**k-Presence-Secrecy: Practical Privacy Model as Extension of k-Anonymity**

Yuji YAMAOKA(a), Nonmember and Kouichi ITOH(b), Member

**SUMMARY**  PPDP (Privacy-Preserving Data Publishing) is technology that discloses personal information while protecting individual privacy. k-anonymity is a privacy model that should be achieved in PPDP. However, k-anonymity does not guarantee privacy against adversaries who have knowledge of even a few uncommon individuals in a population. In this paper, we propose a new model, called k-presence-secrecy, that prevents such adversaries from inferring whether an arbitrary individual is included in a personal data table. We also propose an algorithm that satisfies the model. k-presence-secrecy is a practical model because an algorithm that satisfies it requires only a PPDP target table as personal information, whereas previous models require a PPDP target table and almost all the background knowledge of adversaries. Our experiments show that, whereas an algorithm satisfying only k-anonymity cannot protect privacy, even against adversaries who have knowledge for one uncommon individual in a population, our algorithm can do so with less information loss and shorter execution time.

**key words:** privacy-preserving data publishing, k-anonymity

1. Introduction

Recently, expectations for PPDP (Privacy-Preserving Data Publishing) have increased [1]. PPDP is the conversion of personal data for disclosing as much information as possible while protecting individual privacy. A particularly important application of PPDP is the processing of de-identified information. For example, the Act on the Protection of Personal Information in Japan has been revised to accept the provision of de-identified information without individual consent. Useful knowledge may be obtained from various disclosed personal data through collecting them and mining data from them. However, PPDP is necessary because data holders must consider privacy; thus, raw data cannot be disclosed.

k-anonymity [2] and its extensions, like ℓ-diversity [3], are actively researched as privacy models that should be achieved in PPDP. They have been proposed on the grounds that removing obvious personally identifiable information, like names or national identification numbers, are insufficient to protect privacy. For example, it has been reported that more than 60% of American individuals can be identified by a combination of their sex, birth date, and 5-digit zip code [2], [4].

Many k-anonymity-based studies [2], [3], [5]–[12] assume that target data are tables that consists of tuples, where each tuple is the data from an individual, and table attributes are classified into the following categories:

**QIs (quasi-identifiers)** are attributes that can identify individuals.

**SAs (sensitive attributes)** are attributes that include sensitive information.

Moreover, we adopt the common assumption that an individual’s data is included in at most one tuple. A group containing tuples that have the same QI values is called an EC (equivalence class). If the size of each EC in a table is at least k, it is said that the table satisfies k-anonymity. If adversaries know that data from a specific individual are included in the table, k-anonymity does not guarantee that they will be prevented from inferring the individual’s SA. This problem is called attribute linkage [1].

By using knowledge of the k-anonymity algorithm, there are cases in which adversaries can deduce tuples in the target table in accordance with properties of the k-anonymity algorithm to achieve minimum information loss. This attack is called a “minimality attack” [13]. k-anonymity does not mitigate attribute linkage when faced with such attacks because adversaries can know that data from a specific individual is included in the k-anonymized table. Note that the minimality attack was originally reported as an attack on ℓ-diversity, and then for k-anonymity [14]. In this paper, we address the minimality attack as an attack on k-anonymity. A minimality attack is based on knowledge of the k-anonymity algorithm used on the target table. Although a minimality attack requires knowledge of the k-anonymity algorithm, protection against these attacks is important. For example, the amended Act on the Protection of Personal Information in Japan requires that business operators clearly indicate that information provided to third parties has been de-identified; publishing the k-anonymity algorithm is an effective way to satisfy this requirement.

ℓ-diversity was proposed to mitigate attribute linkage. If the proportion of each value of each SA in each EC in the table is at most 1/ℓ, it is said that the table satisfies ℓ-diversity. Various algorithms [3], [5]–[11] have been proposed that make raw tables satisfy k-anonymity or ℓ-diversity, and they have been evaluated by the utilities of the post-conversion tables (e.g., information loss) and the

---

Manuscript received June 22, 2016.
Manuscript revised November 7, 2016.
Manuscript publicized January 17, 2017.

The authors are with FUJITSU LABORATORIES LTD., Kawasaki-shi, 211–8588 Japan.

†This paper was presented at the 32nd Symposium on Cryptography and Information Security, in Fukuoka, Japan, January 2015.

a) E-mail: yamaoka.yuji@jp.fujitsu.com
b) E-mail: ito.kouichi@jp.fujitsu.com
DOI: 10.1587/transinf.2016DAP0015

Copyright © 2017 The Institute of Electronics, Information and Communication Engineers
computational efficiency of the algorithm. However, it is actually difficult to classify attributes as QIs or SAs [14]. For example, although the “zip code” attribute is normally classified as a QI, it is sensitive information for some individuals (for example, those who are threatened by stalkers). Conversely, although the “disease” attribute is normally classified as an SA, some individuals may disclose their own or family members’ diseases on social media, thus making this information a QI. In these cases, $\ell$-diversity cannot be used because attribute linkage is still possible.

In the following, we assume that attributes cannot be classified into QIs and SAs. Accordingly, we reconsidered attribute linkage as a problem in which adversaries who know part of a specific individual’s data can infer the remainder of these data by seeing a table.

$\delta$-presence [15] was proposed as another model that mitigates attribute linkage. $\delta$-presence achieves the following two properties:

**Hiding presence**, so that adversaries cannot determine that the table includes a specific individual’s data.

**Hiding absence**, so that adversaries cannot determine that the table does not include a specific individual’s data.

Hiding presence mitigates attribute linkage because adversaries cannot infer the attributes of individuals’ data that are not included in the table. Therefore, $\delta$-presence mitigates attribute linkage. However, $\delta$-presence is not practical [1] because it requires the PPDP executor to have almost all the adversaries’ background knowledge. That is, the PPDP executor must know the probability distribution of the attribute or combination of attributes that fits the adversaries’ background knowledge.

To solve this problem, we propose a model that achieves both hiding presence and hiding absence without detailed adversary background knowledge. Hiding absence is easily achieved by suppressing some records or sampling. Therefore, our main objective is hiding presence. Our proposed model is based on $k$-anonymity. $k$-anonymity has a partial presence-hiding effect because $k$-anonymity tables have no tuples that include uncommon (i.e., fewer than $k$ instances) values in the QIs.

However, many existing algorithms satisfying $k$-anonymity are vulnerable if known by the adversaries. A minimality attack [13] can be used by adversaries who employ algorithm knowledge to partly invert the algorithm conversion and breach $k$-anonymity. These attacks can be performed because $k$-anonymity is a property for the table, not for the algorithm. Thus, $k$-anonymity does not guarantee hiding presence because it may be breached by a minimality attack.

We therefore improve $k$-anonymity to guarantee hiding presence, even when the algorithm is known by adversaries. We propose a new privacy model and algorithm that hide presence and absence by improving $k$-anonymity through the introduction of randomness to the generalization. In contrast with other works [16]–[18], noise is not added. That is, no false information is included in the post-conversion table, which enables the acquisition of useful information from the table. (Detail of these other works [16]–[18] are described in Sect. 6.) Our model is practical because the algorithm requires only the table to be processed as personal information, whereas $\delta$-presence needs detailed adversary background knowledge.

1.1 Privacy Breach by Minimality Attack

Table 1 shows an example of personal medical data in a table whose elements are one tuple per individual. A cell value of the “Zip Code” attribute of a tuple, for example, represents the zip code of the individual corresponding to the tuple. This table may cause privacy issues although it has no obvious identifiable attributes, such as names. For example, let us assume that adversaries know that Ms. X is the only individual whose zip code is “998,” age is “20–40,” and sex is “female.” If adversaries see the table, they can infer that tuple 1 is Ms. X’s data, and therefore that Ms. X has lung cancer, which is a breach of privacy if Ms. X does not want others to know about her disease. Therefore, PPDP is necessary to prevent such privacy breaches.

Let us also assume that there are many individuals whose zip code is “998,” age is “20–40,” and disease is “lung cancer;”—the same as Ms. X—but whose sex is “male.” Here, if adversaries see the table, they cannot be certain to whom the data in tuple 2 belongs. We can therefore see that individuals who have uncommon QI values need be protected.

Table 2 shows an example in which Xu et al.’s algorithm (the Top-Down method) [8] converts Table 1 to satisfy $k$-anonymity ($k = 2$). The conversion is performed by cell suppression, that is, converting cell values to “*.” There are at least 2 identical tuples for every tuple in the post-conversion table in Table 2 and no tuples can be identified as Ms. X’s data.

However, with a minimality attack, adversaries can still infer that Ms. X has lung cancer from Table 2. If adversaries who know the algorithm see Table 2, (and if they know that each tuple of the pre-conversion table has either “female” or “male” in the “sex” attribute), they can infer that tuples 1 and 5 are Ms. X’s data. Therefore, PPDP is necessary to prevent such privacy breaches.

**Table 1** Example of personal data (“No.” is an attribute added for explanation purposes and does not actually exist within the table)

| No. | Zip Code | Age | Sex | Disease     |
|-----|----------|-----|-----|-------------|
| 1   | 998      | 20–40 | female | lung cancer |
| 2   | 998      | 20–40 | male  | lung cancer |
| 3   | 998      | 60–80 | female | cold        |
| 4   | 998      | 60–80 | male  | influenza   |
| 5   | 998      | 60–80 | male  | cold        |

**Table 2** Example of $k$-anonymity, after converting Table 1 with $k = 2$

| No. | Zip Code | Age | Sex | Disease     |
|-----|----------|-----|-----|-------------|
| 1   | 998      | 20–40 | *   | lung cancer |
| 2   | 998      | 20–40 | *   | lung cancer |
| 3   | 998      | 60–80 | *   | *           |
| 4   | 998      | 60–80 | *   | *           |
| 5   | 998      | 60–80 | *   | *           |
2 contain both “female” and “male”, because Xu et al.’s algorithm does not convert tuples if they satisfy \(k\)-anonymity; hence, there must be multiple values in the pre-conversion tuples if the cells have been converted. Thus, the adversaries can infer that either tuple 1 or 2 is Ms. X’s data and that Ms. X has lung cancer. Note that while it may not be a privacy issue that adversaries can infer that the original values of the “#”s in the “sex” attribute include “female”, it must be privacy issue that they can infer a disease that a specific individual has.

From the above, we can see:

- Satisfying \(k\)-anonymity prevents tuples from having uncommon QI values; however, the reversibility of the algorithm may cause a breach of its protection.

1.2 Contributions and Organization

We propose a new privacy model and a new algorithm that prevent adversaries who know a few uncommon individuals in a population and know the algorithm used from inferring presence and absence with high probability. Our contributions are as follows.

First, we propose the first model that requires a PPDP algorithm to hide presence for only uncommon individuals, based on the premise that the algorithm is open. We call the model \(k\)-presence-secrecy. Even if adversaries who know a few individuals in a population whose QI values are uncommon (i.e., possessed by fewer than \(k\) individuals) and know the algorithm see a table that has been converted by an algorithm that satisfies \(k\)-presence-secrecy, the adversaries cannot infer the presence and absence of any individuals with high probability. \(k\)-presence-secrecy deals only with QIs; it is not necessary to forcibly classify any attributes that are potentially QI into SA. This model is practical because the target table is sufficient as personal information and detailed adversary background knowledge is not needed. Moreover, the PPDP executor can benefit from algorithm disclosure.

Second, we propose an algorithm that satisfies \(k\)-presence-secrecy. Our algorithm realizes irreversible \(k\)-anonymity by applying a conversion that satisfies \(k\)-anonymity irreversibly for a single attribute to all attributes by introducing randomness to the generalization. We show experimental results on real personal data compared with the most competent existing algorithm for \(k\)-anonymity, and show that our algorithm hides presence and absence and causes less information loss in a shorter computation time.

The remainder of the paper is organized as follows: We introduce notation and define \(k\)-anonymity and explore its problems in Sect. 2. We propose our new privacy model \(k\)-presence-secrecy in Sect. 3 and our new algorithm that satisfies \(k\)-presence-secrecy in Sect. 4. We show experimental results in Sect. 5. We review related work in Sect. 6, and outline our conclusions in Sect. 7.

2. Problems with \(k\)-Anonymity

In this section we introduce some basic notation, define \(k\)-anonymity, and discuss the flaws of \(k\)-anonymity.

Let \(t\) be a tuple that denotes the information from one individual with attributes \(A = \{a_1, a_2, \ldots , a_m\}\). Let \(T = \{t_1, t_2, \ldots , t_n\}\) be a table that is a multiset of such tuples. All the attributes in \(A\) are regarded as QI for \(k\)-anonymity, even if they may include sensitive attributes. Let \(U \supseteq T\) be a population and also a multiset of tuples with attributes \(A\).

We assume that individuals who correspond to \(t \in U\) are different from each other. Let \(o(t)\) be an individual who corresponds to \(t\) for whom the following holds: \(\forall t_i \in U, \forall t_j \in U, i \neq j \Leftrightarrow o(t_i) \neq o(t_j)\). \(k\)-anonymity is defined as follows:

**Definition 1** (\(k\)-anonymity). Let the EC of each \(t_i \in T\) be an EC that is a multiset \(\{t \in T, \forall a \in A, t[a] = t_i[a]\}\). A table \(T\) satisfies \(k\)-anonymity if each EC has a size of at least \(k\).

The assumed adversary knowledge in \(k\)-anonymity, \(K_1\), is the following:

**Definition 2** (adversaries’ knowledge \(K_1\)). The adversaries’ knowledge, \(K_1\), is as follows.

- The post-conversion table \(T' = f_A(k, T)\)
- Arbitrary tuples in the population \(E \subseteq U\) and corresponding individuals \(I = \{i_1, i_2, \ldots , i_E\}\) (\(\forall e_j \in E, i_j = o(e_j)\))

where, \(T'\) satisfies \(k\)-anonymity and \(f_A\) is the algorithm satisfying \(k\)-anonymity.

A basic goal of \(k\)-anonymity is to prevent adversaries who have knowledge \(K_1\) from narrowing down individual candidates for each tuple of the post-conversion table \(T'\) to less than \(k\).

\(k\)-anonymity does not guarantee the prevention of the following [1]:

**Attribute linkage** links specific individuals with sensitive information.

**Minimality attacks** allow adversaries with knowledge of the conversion algorithm infer pre-conversion values.

PPDP should guarantee the prevention of attribute linkage, because it is direct breach of privacy; two main measures have been proposed for doing so. One is a method represented by \(\ell\)-diversity that classifies attributes as QI or SA and diversifies of SA values. The other is a method represented by \(\delta\)-presence that hides presence. Because we believe it is difficult to classify attributes as QI or SA as described above, we adopted the latter. Moreover, we propose a model that does not require personal information other than the pre-conversion table \(T\), unlike \(\delta\)-presence.

If conversion algorithms are highly reversible, a minimality attack can cause a privacy breach. Reversible algorithms tend to be designed if adversaries’ knowledge about
the algorithms is unclear. A measure to eliminate this weakness is to make algorithm disclosure a premise of the model. For example, Wong et al. [10] redefined k-anonymity with the precondition that adversaries know the algorithm.

We propose a privacy model that includes both measures: hiding presence and assuming algorithm disclosure.

3. k-Presence-Secrecy

In this section, we propose k-presence-secrecy as a new privacy model. We first define the adversaries’ knowledge, and then the model against the adversaries.

The assumed knowledge of the adversaries, K2, in k-presence-secrecy is as follows:

Definition 3 (adversaries’ knowledge K2). Adversaries’ knowledge, K2, is:

- The conversion algorithm fA, the parameter k
- The post-conversion table T' = fA(k, T)
- Arbitrary but not all tuples in the population E− ⊆ U, and corresponding individuals I = {i1, i2, ..., ik-1} (∀e \in E−, ij = e(j))
- The set D of all combinations of attribute values that can be in the pre-conversion table T; D := d(U), D ≠ U, |D| > 1, where d is a function that converts a multiset to a set by removing the multiplicity of the multiset
- The multiset R of uncommon values that are had by less than k individuals in the population; R ⊆ U, |R| < k, ∀r ∈ R, r \notin (U − R)

We assume that fA and k are included in K2 because disclosure of the algorithm and parameters has several advantages:

- The individual data owners can feel assess the algorithm and decide whether it is reliable.
- Data users can improve accuracy of their data analyses because they can know the details of information loss.
- Data holders can manage easily because they do not need to be concerned with algorithm leakage prevention.

Additionally, it is undesirable to rely on “security through obscurity” as a general consideration. By including the algorithm in the adversaries’ knowledge, the privacy model comes to require measures against minimality attacks.

We assume that T' is included in K2 because we assume that data users can be adversaries, as in preceding studies.

From these assumptions, the algorithm must be irreversible. If adversaries can derive an inverse conversion of the algorithm, they can obtain the pre-conversion table by T = fA−1(k, T'). Even if they can derive a part of the inverse conversion, a minimality attack may have original values of T' and may breach privacy.

We assume that E− is included in K2 because there are real cases where such knowledge can be easily obtained. For example, Sweeney [2] reported that many individuals’ QI values can be obtained from voter registration records. We assume E− ⊆ U, that is, that adversaries do not know tuples of all individuals in the population. That means adversaries may know arbitrary tuples but do not know their rarity, that is, the number of corresponding individuals. Adversaries normally do not know all tuples in the population. In Sweeney’s work, voter registration records were used as adversaries’ knowledge in PPDP of medical records, and cannot all be in both populations. K2 includes E− ⊆ U, whereas K1 includes E ⊆ U; this difference is the only knowledge that K2 does not include but K1 does.

We assume that D is included in K2, because the combination of attribute values that can be in T depends on the schema and population; therefore, it is difficult to model it flexibly. An example of such knowledge is that the only values in the “sex” attribute are likely to be “female” and “male,” and that only males (“male” in the “sex” attribute) have prostate cancer (“prostate cancer” in the “disease” attribute), etc. Thus, adversaries know the combination of attribute values that can not be in T. For example, there cannot be tuples that have both “female” and “prostate cancer.” It is reasonable to let adversaries know such information because it is often common knowledge or information that can be easily known. We assume D ≠ U and |D| > 1. If A does not include any ID attributes and A contains many tuples, these assumptions are normally correct.

We assume that R is included in K2 because it is necessary to protect only the minority of the population as described above, and the model has affinity with k-anonymity. The minority can be protected adequately by letting adversaries know R. Adversaries may often know a small minority or a unique individual in the population as R.

In Sweeney’s work, for example, the “ZIP” attribute is used as a QI, and adversaries may infer the “ZIP” values of a small minority or a unique individual in a depopulated area. Because |R| is less than k and there are no uncommon tuples (whose count is less than k) in a k-anonymity-satisfied table, the model has affinity with k-anonymity. One of k-anonymity’s assumptions is that the value of each tuple whose count is at least k is popular, and not private, information. Hence, it is reasonable that R models that adversaries do not know the number of individuals who have popular information.

Definition 4 (k-presence-secrecy). An algorithm fA satisfies k-presence-secrecy if adversaries who have K2 cannot infer the presence and absence of arbitrary individuals with high probability. Such fA satisfies ∀e ∈ E−, 0 ≤ Pr[e ∈ T|K2] ≤ 1. Note that K2 depends on k as shown in Definition 3.

It is necessary for the algorithm not only to satisfy k-presence-secrecy but to lessen information loss for PPDP.

4. Algorithm

In this section, we propose an algorithm that satisfies k-presence-secrecy.

Suppression is a conversion that generalizes a cell
value to a suppressed value “∗”. A tuple for which each attribute value is suppressed is called a suppressed tuple.

The only conversion methods used by our algorithm (other than sorting) is suppression. In this case, the owners of tuples are maintained. Therefore, it is the same to infer whether T′ includes a specific individual’s data and to infer presence and absence.

In this section, we explain hiding presence, hiding absence, and the algorithm that integrates them.

4.1 Hiding Presence

First, we consider a method to prevent adversaries from inferring presence with high probability.

If adversaries cannot know the rarity of any tuples, they cannot infer presence with high probability, because they cannot know the number of corresponding individuals.

We introduce suppression-considered k-anonymity as follows:

**Definition 5** (Suppression-considered k-anonymity). A table T satisfies suppression-considered k-anonymity if each EC has at least k tuples or consists of suppressed tuples.

Adversaries cannot infer presence for a suppression-considered k-anonymity-satisfied table with high probability if they cannot infer the original values of suppressed values with high probability with K_2. For nonsuppressed tuples, the EC size of each tuple is at least k and the adversaries do not know the number of corresponding individuals. For suppressed tuples, the adversaries cannot infer their original values with high probability.

Therefore, we can prevent adversaries from inferring presence with high probability if they cannot infer the original values of suppressed values with high probability using K_2, and if the algorithm satisfies suppression-considered k-anonymity.

We initially assume |A| = 1. An algorithm that prevents adversaries from inferring the original values of suppressed values with high probability and that satisfies suppression-considered k-anonymity for a table T that has attribute A = {a} is called (a, k)-presence-hiding. As notation, for example, 5-tuple table that consists of 4 individuals who have value “x” and 1 individual who has value “y” is written as T = {{"x":4, "y":1}}.

First, we demonstrate that k-presence-secrecy is not satisfied by algorithms that suppress only uncommon tuples deterministically. Then, we propose an algorithm that suppresses not only uncommon tuples deterministically, but also some common tuples probabilistically.

We explain a case in which k = 3 and T_1 = \{"x":4, "y":1\}. Algorithms that convert T_1 to \{"∗∗∗":5\} with high probability are not practical because of excessive information loss. Hence, a better algorithm must convert T_1 to T_1′ = \{"x":4, "∗∗":1\} with a certain probability. In this case, with D = \{"x", "y"\} and R = \{"y":1\} (i.e., only one individual in the population has “y,” and the others have “x”), the algorithms by which the pre-conversion table of T_1′ is T_1 with high probability do not satisfy k-presence-secrecy, because adversaries can infer presence of the individual who has “y” with high probability. Hence, algorithms that convert T_1 to T_1′ must convert \{"x":5\} to T_1′ with a certain probability.

Similarly, with k = 3, T_2′ = \{"x":3, "∗":2\}, D = \{"x", "y"\}, and R = \{"y":2\}, the algorithms for which the pre-conversion table of T_2′ is T_2 = \{"x":3, "y":2\} with high probability do not satisfy k-presence-secrecy. Hence, the algorithms that convert T_2 to T_2′ must convert \{"x":4, "y":1\} or both to T_2′ with a certain probability.

Similarly, with k = 3, T_3′ = \{"x":3, "∗∗":4\}, D = \{"x", "y", "z"\}, and R = \{"y":2\}, the algorithms for which the pre-conversion table of T_3′ is T_3 = \{"x":3, "y":2, "z":2\} with high probability do not satisfy k-presence-secrecy. Hence, algorithms that convert T_3 to T_3′ must convert tables including four or more “x’s,” such as \{"x":5, "y":2\} to T_3′ with a certain probability. Note that the PPDP executor may not know D and R.

Hence, adversaries can be prevented from inferring presence with high probability by additionally suppressing some values (tuples) whose count (EC size) is k or more with a certain probability.

We propose an algorithm that satisfies (a, k)-presence-hiding in Algorithm 3. Algorithm 3 consists of the following steps: suppressing each value whose count is less than k, suppressing c additional values where c is calculated by parameter p and a random number, and suppressing each value whose count is less than k.

The parameter 0 < p < 1 corresponds to randomness; the larger the value is, the lower the probability with which adversaries can infer presence is and the larger information loss is. The additional number of cells that are suppressed is calculated using c ← [log_2 u] where u is a uniform random number in the range (0, 1). For example, with k = 3, T_1′ = \{"x":4, "∗∗":1\}, D = \{"x", "y"\}, and R = \{"y":1\}, the conversion probability for T_0 = \{"x":5\} is Pr[c = 1] = (1 - p), the conversion probability for T_1 = \{"x":4, "y":1\} is Pr[c = 0] = 1 - p, and there are no other conversions. Hence, assuming Pr[T_0] = Pr[T_1] as the prior probability Pr[T] for adversaries, the posterior probability Pr[T_1′|T_1] that the only individual who has the value “y” is included in T is as follows, per Bayes’ theorem:

\[
Pr[T_1′|T_1] = \frac{Pr[T_1′|T_0] Pr[T_0]}{\sum_T Pr[T_1′|T] Pr[T]}
\]

\[
= \frac{Pr[T_1′|T_0] + Pr[T_1′|T_1]}{Pr[T_1′|T_0] + Pr[T_1′|T_1]}
\]

\[
= \frac{1 - p}{(1 - p)p + (1 - p)}
\]

\[
= \frac{1}{1 + p}
\]

Similarly, with T′ = T_2′ = \{"x":3, "∗":2\} and k, D, R are the same as above, and the posterior probability is Pr[T_1′|T_2′] = 1/(1 + p).
We assume that the prior probability $Pr[T]$ is the uniform distribution for adversaries, and we show the maximum probability with which the adversaries can infer presence. We assume that $|D|$ has a minimum value, $|D| = 2$, because the smaller $|D|$ is, the easier inference is. Let a postconversion table in which the number of suppressed values is $m > 0$ be $T_m$, the number of uncommon values $|R| > 0$ be $r$, and a pre-conversion table in which the number of uncommon values $i \leq m$ be $T_i$. Because $Pr[T_m'[T_i] = (1 - p)r^{m-i}$ holds, the following can be determined using Bayes’ theorem:

$$
Pr[T_i|T_m'] = \frac{Pr[T_m'[T_i] Pr[T_i]}{\sum_i Pr[T_m'[T_i]] Pr[T_i]}
$$

$$
= \frac{(1 - p)r^{m-i}}{\sum_i (1 - p)r^{m-i}}
$$

$$
= \frac{1 - p}{p^{i-min(m,r)} - p^{i+1}}
$$

The probability $Pr[e \in T[T_m']$ with which adversaries can infer the presence of an individual’s tuple $e$ is the sum of each prior probability for $T_i$ including $e$. Hence, the following holds:

$$
Pr[e \in T[T_m'] = \sum_{i=1}^{min(m,r)} \frac{1}{r} Pr[T_i|T_m']
$$

$$
= \sum_{i=1}^{min(m,r)} \frac{1 - p}{r p^{i-min(m,r)} - p^{i+1}}
$$

$$
= \frac{1 - p}{1 - p^{r+1}} \sum_{i=1}^{r} (1 - j/r)p^j
$$

If $m \leq r$ holds, then $min(m,r) = m$, and Eq. (1) increases monotonically with decreasing $r$ and reaches its maximum at $r = m$. Therefore, Eq. (1) is maximized at $min(m,r) = r$. Hence, the following holds:

$$
Pr_{max}[e \in T[T'] = \frac{1 - p}{1 - p^{r+1}} \sum_{j=0}^{r-1} (1 - j/r)p^j
$$

Let Eq. (2) be $g(p, r)$, where $g(p, r)$ increases monotonically with increasing $r$, as follows:

$$
g(p, r) = \frac{1 - p}{1 - p^{r+1}} \sum_{j=0}^{r-1} (1 - j/r)p^j
$$

$$
< \frac{1 - p}{1 - p^{r+2}} \sum_{j=0}^{r-1} (1 - j/r)p^j
$$

$$
= \frac{1 - p}{1 - p^{r+2}} \sum_{j=0}^{r-1} (1 - j/r + 1)p^j
$$

**4.2 Hiding Absence**

Next, we consider a method to prevent adversaries from inferring absence with high probability.

This is easily achieved by suppressing one or more tuple randomly. Because $K_2$ does not include the population size, adversaries cannot narrow down the candidates for the original values of suppressed tuples.

We propose that $k$ tuples are randomly suppressed at the beginning of our algorithm provided that each uncommon tuple whose count is less than $k$ is preferentially sup-
pressed. This approach can serve as a protective measure in the case in which adversaries know that \( k-1 \) specific individuals are included in the pre-conversion table. If \( k \) or more suppressed tuples are in the post-conversion table, the adversaries think each of these individuals may correspond to one of the suppressed tuples. Moreover, the post-conversion table satisfies \( k \)-anonymity. Uncommon tuples are preferentially suppressed because sensitive values are often uncommon.

4.3 Satisfying \( k \)-Presence-Secrecy

The summary of our algorithm is as follow:

- Initially, for hiding absence, \( k \) tuples are suppressed randomly provided that uncommon tuples are preferentially suppressed.
- For hiding presence, in which it is difficult and time-consuming to process multiple attributes simultaneously, each attribute is processed sequentially. \( (a, k) \)-presence-hiding is applied to each attribute of a table or subtables. \( (a, k) \)-presence-hiding suppresses not only each cell value whose count is less than \( k \), but also additional cells that are selected probabilistically according to an exponential distribution.

The algorithm generates a post-conversion table in which there are suppressed tuples, there are no uncommon tuples, and the original value of each suppressed cell cannot be inferred with high probability. Although some common cell values may be suppressed, the number of suppressed cells and information loss can be kept small by using an exponential distribution.

The details of our algorithm are shown in Algorithm 1, 2, 3. Algorithm 1 is the entry of the algorithm, and calls Algorithms 2 and 3 as functions. Algorithm 3 is an algorithm that satisfies \( (a, k) \)-presence-hiding, as described above. Algorithm 3 does not suppress tuples but outputs the tuples to suppress, whereas Algorithms 1 and 2 actually suppress them. The sorting at the end of Algorithm 1 is a process that is generally applied for irreversibility in PPDP.

Note that there are side effects in table \( T \), which is the first argument of Algorithm 2. In other words, suppression of table \( T \) is reflected in the caller. Although the parameter \( p \) is used in common among all attributes for brevity, we can use a different value for each attribute.

**Algorithm 1** Our Algorithm

**Input:** a table \( T \), an attribute list \( A \), security parameters \( 1 < k \in \mathbb{N} \) and \( 0 < p < 1 \)

**Output:** a converted table \( T \)

1: suppress \( k \) tuples in \( T \) randomly provided that uncommon tuples are preferentially suppressed
2: \( a \leftarrow \) the first attribute in \( A \)
3: \( S \leftarrow \) anonymize\((T, A, a, \emptyset, k, p)\) \{All tuples except \( S \) are \( k \)-anonymized\}.
4: suppress tuples \( S \) in \( T \)
5: sort all tuples in \( T \)
6: return \( T \)

Algorithm 2 anonymize\((T, A, a, D, k, p)\)

**Input:** a table \( T \), an attribute list \( A \), an attribute \( a \subseteq A \), an attribute set \( D \subseteq A \), security parameters \( 1 < k \in \mathbb{N} \) and \( 0 < p < 1 \) \((D \) is attributes that inhibit \( k \)-anonymity before in \( T \))

**Output:** a multiset of tuples to suppress \{All other tuples in \( T \) satisfy \( k \)-anonymity\}.

1: if \( a = \emptyset \) then
2: suppress all cells of \( D \) in \( T \) \{\( T \) satisfies \( k \)-anonymity\}.
3: return \( \emptyset \)
4: end if
5: \( S \leftarrow \) anonymize\(Attr(T, a, k, p)\) \{\( S \) is the tuples that do not satisfy \( k \)-anonymity in \( a \) in \( T \)\}
6: \( a' \leftarrow \) the attribute following \( a \) in \( A \)
7: for all \( v \) values of \( a \) whose count is \( k \) or more in \( T \) do
8: \( T' \leftarrow \) (a multiset of tuples each of whose value of \( a \) is \( v \) in \( T \))-\( S \)
9: if \( T' \neq \emptyset \) then
10: \( S \leftarrow S \cup \) anonymize\(Attr(T', a, a', D, k, p)\)
11: end if
12: end for\(\)\{All tuples except \( S \) in \( T \) satisfy \( k \)-anonymity.\}
13: if \( |S| < k \) then
14: return \( S \)
15: end if
16: \( D' \leftarrow D \cup a \) \{a is added to inhibiting attributes, then \( S \) with \( a' \) is processed recursively.\}
17: return anonymize\(\)\((S, A, a', D', k, p)\)

Algorithm 3 anonymize\(Attr(T, a, k, p)\)

**Input:** a table \( T \), an attribute \( a \), security parameters \( 1 < k \in \mathbb{N} \) and \( 0 < p < 1 \)

**Output:** a multiset of tuples to suppress \{All tuples except them in \( T \) satisfy \( k \)-anonymity in \( a \)\}.

1: \( S \leftarrow \) a multiset of tuples each of which has a value of \( a \) whose count is less than \( k \) in \( T \)
2: \( u \leftarrow \) a uniform random number in the range \((0, 1)\)
3: \( c \leftarrow \lceil \log_2 u \rceil \)
4: \( S \leftarrow S \cup \) (\( c \) tuples selected randomly from \( T - S \))
5: \( S \leftarrow S \cup \) (a multiset of tuples each of which has a value of \( a \) whose count is less than \( k \) in \( T - S \))
6: return \( S \)

| No. | Zip Code | Age | Sex | Disease |
|-----|----------|-----|-----|---------|
| 1   | *        | *   | *   | *       |
| 2   | *        | *   | *   | *       |
| 3   | *        | *   | *   | *       |
| 4   | 998      | 60–80 | male | *       |
| 5   | 998      | 60–80 | male | *       |

Table 3 shows an example of our algorithm’s output with \( k = 2 \) for Table 1. As input, we used \( A = [\text{Zip Code, Age, Sex, Disease}], p = 0.3 \). The output of our algorithm satisfies \( k \)-anonymity and is changed randomly. Hence, adversaries who only have \( K_2 \) cannot infer the presence of arbitrary individuals with high probability.

5. Experiment

We confirmed that our algorithm has practical performance and better information loss than an existing typical \( k \)-anonymization algorithm by applying it to publicly available data.
We used the Adult dataset from the UCI Machine Learning Repository\textsuperscript{7} and pseudo data generated from it as target tables.

From the Adult dataset, we used the 45,222 tuples that do not include unknown values. We generated pseudo data using the frequency distribution of each attribute of that do not include unknown values. We generated pseudo tables.

We implemented two algorithms for comparison. The first is Xu et al.’s algorithm (the Top-Down method)\textsuperscript{8}, which we chose because it is representative of cell generalization schemes\textsuperscript{11} that can minimize information loss generally. While Ghinita et al. proposed an algorithm\textsuperscript{9} with less information loss, they did not report details of their algorithm parameters. Since we could not set the optimal parameter values, we did not compare our algorithm with theirs directly. Since Ghinita et al. also reported their comparison with Xu et al.’s algorithm on the Adult dataset\textsuperscript{9}, comparison with Xu et al.’s algorithm serves as an indirect comparison with Ghinita et al.’s. As generalization hierarchies were used in Xu et al.’s algorithm, we used the same ones\textsuperscript{8}. Figure 2 shows the generalization hierarchy on the “workclass” attribute as an example.

The second algorithm is one of simplest; hence, we call it the simple algorithm in this paper. This algorithm suppress each tuple whose EC size is less than $k$. We chose this algorithm because it is one of the best-performing algorithms and has some resistance to minimality attacks (because it does not actively decrease information loss).

5.2 Information Loss Metrics

Because it is significant for data users to be able to find useful information, a desirable PPDP algorithm should minimize information loss. Hence, various information loss metrics have been proposed, some of which are for specific analyses\textsuperscript{11}. If PPDP executers know the purpose of the analysis in advance, the method in which they analyze the data as a proxy and publish the result is effective, and used widely. Therefore, if PPDP executers do not know the purpose of the analysis (e.g., if data users want to conceal the purpose), PPDP is necessary. Hence, the more general the purpose is, the more practical and desirable the model is.

We used the following two general-purpose metrics of information loss.

The first is NCP (Normalized Certainty Penalty)\textsuperscript{8} proposed by Xu et al. NCP is advantageous to Xu et al.’s algorithm because their algorithm is optimized in NCP. The NCP of a tuple $t$ is defined as follows:

$$NCP(t) := \frac{1}{|A|} \sum_{i=1}^{|A|} \left| n_{a_i} \right|$$

where $n_{a_i}$ is the node in the generalization hierarchy on attribute a corresponding to cell $t[a]$, $a_i$ is the root node in the generalization hierarchy on attribute $a$, and $|n|$ is the number of leaf nodes that are descendants of node $n$. To obtain a utility metric $U_N (0 \leq U_N \leq 1)$, we normalized this value as follows:

$$U_N := 1 - \frac{1}{|T|} \sum_{t \in |T|} NCP(t)$$

The second metric is information gain (Kullback-Leibler divergence), which is commonly used in the statistics community\textsuperscript{19} and is one of the most general metrics that react to changes in correlation among attributes, in contrast with NCP. The information gain $D_{KL}$ of $T$ from $T'$ is defined as follows:

$$D_{KL}(T, T') := \sum_i Pr[T] log\frac{Pr[T]}{Pr[T']}$$

where $Pr[T]$ is the probability distribution (of the values of leaf nodes in the generalization hierarchies) of $T$. To obtain a utility metric $U_K (0 \leq U_K \leq 1)$, we normalize as follows:

$$U_K := 1 - D_{KL}(T, T')/D_{KL}(T, T_0)$$

where $T_0$ is a table that consists of suppressed tuples.

5.3 Experimental Result

We used Java 8 in our implementation and experimental execution on a PC (Intel Core i7-2600 CPU @ 3.4 GHz, 16 GB RAM, 64-bit Windows 7 Professional SP1).

Figure 3 shows the results for the Adult dataset, and Fig. 4 shows the results for the pseudo data. “Utility (NCP)” values were calculated by Eq. (4) and “Utility (KLD)” values were calculated by Eq. (5). Note that “Row Count” and “Time” in Fig. 4 are logarithmic scales.

Figure 3 shows that our algorithm tends to be better in both utility and execution time than Xu et al.’s algorithm. Additionally, our algorithm tends to be better in utility than the simple algorithm. Figure 4 shows that our algorithm tends to be better in utility than Xu et al.’s algorithm as the tuple count increases. Figure 4 also shows that our algorithm is practically fast, even if the tuple count is large.

We found an example that may become a privacy issue in algorithms that do not satisfied $k$-presence-secrecy in the

\textsuperscript{7}http://archive.ics.uci.edu/ml/
Adult dataset. Table 4 shows both tuples that have values “1st–4th,” “Other-service,” and “Guatemala” in the Adult dataset.

Table 5 shows the tuples corresponding to Table 4 converted by Xu et al.’s algorithm with $k = 2$. Note that the value of the “sex” attribute in the Adult dataset is either “Female” or “Male.” Hence, if adversaries know the domain of the “sex” attribute and that an individual who has values “1st–4th,” “Other-service,” “Guatemala,” and “Male” is unique in the population, the adversaries can infer that he is included in the table, and his “INCOME” is “$\leq 50K$”. The reason that the adversaries can infer that he is included in the table is that the post-conversion table does not become Table 5 if he is not.

In contrast, this problem does not occur with our algorithm. For example, Table 6 shows both tuples that have the values “Local-gov,” “1st–4th,” and “Mexico” in the Adult dataset. Table 7 shows the tuples corresponding to Table 6, which has been converted by our algorithm with $k = 2, p = 1/8$. Even if adversaries know the domain of the “sex” attribute and that an individual who has values “Local-gov,” “1st–4th,” “Mexico,” and “Female” is unique in the population, the adversaries cannot infer whether she is included in the table because there is a possibility that

### Table 4: Example of tuples in the Adult dataset

| workclass | education | marital-status | occupation | race | sex | native-country | INCOME |
|-----------|-----------|---------------|------------|------|-----|----------------|--------|
| Private   | 1st–4th   |               | Other-service | White | Male | Guatemala       | $<\leq 50K$ |
| Private   | 1st–4th   | Never-married | Other-service | White | Female | Guatemala       | $<\leq 50K$ |

### Table 5: Tuples corresponding to Table 4 in the converted Adult dataset after Xu et al.'s algorithm is applied with $k = 2$

| workclass | education | marital-status | occupation | race | sex | native-country | INCOME |
|-----------|-----------|---------------|------------|------|-----|----------------|--------|
| Private   | 1st–4th   |               | Other-service | White | *   | Guatemala       | $<\leq 50K$ |
| Private   | 1st–4th   |               | Other-service | White | *   | Guatemala       | $<\leq 50K$ |

### Table 6: Another example of tuples in the Adult dataset

| workclass | education | marital-status | occupation | race | sex | native-country | INCOME |
|-----------|-----------|---------------|------------|------|-----|----------------|--------|
| Local-gov | 1st–4th   | Married-civ-spouse | Other-service | White | Male | Mexico         | $<\leq 50K$ |
| Local-gov | 1st–4th   | Married-civ-spouse | Other-service | White | Male | Mexico         | $<\leq 50K$ |

### Table 7: Tuples corresponding to Table 6 in the converted Adult dataset after our algorithm is applied with $k = 2, p = 1/8$

| workclass | education | marital-status | occupation | race | sex | native-country | INCOME |
|-----------|-----------|---------------|------------|------|-----|----------------|--------|
| Local-gov | 1st–4th   |               | Other-service | White | *   | Mexico         | $<\leq 50K$ |
| Local-gov | 1st–4th   |               | Other-service | White | *   | Mexico         | $<\leq 50K$ |
As described in Sect. 5.4, the state-of-the-art general k-anonymization algorithm is from Wong et al [10]. None of the existing generalization-based k-anonymization algorithms, including Wong et al.’s, satisfies k-presence-secrecy. Our algorithm is similar to Wong et al.’s in dividing tuples into clusters for each attribute. Our algorithm suppresses extra tuples probabilistically when clustering, to prevent against minimality attacks, and satisfies k-presence-secrecy.

Sampling is commonly used in real-world PPDP [19]. However, its effect on hiding presence is not clear. It normally removes or suppresses tuples at random. Hence, it hides presence and absence if we are lucky. However, it retains uncommon tuples if we are unlucky, and it unnecessarily removes or suppresses tuples that need not be concealed in many cases. In contrast, we have a clearer purpose, and avoid excess information loss.

Differential privacy is a privacy model that can protect privacy quantitatively [16]. For example, there are methods of adding Laplace noise [17] to achieve differential privacy (or P(k-anonymity) [18], which is the property of being similar to differential privacy) that satisfy k-presence-secrecy. However, because additive noise creates fictitious tuples, these methods cannot be used when data users require truthfulness [1] and make the evaluation of information loss difficult. Additionally, such methods tend to increase information loss because they also protect common tuples. There is an approach that achieves differential privacy without additive noise, but through the combination of sampling and k-anonymization [14]; however, its information loss is not compared with that of other methods. Our algorithm does not generate significant information loss, instead not protecting common tuples on principle. Clifton et al. [11] propose using k-anonymity and differential privacy properly, according to the user’s purpose.

Secret computation, such as SFE (Secure Function Evaluation) [20], is a method in which a data holder applies the process that a data user wants to apply to the data secretly. The merit of this approach is that it conceals the process from the data holder, unlike traditional statistical services. However, it is difficult to construct a mechanism with which raw data cannot be acquired, even if data users devise their processes. By contrast, PPDP has the advantage of clarity, because the information that be acquired by data holders is limited to the published plaintext.

7. Conclusion

k-anonymity is actively studied as a privacy model that should be achieved in PPDP. However, it does not guarantee the prevention of attribute linkage or protect against minimality attacks. We proposed a new model, called k-presence-secrecy, that prevents adversaries who have knowledge of a few uncommon individuals in a population and the algorithm from inferring whether an arbitrary individual is present in or absent from in a personal data table with high probability; further, we also proposed an algorithm that satisfies the model. k-presence-secrecy is more practical than δ-presence because an algorithm that satisfies
$k$-presence-secrecy needs only a PPDP target table as personal information. We showed that our algorithm, which satisfies $k$-presence-secrecy tends to be better in both utility and execution time than a typical $k$-anonymization algorithm on real personal data. In the future, we believe that the utility of this algorithm can be improved using generalization hierarchies.

References

[1] B.C.M. Fung, K. Wang, R. Chen, and P.S. Yu, “Privacy-preserving data publishing: A survey of recent developments,” ACM Comput. Surv., vol.42, pp.14:1–14:53, June 2010.

[2] L. Sweeney, “$k$-anonymity: a model for protecting privacy,” Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol.10, no.5, pp.557–570, Oct. 2002.

[3] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “$L$-diversity: Privacy beyond $k$-anonymity,” ACM Trans. Knowl. Discov. Data, vol.1, no.1, March 2007.

[4] P. Golle, “Revisiting the uniqueness of simple demographics in the us population,” Proceedings of the 5th ACM Workshop on Privacy in Electronic Society, WPES ’06, New York, NY, USA, pp.77–80, ACM, 2006.

[5] L. Sweeney, “Achieving $k$-anonymity privacy protection using generalization and suppression,” Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol.10, no.5, pp.571–588, Oct. 2002.

[6] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, “Incognito: efficient full-domain $k$-anonymity,” Proceedings of the 2005 ACM SIGMOD international conference on Management of data, SIGMOD ’05, New York, NY, USA, pp.49–60, ACM, 2005.

[7] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, “Mondrian multidimensional $k$-anonymity,” Proceedings of the 22nd International Conference on Data Engineering, ICDE ’06, Washington, DC, USA, p.25, IEEE Computer Society, 2006.

[8] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, “Utility-based anonymization using local recoding,” Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’06, New York, NY, USA, pp.785–790, ACM, 2006.

[9] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, “Fast data anonymization with low information loss,” Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB ’07, pp.758–769, VLDB Endowment, 2007.

[10] W.K. Wong, N. Mamoulis, and D.W.L. Cheung, “Non-homogeneous generalization in privacy preserving data publishing,” Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD ’10, New York, NY, USA, pp.747–758, ACM, 2010.

[11] C. Clifton and T. Tassa, “On syntactic anonymity and differential privacy,” Trans. Data Privacy, vol.6, no.2, pp.161–183, Aug. 2013.

[12] N. Li, T. Li, and S. Venkitasubramaniam, “$t$-closeness: Privacy beyond $k$-anonymity and $l$-diversity,” Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, pp.106–115, April 2007.

[13] R.C.W. Wong, A.W.C. Fu, K. Wang, and J. Pei, “Minimality attack in privacy preserving data publishing,” Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB ’07, pp.543–554, VLDB Endowment, 2007.

[14] N. Li, W. Qardaji, and D. Su, “On sampling, anonymization, and differential privacy or, $k$-anonymization meets differential privacy,” Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS ’12, New York, NY, USA, pp.32–33, ACM, 2012.

[15] M.E. Nergiz and C. Clifton, “$\delta$-presence without complete world knowledge,” Knowledge and Data Engineering, IEEE Transactions on, vol.22, no.6, pp.868–883, June 2010.

[16] C. Dwork, “Differential privacy,” Automata, Languages and Programming, vol.4052, pp.1–12, Springer, 2006.

[17] G. Cormode, C.M. Procopiuc, D. Srivastava, and T.T.L. Tran, “Differentially private publication of sparse data,” CoRR, vol. abs/1103.0825, 2011.

[18] M. Kakizawa, C. Watanabe, R. Furukawa, and T. Takahashi, “Improvement of pk-anonymization,” Proceedings of the 2014 IEEE 33rd International Symposium on Reliable Distributed Systems Workshops, SRDSW ’14, Washington, DC, USA, pp.82–87, IEEE Computer Society, 2014.

[19] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, “Privacy-preserving data publishing,” Found. Trends databases, vol.2, no.1–2, pp.1–167, Jan. 2009.

[20] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella, “Fairplay - a secure two-party computation system,” Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13, SSYM ’04, Berkeley, CA, USA, pp.20–20, USENIX Association, 2004.

Yuji Yamaoka received his M.E. degree from the University of Tokyo in 2003. He joined Fujitsu Laboratories Ltd. in 2003 and engaged in the research and development of information security technologies.

Kouichi Itoh received his B.E. degree from the Tokyo Institute of Technology in 1995, M.E. degree from the Japan Advanced Institute of Science and Technology in 1997, and PhD. degree from the Tokyo Institute of Technology in 2009. In 1997, he joined Fujitsu Laboratories Ltd., where he has been involved in the research and development of the implementation of public-key and common-key cryptosystems, side channel attacks, and data anonymization technology. He was awarded SCIS (the Symposium on Cryptography and Information Security) paper prize in 2002 and 2011, and CSS (the Computer Security Symposium) paper prize in 2002.