comes from the querying step. In particular, it takes less than 1 ms on average, and 10 ms in the worst case, for the iterative process to converge. The median and average of the number of iterations is 16 and 325, respectively. In total, the average measured time for a query is 1612 ms, which poses little computational burden on users.

6.6. Discussion on thresholds and parameter setting

As pointed out in Section 2, the thresholds for small sum privacy and large sum utility can be determined by users. The users may decide based on both theoretical analysis and experimental results. With *SPLU*-Gen, we have the properties as given in Lemmas 1 and 2, and some analytical results are shown in Figs. 4 and 5. The figures show the utility for large sums of 100 or above for different values of ϵ, and privacy guarantee for small sums of less than 10. Our experimental results in this section support similar findings. The user can decide on the suitable thresholds based on these results. We have set the parameter of γ to values of 5, 9, and 10 with satisfactory outcomes. Hence, we would recommend the use of γ values between 5 and 10 for *SPLU*-Gen.

[5] See http://docs.python.org/library/sqlite3.html.

7. Related works

Since the pioneering works on protecting privacy while disclosing information in [25,22] there have been many interesting research works. The models of k-anonymity [26] and ℓ-diversity [20] are among the first major models on this problem. ϵ-differential privacy has been introduced for query answering and a common technique is based on distortion to the query answer by a random noise that is i.i.d. from a Laplace distribution and calibrated to the sensitivity of the querying [14,12]. Laplace noise has been used in recent works on reducing relative error [29] and the publishing of data cubes in [9].

Some previous works study the trade-off between utility and privacy (risk). They include the work on risk-utility (RU) confidentiality maps by Duncan et al. [11] and the study of utility versus privacy trade-off in k-anonymization by Loukides et al. [19]. These works set up different metrics for privacy and utility, so that adjustments of parameters in the anonyimzation process give rise to different utility and privacy measurements. Typically, higher utility leads to lower privacy, and vice versa. Based on these metrics, Duncan et al. generate RU plots to identify the *best* solution in terms of both privacy and utility. Such works differ from ours in that they treat all concepts equally in terms of privacy or utility. There is no separation of concepts that need protection versus those that do not. After the parameter setting, these works follow conventional mechanisms and are susceptible to the problem posted by [7].

The problem posted by [7] is that a classifier built on top of data anonymized by conventional mechanisms can be very accurate in the prediction of sensitive values for some individuals. This is true even when the anonymization process is known to have certain privacy protection for the individuals. In their experiment, they showed that with the Adult dataset from the UCI Machine Learning repository, when the data is sanitized by a differential privacy mechanism, a classifier built on top of the published data can be very accurate for a few hundred individuals when the target attribute is Occupation.

In the literature of statistical databases, the protection of small counts has been well-studied in the topic of security in statistical databases [1]. A concept similar to ours is found in [27] where the aim is to ensure that the error in queries involving a large number of tuples will be significantly less than the perturbation of individual tuples. It has been pointed out in previous works [15,16] that the security of a database is endangered by allowing answers to counting queries that involve small counts. However, these previous works are about the secure disclosure of statistics from a dataset and do not deal with the problem of sanitization of a dataset for publishing.

Some significant impossibility results have been found in previous works. In [10], it is shown that any algorithm which guarantees an absolute query estimation error of $o(\sqrt{N})$, for a dataset size of N, for more than $\Omega(N\log^2 N)$ queries will not guarantee privacy. In [21], it is shown that utility is impossible for privacy preservation if the attacker's prior knowledge is above a certain threshold, where the prior knowledge is based on the prior probability of a certain tuple in the dataset. Both impossibility results assume uniform utility requirements for all queries and hence they do not contradict our result for non-uniform utilities.

Randomization technique has been used in previous works in privacy preservation, such as [2,4]. The usefulness of such a technique is verified in [3], where the randomized data is shown to have similar utility as the original dataset for classification.

Cormode et al. [8] suggested the notion of empirical privacy. In this work, privacy is defined to be the fraction of tuples for which one can predict the correct SA value from the published data. In the attack model, a classifier is built to determine the empirical privacy. A comparison of syntactic anonymity and differential privacy is given in [6], with the conclusion that both methodologies have

their pros and cons, each holding its value. Other studies consider combining the strengths of the two methodologies. It is found that t-closeness can yield ϵ-differential privacy when $t = exp(\epsilon)$ in [23], and k-anonymity can improve the utility of anonymized data released by differential privacy techniques [24].

8. Conclusion

While the results in [7] raise an alarm on the existing sanitization models, our finding shows the possibility that existing methods may have covered the needed ground of privacy, and the uncovered ground is not meant to be covered but instead is meant for data utility, which is the goal of privacy preserving data publishing. We introduce the model of SPLU which advocates the support of utility for large sum querying and the guarantee of privacy protection by introducing high inaccuracy for small sum querying. We propose a data sanitization mechanism, called *SPLU*-Gen, based on randomized perturbation on the sensitive values of the given dataset. We show that *SPLU*-Gen satisfies the requirements of the SPLU model. Our empirical studies on two real datasets show desirable performance of our method. For future work, it will be interesting to examine other existing privacy preserving mechanisms to consider how they can be extended to support the paradigm of SPLU.

Acknowledgements

We are very thankful to the reviewers for their careful checking and very helpful comments and suggestions, which greatly improved the manuscript. We thank the authors of [30] for the source code of Anatomy. This research is supported by the Hong Kong UGC/RGC GRF grant 412313 and the CUHK RGC Direct Grant Allocation 2050497.

References

[1] Adam NR, Wortmann JC. Security-control methods for statistical databases: a comparative study. ACM Comput Surv 1989;21(4):515–56.
[2] Agrawal D, Aggarwal CC. On the design and quantification of privacy preserving data mining algorithms. In: ACM PODS; 2001. p. 247–55.
[3] Agrawal R, Srikant R. Privacy-preserving data mining. In: SIGMOD. ACM Press; 2000. p. 439–50.
[4] Agrawal R, Srikant R, Thomas D. Privacy preserving olap. In: SIGMOD; 2005. p. 251–62.
[5] Chaytor R, Wang K. Small domain randomization: same privacy, more utility. In: VLDB; 2010. p. 608–18.
[6] Clifton C, Tassa T. On syntactic anonymity and differential privacy. Trans Data Privacy 2013;6(2):161–83.
[7] Cormode G. Personal privacy vs population privacy: learning to attack anonymization. In: KDD; 2011. p. 1253–61.
[8] Cormode G, Procopiuc CM, Shen E, Srivastava D, Yu T. Empirical privacy and empirical utility of anonymized data. In: ICDE workshop on privacy-preserving data publication and analysis (PRIVDB); 2013. p. 77–82.
[9] Ding B, Winslett M, Han J, Li Z. Differentially private data cubes: optimizing noise sources and consistency. In: SIGMOD; 2011. p. 217–28.
[10] Dinur I, Nissim K. Revealing information while preserving privacy. In: PODS; 2003. p. 202–10.
[11] Duncan GT, Keller-McNulty SA, Stokes SL. Disclosure risk vs data utility: the R – U confidentiality map. In: Technical report number 121, National Institute of Statistical Sciences; December 2001.
[12] Dwork C. Differential privacy. In: International colloquium on automata, languages and programming (ICALP); 2006. p. 1–12.
[13] Dwork C. A firm foundation for private data analysis. Commun ACM 2011;54(1):86–95.
[14] Dwork C, McSherry F, Nissim K, Smith A. Calibrating noise to sensitivity in private data analysis. In: Proc 3rd theory of cryptography conference; 2006. p. 265–84.
[15] Feliegi I. On the question of statistical confidentiality. J Am Stat Assoc 1972;67(337):7–18.
[16] Hoffman L, Miller WF. Getting a personal dossier from a statistical data bank. Datamation 1970;16(5):74–5.
[17] Hurkens CJ, Tiourine S. Models and methods for the microdata protection problem. J Off Stat 1998:437–47.
[18] Kifer D. Attacks on privacy and definetti's theorem. In: SIGMOD; 2009. p. 127–38.
[19] Loukides G, Shao J. Data utility and privacy protection trade-off in k-anonymisation. In: PAIS '08 proceedings of the 2008 international workshop on privacy and anonymity in information society; 2008. p. 36–45.
[20] Machanavajjhala A, Gehrke J, Kifer D. l-diversity: privacy beyond k-anonymity. In: ICDE; 2006. p. 24.
[21] Rastogi V, Suciu D, Hong S. The boundary between privacy and utility in data publishing. In: VLDB; 2007. p. 531–42.
[22] Samarati P. Protecting respondents' identities in microdata release. IEEE Trans Knowl Data Eng 2001;13(6).
[23] Soria-Comas J, Domingo-Ferrer J. Differential privacy via t-closeness in data publishing. In: 11th IEEE annual conference on privacy, security and trust-PST; 2013. p. 27–35.
[24] Soria-Comas J, Domingo-Ferrer J, Sánchez D, Martínez S. Improving the utility of differentially private data releases via k-anonymity. In: 12th IEEE international conference on trust, security and privacy in computing and communications; 2013. p. 371–9.
[25] Sweeney L. Weaving technology and policy together to maintain confidentiality. J Law Med Ethics 1997;25(2–3):98–110.
[26] Sweeney L. k-anonymity: a model for protecting privacy. Int J Uncertain Fuzz Knowl Syst 2002;10(5):557–70.
[27] Traub J, Yemini Y, Wozniakowski H. The statistical security of a statistical database. ACM Trans Database Syst 1984;9(4).
[28] Wong R, Fu A, Wang K, Xu Y, Yu P, Pei J. Can the utility of anonymized data be used for privacy breaches. ACM Trans Knowl Discovery Data 2011;5(3):1–23. 39.
[29] Xiao X, Bender G, Hay M, Gehrke J. ireduct: differential privacy with reduced relative errors. In: SIGMOD; 2011. p. 229–40.
[30] Xiao X, Tao Y. Anatomy: simple and effective privacy preservation. In: VLDB; 2006. p. 139–50.