# Differential Private Data Collection and Analysis Based on Randomized Multiple Dummies for Untrusted Mobile Crowdsensing



doi: 10.1109/TIFS.2016.2632069

# Differential Private Data Collection and Analysis Based on Randomized Multiple Dummies for Untrusted Mobile Crowdsensing

Yuichi Sei and Akihiko Ohsuga, *Member, IEEE*

***Abstract*—Mobile crowdsensing, which collects environmental information from mobile phone users, is growing in popularity. These data can be used by companies for marketing surveys or decision making. However, collecting sensing data from other users may violate their privacy. Moreover, the data aggregator and/or the participants of crowdsensing may be untrusted entities. Recent studies have proposed randomized response schemes for anonymized data collection. This kind of data collection can analyze the sensing data of users statistically without precise information about other users' sensing results. However, traditional randomized response schemes and their extensions require a large number of samples to achieve proper estimation. In this paper, we propose a new anonymized data-collection scheme that can estimate data distributions more accurately. Using simulations with synthetic and real datasets, we prove that our proposed method can reduce the mean squared error and the JS divergence by more than 85% as compared to other existing studies.**

***Index Terms*—mobile crowdsensing, privacy, data mining.**

## I. INTRODUCTION

Owing to the development of ubiquitous computing and sensing technologies, numerous research methods for crowdsensing have been proposed to collect and analyze sensed environmental information from mobile phone users [1]. In the crowdsensing, individuals collectively share environmental data with a data aggregator, and the aggregator analyzes the collected data for decision making or marketing surveys. However, sensing aspects of a crowdsensing participant's surrounding environment, such as radiation level and location, may involve information that identifies an individual, and thus private information may be leaked.

A lot of studies have been proposed for privacy-preserving data aggregation. However, most of them require an a priori estimate of the fraction of malicious participants. In crowdsensing, it is difficult for each participant to know how many participants there are.

*Randomized response (RR)* [2] is a promising method for anonymized data collection. It can protect each participant's data even if the aggregator and $N-1$ of $N$ participants collude with each other. In RR, a sensed value is categorized as one of the predefined categories. That category is replaced by another category with certain probability, and then the disguised category is sent to the aggregator. Because the participant sends the true data to the server with probability $p$ and the disguised data to the server with probability $1-p$, the privacy of the participant is protected at a certain level.

All methods of existing RR schemes generate and report one disguised value from one sensed value. In contrast, we propose two methods: Single to Randomized Multiple Dummies (S2M) and S2M with Bayes (S2Mb), both of which generate a set of disguised values from one sensed value for anonymization. This concept is simple and novel. S2M and S2Mb consist of not only this anonymization algorithm, but also an algorithm that reconstructs the true data distribution—that is, an algorithm that generates an estimated contingency table (also called a cross tabulation or a multi-dimensional histogram) at the aggregator.

We use as the privacy metric $\epsilon$-differential privacy [3], which is one of the most promising privacy models in privacy-preserving data mining.

Further, we propose an algorithm that calculates optimized values of the parameters that constitute S2M and S2Mb. The optimized parameters satisfy $\epsilon$-differential privacy and minimize the expected values of Mean Squared Errors (MSE) and Jensen-Shannon (JS) divergence, which are the popular utility metrics.

In summary, our contributions are as follows: we propose S2M and S2Mb, both of which can make a better trade-off between privacy and utility; we propose an algorithm that calculates the expected MSE and optimized parameters of S2M and S2Mb; and we give the implementation results of synthetic and real datasets and prove that S2M and S2Mb outperform existing RR schemes. In the proposed method, the aggregator in crowdsensing systems can be used to estimate data distributions more accurately than other randomization methods. Moreover, the participants do not need to confirm the fraction of malicious participants.

The rest of this paper is organized as follows. Section II presents the application and attack models and defines privacy and utility as used in this study. Section III discusses the related methods. Section IV presents the design of our algorithm, and Section V presents the parameter optimization. Section VI presents the results of our simulations using synthetic and real datasets. Section VII discusses several design issues in our method. Section VIII concludes the paper.



Y. Sei and A. Ohsuga are with the University of Electro-Communications, Tokyo 182-8585, Japan (e-mail: seiuny@uec.ac.jp; ohsuga@uec.ac.jp).

This work was supported by JSPS KAKENHI Grant Numbers 26330081, 26870201, 16K12411 and the Telecommunications Advancement Foundation.

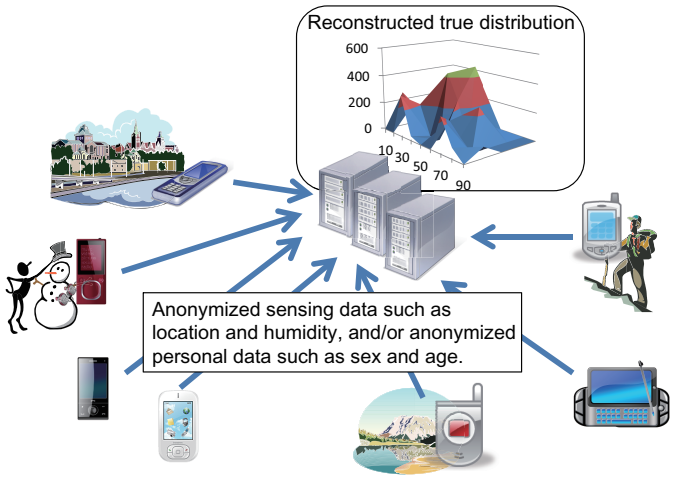



Fig. 1. Application model

## II. MODELS

### A. Application Model

Participants of crowdsensing perceive their surrounding environment through their mobile phones, and the mobile phones send the sensed data (e.g. radiation level, location) to the aggregator (Figure 1). We assume that the aggregator reconstructs the true data distribution, that is, it generates an estimated contingency table of the sensed data. For this reason, the aggregator requires categorical attribute values.

In regard to mobile crowdsensing applications, we can consider the noise, the name of the city that each participant resides in, and other factors of the participants' surrounding environment for urban planning [1], radiation levels [4], or the speed and type of cars, such as ambulance and taxi (in the anonymous monitoring of drivers). The data to be collected might also include personal data such as sex and age.

The process of the mobile crowdsensing application is as follows. First, the aggregator determines the *crowdsensing application ID* and the details of the attributes to be collected. We need the crowdsensing application ID because several crowdsensing applications may be executed at the same time and the aggregator should distinguish them. Then, the aggregator recruits participants. We assume that participants have an electronic device such as a smartphone and that they can decide whether or not they will participate in the crowdsensing application. If a participant agrees to participate in the crowdsensing application, the smartphone executes the proposed anonymization algorithm. Finally the aggregator analyzes the collected data by using proposed S2M or S2Mb.

### B. Attack Model

The attack model is the semi-honest adversary model. That is, the aggregator follows the proposed protocol but tries to analyze the individual information from each disguised data. Moreover, the aggregator can run an unlimited number of emulators that play smartphones. Those emulators can participate in the arbitrary crowdsensing applications. The aggregator can assign a certain crowdsensing application ID to one honest participant and the $N-1$ emulators it runs.

In this case, in the crowdsensing, there are one honest participant and $N-1$ emulators that are completely under the control of the aggregator. It is difficult for the honest participant to know how many honest participants are in the same crowdsensing application.

### C. Privacy Metric

Differential privacy [3] is one of most important privacy metrics, and it has been widely studied in data-mining research publications such as [5], [6]. Suppose that there is a data holder who is an honest entity and has a database of participants' true information, and a data analyst who may be a malicious entity and wants to use the database. When the data analyzer asks a query to the database, a randomized mechanism $\mathcal{A}$ adds noise to the query response. Intuitively, differential privacy is satisfied if the distribution of the output of the mechanism does not change observably when one participant's information in the database is changed.

Let $\epsilon$ be a positive real number. More specifically, the differential privacy is defined as follows:

**DEFINITION II.1** ($\epsilon$-differential privacy)**.** *Let $D$ and $D'$ be databases differing on at most one record. A randomized mechanism $\mathcal{A}$ satisfies $\epsilon$-differential privacy if and only if for all $Y \subset Range(\mathcal{A})$, the following equation holds:*

$$P(\mathcal{A}(D) \in Y) \leq e^{\epsilon} P(\mathcal{A}(D') \in Y) \quad \text{for all} \quad D, D'.$$

Kasiviswanathan et al. [7] show that this definition can be applied to anonymized data collection.

**DEFINITION II.2** (local privacy)**.** *Let $x$ and $x'$ be a database of size = 1, and let $\epsilon$ be a positive real number. A randomized mechanism $\mathcal{A}$ satisfies $\epsilon$-differential privacy if and only if for any output $y$, the following equation holds:*

$$P(\mathcal{A}(x) = y) \leq e^{\epsilon} P(\mathcal{A}(x') = y) \quad \text{for all} \quad x, x'. \qquad (1)$$

### D. Utility Metric

The aggregator can generate a contingency table (also called a cross tabulation or a multi-dimensional histogram) of data distribution for data analysis. We use as utility metrics MSE and JS divergence that measure the difference between the contingency table created from the original data (which is unknown to the aggregator) and the one estimated by the aggregator from the reported data.

Let $N$ denote the number of participants, and let $H_1, H_2, \ldots, H_F$ denote each category of sensed data. $F$ represents the number of data categories. Let $x_i$ denote the number of participants who sensed a value categorized to $H_i$, and let $\widehat{x_i}$ denote the number of participants categorized to $H_i$ in the reconstructed contingency table at the aggregator.

**DEFINITION II.3** (MSE)**.** *We use the MSE between $\widehat{x_i}$ and $x_i$ to quantify the utility for the reconstructed values:*

$$MSE = \frac{1}{F} \sum_{i=1}^{F} \left(\frac{x_i}{N} - \frac{\widehat{x_i}}{N}\right)^2. \qquad (2)$$

The Kullback-Leibler (KL) divergence and the JS divergence also have been used as utility metrics in literature. Because KL divergence cannot be used when the values of the distribution can be zero, we use JS divergence in this paper.

**DEFINITION II.4** (JS divergence)**.** *Let $\boldsymbol{P_X}$ and $\widehat{\boldsymbol{P_X}}$ represent two discrete probability distributions in which the $i$th value of them are $x_i/N$ and $\widehat{x_i}/N$, respectively.*

*We can use the JS divergence between $\widehat{x_i}$ and $x_i$ to quantify the utility for the reconstructed values:*

$$\begin{aligned} \text{JS divergence} &= \frac{1}{2}KL(\boldsymbol{P_X}\|\boldsymbol{R}) + \frac{1}{2}KL(\widehat{\boldsymbol{P_X}}\|\boldsymbol{R}), \\ \text{where} \quad \boldsymbol{R} &= \frac{1}{2}(\boldsymbol{P_X} + \widehat{\boldsymbol{P_X}}), \end{aligned} \tag{3}$$

where KL divergence is defined as follows:

**DEFINITION II.5** (KL divergence)**.** *Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ represent two discrete probability distributions, and let $P(i)$ and $Q(i)$ represent the $i$th value of $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively.*

*The KL divergence of $\boldsymbol{Q}$ from $\boldsymbol{P}$ is defined as*

$$KL(\boldsymbol{P}\|\boldsymbol{Q}) = \sum_i P(i)\log\frac{P(i)}{Q(i)}. \tag{4}$$

### *E. Problem*

Our aim is to anonymize each sensed value to make it satisfy $\epsilon$-differential privacy and to construct a contingency table at the aggregator while maintaining a high quality (that is, minimizing the MSE and/or the JS divergence). This challenge is defined as follows:

**PROBLEM II.6.** *Given a set of participants $U$, their sensed values $s_i$ $(i = 1, \ldots, |U|)$, and a privacy parameter $\epsilon$, find anonymized values $s_i^*$ satisfying $\epsilon$-differential privacy for all $i$. Moreover, given the anonymized values $s_i^*$, find reconstructed values $\widehat{x_i}$ $(i = 1, \ldots, F)$ that minimize the MSE and the JS divergence.*

## III. RELATED WORK

A lot of studies have been proposed for privacy-preserving data aggregation.

Studies of trusted aggregator schemes have been proposed widely [8], but they cannot be applied in our situation.

There are other techniques that preserve the privacy of aggregated data without a loss of utility based on encryption. Such techniques include [9], [10], [11]. These methods are categorized as encryption schemes. They assume that the aggregator is untrusted, but they also assume that the fraction of participants who collude with the aggregator is at most $\gamma$, which is a predefined parameter. If the aggregator in these methods colludes with more than $\gamma$% of participants, then the private information of honest participants may be disclosed. Note that the aggregator can easily generate emulators of smartphones who collude with it without limitation in mobile crowdsensing scenarios as described in II-A.

Recently, differential privacy algorithms that can be used for privacy-preserving data aggregation have been proposed [12], [13], [14]. These methods assume that the number of participants who collude with the aggregator is less than $\gamma$, which is the same predefined parameter used in the encryption schemes. When many participants collude with the aggregator, we cannot protect the honest participant's data. Huai et al. [15] have proposed a method that does not require the predefined parameter. According to their paper, if there are $N$ participants and $m$ of $N$ participants collude with the aggregator, the honest participant's data is disclosed with probability $m(m-1)/(N(N-1))$. If the aggregator generates many emulators of smartphones who collude with it, the probability that the honest participant's data is disclosed increases.

Randomized response (RR) schemes can protect each participant's data if the aggregator and $N-1$ of $N$ participants collude with each other. Let $H_1, \ldots, H_F$ denote each category of sensed data. $F$ represents the number of data categories. When a participant $u$'s actual category is $H_i$, we say that $H_i$ is $u$'s *true category*. When a participant $u$'s true category is $H_i$, the participant's mobile phone sends a category ID $i$ with probability $p$, and a category ID other than $H_i$ with probability $1-p$. We say that the category ID sent to the aggregator is a *disguised category*.

Let $p_{j,i}$ denote the probability by which a true category $H_i$ is disguised to a disguised category $H_j$. This is captured by the following probability matrix $M$:

$$M = \begin{pmatrix} p_{1,1} & \cdots & p_{1,F} \\ \vdots & \ddots & \vdots \\ p_{F,1} & \cdots & p_{F,F} \end{pmatrix}.$$

In most RR studies, all $p_{i,i}$ are set to the same $p$ value, and other elements $p_{j,i}$ $(j \neq i)$ are set to $(1-p)/(F-1)$. We refer to $p$ as the *probability of unchanging category*.

Let $x_i$ be the number of participants who sensed values categorized to $H_i$, let $y_i$ be the number of participants who reported disguised values categorized to $H_j$. and let $\widehat{x_i}$ be the *estimated* number of participants who sensed values categorized to $H_i$. We can estimate $\widehat{x_i}$ by

$$\begin{aligned} &\overrightarrow{\widehat{X}} = M^{-1}\overrightarrow{Y} \\ &\text{where } \overrightarrow{\widehat{X}} = (\widehat{x_1}, \ldots, \widehat{x_F})^\tau, \overrightarrow{Y} = (y_1, \ldots, y_F)^\tau, \end{aligned} \tag{5}$$

and $M^{-1}$ is the inverse matrix of $M$. The resulting $\overrightarrow{\widehat{X}}$ is an unbiased maximum likelihood estimation (MLE) of $\overrightarrow{X}$ [16].

This basic RR scheme is the same as FRAPP (DET), proposed by Agrawal et al. [2]. In the same paper, they also proposed another scheme, in which the value of the probability of unchanging category is determined randomly. This scheme is called FRAPP (RAN). Rizvi et al. proposed MASK [17] not for categorical attributes but for boolean attributes.

The above schemes use the MLE technique. Alternatively, the aggregator can reconstruct the distribution of categories based on the Bayes technique introduced in [18]. We call this scheme PRO. PRO can reduce the MSE, in some cases at the expense of the computational cost of the aggregator.

Groat et al. proposed another scheme, MDN, for multidimensional categories [19], [20]. Let $g$ denote the number of attributes to be collected. In MDN, we create each probability matrix $M_1, \ldots, M_g$, where probability of unchanging

TABLE I
SYMBOLS

| | |
|---|---|
| $N$ | Number of participants |
| $g$ | Number of attributes to be collected |
| $A_i$ | $i$th attribute to be collected ($i = 1, \ldots, g$) |
| $F_i$ | Domain size of attribute $A_i$ |
| $F$ | $\prod_{i=1}^{g} F_i$ |
| $H_{i,j}$ | $j$th category of attribute $A_i$ ($i = 1, \ldots, g$, $j = 1, \ldots, F_i$) |
| $H_i$ | $i$th element of the Cartesian product $(H_{1,1}, \ldots, H_{1,F_1}) \times \ldots \times (H_{g,1}, \ldots, H_{g,F_g})$ |
| $x_i$ | Number of participants who sensed a value categorized to $H_i$ ($i = 1, \ldots, \prod_{j=1}^{g} F_j$) |
| $\widehat{x_i}$ | Number of participants categorized to $H_i$ in the reconstructed contingency table at the aggregator |
| $s$ | Number of category IDs to be reported in the proposed method |
| $p$ | Probability that the original category ID is reported in the proposed method |
| $q$ | $(s-p)/(F-1)$ |
| $\epsilon$ | Privacy parameter of $\epsilon$-differential privacy |
| $w_i$ | Number of reported sets that contain category ID $i$ |

categories are all zeros for each attribute, and then calculate $M' = (((M_1 \otimes M_1) \otimes M_3) \ldots \otimes M_g)$ where $\otimes$ is the Kronecker product operator.

Erlingsson et al. [21] proposed RAPPOR. Kairouz et al. [22] analyzed the two algorithms; $k$-ary RR and RAPPOR, and proposed O-RR and O-RAPPOR, which extend $k$-ary RR and RAPPOR, respectively. Note that $k$-ary RR is the same as PRO in this paper. O-RAPPOR outperforms $k$-ary RR, RAPPOR, and O-RR in the usual value range of $\epsilon$ (i.e., from 0.1 to 1.0). O-RAPPOR uses Bloom filters [23] which form an efficient data structure and reduce the domain space. O-RAPPOR changes each Bloom filter value at random so that it can protect the participants' data.

There are several other schemes for RR. Chaytor et al. proposed Perturbation Partitioning (PP) [24] for a situation where a data holder has all original participant data. Evfimievski et al. [25] proposed RR for transactions (e.g., history of purchase). When the transaction size increases, their method can anonymize the transactions efficiently. However, they did not assume that one attribute of a participant has exactly one value. Moreover, they did not propose an algorithm that generates contingency tables from the anonymized transactions.

## IV. PROPOSED METHOD

Our main symbols are summarized in Table I.

### A. Analysis of Randomized Response

If each value $y_i$ ($i = 1, \ldots, F$) is exactly the same as the value represented by

$$y_j = \sum_{i=1}^{F} p_{j,i} x_i, \tag{6}$$

the MSE and the JS divergence are 0 (that is, the utility is maximized) because we use the maximum likelihood estimation. However, because the disguised category ID is determined in a probabilistic way, an actual value of $y_i$ is different from the "ideal" value represented by (6) in most cases.

For example, assume that there are 100 participants and that the number of categories $F$ is 11. Assume that the value of the probability of unchanging category is 0.5 and $p_{j,k}$ ($j \neq k$) of the probability matrix is $(1 - 0.5)/10$ = 0.05, and assume that all participants' true categories are $H_1$ for simplicity. In this case, we have $x_1$ = 100. The "ideal" value of $y_1$ is 100 $\times$ 0.5 = 50, and $y_i$ (s.t. $i \neq 1$) is 5. We can calculate $\widehat{x_i}$ from values of $y_i$ using (5). We get $\widehat{x_1}$ = 100, and $\widehat{x_i}$ (s.t. $i \neq 1$) is 0. The estimated values $\widehat{x_i}$ are exactly the same as the original values $x_i$; therefore, MSE and JS divergence are 0.

Unfortunately, there is not much likelihood of each $y_i$ becoming the ideal value. If $y_1$ = 40 and $y_i$ (s.t. $i \neq 1$) = 6, for example, we get $\widehat{x_1} \approx 77.8$ and $\widehat{x_i}$ (s.t. $i \neq 1$) $\approx 2.22$. The resulting MSE is 0.0049.

If we can ensure that the actual value $y_i$ is more ideal, we can reduce the MSE and the JS divergence. In existing RR schemes, each participant sends one randomized category ID to the aggregator. If we can increase the number of reported category IDs, the difference between the distribution of the actual reported category IDs and the "ideal" distribution of reported category IDs decreases, based on the *law of large numbers*. In contrast, if each participant sends more than one category ID, the privacy protection level falls to a low level. In this study, we analyze the tradeoff between utility and privacy, and then we introduce the optimal values of each that can make the best trade-off.

### B. The Anonymization Algorithm

We assume that each participant installs a smartphone application for anonymized data collection provided by the aggregator. If the participant agrees to join mobile crowdsensing, the application's anonymization algorithm of the application starts.

The application (i.e., the node) receives the details of the attributes to be collected, their category definitions, and parameters $s$ and $p$, which are necessary to execute the proposed anonymization algorithm, from the aggregator.

The node senses values such as location and radiation level from its environment, according to the specified attributes. When there are $g$ multiple attributes $A_1, A_2, \ldots, A_g$, we consider that these attributes are a single attribute with domain $A_1 \times A_2 \times \ldots \times A_g$. In detail, let $F_i$ denote the domain size of attribute $A_i$ and let $H_{i,j}$ denote the $j$th category of attribute $A_i$. Let $c_i$ represent the sensed category ID of attribute $A_i$. If a participant's sensed categories of attributes $A_1$, ..., $A_g$ are $H_{1,c_1}$, ..., $H_{g,c_g}$, the resulting true category is calculated by

$$\text{ID of true category} = \sum_{i=1}^{g} \left( c_i \prod_{k=i+1}^{g} F_k \right).$$

Let $F$ be $\prod_i F_i$, and let $K$ be a set of integers from 1 to $F$. Let $t$ be the ID of the true category of a participant.

First, the node $u$ creates an empty set $R_u$. Next, the node tosses a coin with head probability $p$. If the coin is head, the node adds $t$ and $s-1$ IDs randomly extracted from $K \setminus \{t\}$, to set $R_u$. If the coin is tail, the node adds $s$ IDs randomly extracted from $K \setminus \{t\}$ to $R_u$. Finally, the node $u$ sends the set $R_u$ to the aggregator.

We need parameters $s$ and $p$ to execute the node protocol. The method of determining their values based on the required privacy level $\epsilon$ is described in Section V. The proof that the anonymization algorithm satisfies $\epsilon$-differential privacy is also described in Section V.

**Example 1.** *Assume that the aggregator wants to analyze the radiation level in a certain location, and assume that the level has five categories. Assume that the parameters $s$ and $p$ are 3 and 0.4, respectively. Assume that the node's true category is $H_2$. The node tosses a coin with a head probability of 0.4. If the coin is the head, the node randomly extracts 2 numbers from $\{1, 3, 4, 5\}$ and creates a set containing the extracted numbers and 2. If the coin is tail, the node randomly extracts 3 numbers from $\{1, 3, 4, 5\}$ and creates a set containing the extracted numbers. Finally, the node sends the created set to the aggregator.*

Algorithm 1 shows the node protocol.

**Algorithm 1** Node protocol for a participant $u$

**Input:** $u$'s true category ID $t$, IDs of categories $K$, Parameters $s$ and $p$
**Output:** Set of anonymized values of a participant $u$

```
Creates empty set R_u
if rand() < p then
    R_u ⇐ {t} ∪ getRandElements(K \ {t}, s − 1)
else
    R_u ⇐ getRandElements(K \ {t}, s)
end if
```

The function $rand()$ returns a random value between 0.0 and 1.0, and the function $getRandElements(B, b)$ returns $b$ elements randomly from set $B$.

### C. The Reconstruction Algorithm

*1) S2M (Single to Randomized Multiple Dummies) :* Let $U$ denote the set of participants of anonymized data collection. First, the aggregator counts the reported sets $R_u$ $(u \in U)$ that contain each category ID $(i = 1, \ldots, F)$—that is,

$$w_i = \sum_{u \in U} H(R_u, i), \text{where} \quad H(R_u, i) = \begin{cases} 1 & (R_u \text{ contains category ID } i) \\ 0 & (otherwise). \end{cases} \tag{7}$$

If a participant $u$'s true category is $t$, $t$ is included in $R_u$ with probability $p$. In contrast, $f$ $(s.t., f \neq t)$ is included in $R_u$ with probability

$$q = p \cdot \frac{s-1}{F-1} + (1-p)\frac{s}{F-1} = \frac{s-p}{F-1}. \tag{8}$$

Let $x_i$ be the actual number of participants whose true categories are $H_i$, and let $\widehat{x_i}$ be the maximum likelihood estimate of $x_i$. Thus, we have

$$w_i = p\widehat{x_i} + q \sum_{j=1, j\neq i}^{F} \widehat{x_j} = p\widehat{x_i} + q(N - \widehat{x_i}).$$

From this, we have the following equation:

$$\widehat{x_i} = \frac{-qN + w_i}{p - q}. \tag{9}$$

We show this reconstruction protocol S2M of the aggregator in Algorithm 2.

**Algorithm 2** Aggregator protocol (S2M)

**Input:** Reported sets, category size $F$, parameters $p$ and $s$ (which are determined according to Section V)
**Output:** Distribution of true categories

```
q ⇐ (s − p)/(F − 1)
for i = 1, ..., F do
    w_i ⇐ result calculated by (7)
    x̂_i ⇐ (−qN + w_i)/(p − q)
end for
return x̂_i (i = 1, ..., F)
```

Note that if the number of attributes to be collected is more than one, the category $i$ in Algorithm 2 represents a combination of categories of the multiple attributes. The category $i$ in Algorithm 2 represents a combination of categories $H_{1,c_1}$, ..., $H_{g,c_g}$ where $c_i$ is calculated by

$$c_i = \left\lfloor \frac{i}{\prod_{k=i+1}^{g} F_k} \right\rfloor \pmod{F_i}. \tag{10}$$

*2) S2Mb (S2M with Bayes):* We extend the S2M algorithm by using Bayes' technique [26].

Let $\Omega$ be a discrete sample space. Let $A_i$ be an event, such that $A_i \subseteq \Omega$, and let $\{B_1, B_2, \ldots, B_F\}$ be a family of events, such that $\bigcup_j B_j = \Omega$ and exactly $s$ events in $\{B_1, B_2, \ldots B_F\}$ always occur at the same time.

In general, we have

$$Pr(A_i) = Pr(A_i \cap (\bigcup_{j=1}^{F} B_j)), \tag{11}$$

because $\bigcup_j B_j = \Omega$ and $A_i \subseteq \Omega$. According to the distributive law, we have

$$Pr(A_i \cap (\bigcup_{j=1}^{F} B_j)) = Pr(\bigcup_{j=1}^{F} (A_i \cap B_j)). \tag{12}$$

Given that exactly $s$ events in $\{A_i \cap B_1, A_i \cap B_2, \ldots, A_i \cap B_F\}$ always occur at the same time, we have

$$Pr(\bigcup_{j=1}^{F} (A_i \cap B_j)) = \sum_{j=1}^{F} Pr(A_i \cap B_j)/s = \sum_{j=1}^{F} Pr(B_j) Pr(A_i|B_j)/s. \tag{13}$$

Based on Equations 11-13, we have

$$Pr(A_i) = \sum_{j=1}^{F} Pr(B_j) Pr(A_i|B_j)/s. \tag{14}$$

Let the events of $A_i$ and $B_j$ represent the event where a user's true category ID is $i$ and the event where category ID

$j$ is included in the user's disguised categories, respectively. Then, we have $Pr(A_i) = \widehat{x_i}/N$, $Pr(B_j) = w_j/N$, and

$$Pr(B_j|A_i) = \begin{cases} p & (i=j) \\ q & (otherwise.) \end{cases} \tag{15}$$

Moreover, because $\sum_k \widehat{x_k} = sN$, we have

$$\begin{aligned} &\sum_{k=1}^{F} Pr(B_j|A_k)Pr(A_k) = \frac{1}{N}(p\widehat{x_j} + \sum_{k=1,k\neq j}^{F} q\widehat{x_k}) \\ &= \frac{1}{N}(p\widehat{x_j} + q(sN - \widehat{x_j})). \end{aligned} \tag{16}$$

We have

$$Pr(A_i|B_j) = \frac{Pr(B_j|A_i)P(A_i)}{\sum_{k=1}^{F} Pr(B_j|A_k)Pr(A_k)}, \tag{17}$$

so we get

$$Pr(A_i) = \sum_{j=1}^{F} w_j \cdot \frac{Pr(B_j|A_i)Pr(A_i)}{p\widehat{x_j} + q(sN - \widehat{x_j})} \tag{18}$$

from $Pr(B_j) = w_j/N$ and Equations 14, 16, and 17.

Let us define $L_i$ and $Z$ as

$$L_i = \frac{w_i}{p\widehat{x_i} + q(sN - \widehat{x_i})},\ Z = \sum_{j=1}^{F} L_j. \tag{19}$$

Given that $\sum_j L_j Pr(B_j|A_i) = pL_i + q(Z - L_i)$, we have

$$\widehat{x_i} = \widehat{x_i}(pL_i + q(Z - L_i))/s.$$

Note that the value of $Z$ is not a variable of $i$.

As a result, we get

$$\widehat{x_i}[\sharp + 1] \Leftarrow \widehat{x_i}[\sharp] \cdot (pL_i + q(Z - L_i)) \tag{20}$$

where an element of $\widehat{x_i}[\sharp]$ $(i = 1, \ldots, F)$ represents the iteration at step $\sharp$. We set an initial value $\widehat{x_i}[0]$ to $w_i$ for all $i$ and recalculate (20) $\kappa$ times. We do not determine the value of $\kappa$ in advance, but we repeat the calculation until the difference between the sum of $\widehat{x_i}[\sharp]$ and the sum of $\widehat{x_i}[\sharp+1]$ is very small. After repeating $\kappa$ times, we finally get

$$\widehat{x_i} = \widehat{x_i}[\kappa]/s. \tag{21}$$

We show this reconstruction protocol S2Mb of the aggregator in Algorithm 3.

**Algorithm 3** Aggregator protocol (S2Mb)

**Input:** Reported sets, category size $F$, parameters $p$ and $s$ (which are determined according to Section V)
**Output:** Distribution of true categories

$q \Leftarrow (s-p)/(F-1)$
**for** $i = 1, \ldots, F$ **do**
  $w_i \Leftarrow$ result calculated by (7)
  $\widehat{x_i} \Leftarrow w_i$
**end for**
**for** $j = 1, \ldots, \kappa$ **do**
  **for** $i = 1, \ldots, F$ **do**
    $L_i \Leftarrow w_i/(p\widehat{x_i} + q(sN - \widehat{x_i})$
  **end for**
  $Z \Leftarrow \sum_i L_i$
  **for** $i = 1, \ldots, F$ **do**
    $\widehat{x_i} \Leftarrow \widehat{x_i} \cdot (pL_i + q(Z - L_i))$
  **end for**
**end for**
**for** $i = 1, \ldots, F$ **do**
  $\widehat{x_i} \Leftarrow \widehat{x_i}/s$
**end for**
**return** $\widehat{x_i}$ $(i = 1, \ldots, F)$

As with S2M, if the number of attributes to be collected is more than one, the category $i$ in Algorithm 3 represents a combination of categories with multiple attributes. The category $i$ represents a combination of categories $H_{1,c_1}$, ..., $H_{g,c_g}$, where $c_i$ is calculated by (10).

## V. Parameter Determination

### A. Optimized parameters for MSE

We introduce a method to choose optimized parameters $s$ and $p$ that will minimize the expectation of MSE and satisfy the required privacy level $\epsilon$.

First, we describe the constraints of parameters $s$ and $p$ based on $\epsilon$. Then, we describe their optimal values in terms of minimizing the MSE.

*1) Constraints of p and s:* Suppose that participant $u$'s true category ID is $t$. Let $R_u$ be the set of category IDs to be sent to the aggregator from participant $u$.

The probability that $R_u$ contains $t$ and a specified $s-1$ elements is represented by

$$P_t = \frac{p}{{}_{F-1}C_{s-1}}. \tag{22}$$

In contrast, the probability that $R_u$ does not contain $t$ but contains a specified $s$ elements is represented by

$$P_f = \frac{1-p}{{}_{F-1}C_s}. \tag{23}$$

Therefore, based on (1), the constraints of values of $s$ and $p$ based on $\epsilon$ are represented by

$$e^{\epsilon} \geq \max\left(\frac{P_t}{P_f}, \frac{P_f}{P_t}\right) = \max\left(\frac{p(F-s)}{(1-p)s}, \frac{(1-p)s}{p(F-s)}\right). \tag{24}$$

*2) Expected MSE:* Let us introduce a variable $\widetilde{w_i}$ as follows:

$$\widetilde{w_i} = px_i + q(N - x_i).$$

If each value of $w_i$ is exactly the same as $\widetilde{w_i}$, the MSE is exactly 0 because $\widehat{x_i}$ is the maximum likelihood estimate of $x_i$.

However, $w_i$ and $\widetilde{w_i}$ are different in most cases. Let $d_i$ denote the difference between $w_i$ and $\widetilde{w_i}$. In this case, the reconstructed value $\widehat{x_i}$ is larger than the actual $x_i$ by $d_i/(p-q)$. Therefore, when each $w_i$ is different from $\widetilde{w_i}$ by $d_i$, the MSE is represented by

$$MSE = \frac{1}{FN^2}\sum_{i=1}^{F}(\frac{d_i}{p-q})^2. \tag{25}$$

Let $E[d_i^2]$ denote the expected value of $d_i^2$, and let $r_{i,j}$ denote the number of reported sets that contain $j$ and have original value $i$. Because the number of participants whose true categories are $H_i$ is $x_i$, the possible range of the value of $r_{i,j}$ is from 0 to $x_i$. It is most likely that the value of $r_{i,i}$ is $px_i$ and the value of $r_{i,j}$ (s.t., $j \neq i$) is $qx_i$. The expected squared error value of the difference between the most likely values and the possible values of $r_{i,j}$ for $j = 1, \ldots, F$ can be represented by

$$\begin{aligned}
&\sum_{i=1}^{F} E[d_i^2] = \sum_{i=1}^{F}\sum_{r_{i,1}=0}^{x_i}\sum_{r_{i,2}=0}^{x_i}\cdots\sum_{r_{i,F}=0}^{x_i} \\
&\left[ {}_{x_i}C_{r_{i,i}} \cdot p^{r_{i,i}} \cdot (1-p)^{x_i - r_{i,i}} \prod_{j=1, j\neq i}^{F} {}_{x_i}C_{r_{i,j}} \cdot q^{r_{i,j}} \cdot (1-q)^{x_i - r_{i,j}} \right. \\
&\left. \left\{ (r_{i,i} - px_i)^2 + \sum_{j=1, j\neq i}^{F} (r_{i,j} - qx_i)^2 \right\} \right] \\
&= \sum_{i=1}^{F} x_i(p - p^2 + (F-1)(q - q^2)).
\end{aligned} \tag{26}$$

From (25), (26), and $N = \sum_i x_i$, the expected MSE $E[MSE]$ is represented by

$$E[MSE] = \frac{(1-F)(p^2F + s - (F+2p)s + s^2)}{NF(s - pF)^2}. \tag{27}$$

*3) Optimized parameters:* We have the following theorem from (24) and (27).

**Theorem V.1.** *The combinations of $s$ and $p$ that satisfy the required privacy level $\epsilon$ and will minimize the expected MSE are represented by*

$$s = \max\left(\left\lfloor \frac{F}{1+e^\epsilon} \right\rfloor, 1\right), \quad p = \frac{e^\epsilon s}{F - s + e^\epsilon s}. \tag{28}$$

The proof of Theorem V.1 is described in Appendix A.

We have also the following theorem:

**Theorem V.2.** *The proposed anonymization algorithm where $s$ and $p$ are determined by (28) satisfies $\epsilon$-differential privacy.*

*Proof.* We prove that

$$\frac{P(\mathcal{A}(\text{a database with } H_\alpha) = y)}{P(\mathcal{A}(\text{a database with } H_\beta) = y)} \leq e^\epsilon, \tag{29}$$

for any $\alpha$ and $\beta$, if the parameters $s$ and $p$ are set based on (28) and the value of $\epsilon$ is greater than 0.

In the anonymization algorithm, a user generates a combination of $s$ elements from $F$ elements. The probability that a certain combination is generated is $P_t$ (represented by (22)) or $P_f$ (represented by (23)). Therefore, the possible values of the left side of (29) are $P_t/P_f$, $P_f/P_t$, and 1.

We have the following equation:

$$\begin{aligned}
\frac{P_t}{P_f} &= p \cdot \frac{(s-1)!(F-s)!}{(F-1)!} \cdot \frac{1}{1-p} \cdot \frac{(F-1)!}{s!(F-s-1)!} \\
&= \frac{e^\epsilon s}{F - s + e^\epsilon s} \cdot \left\{ 1/(1 - \frac{e^\epsilon s}{F - s + e^\epsilon s}) \right\} \cdot \frac{F-s}{s} = e^\epsilon.
\end{aligned} \tag{30}$$

Therefore, the possible values of the left side of (29) are $e^\epsilon$, $1/e^\epsilon$, and 1. These values are less than or equal to $e^\epsilon$ if the value of $\epsilon$ is greater than 0. □

### *B. Optimized parameters for JS divergence*

We have the following theorem:

**Theorem V.3.** *The optimized parameters for the MSE are also optimal for the JS divergence.*

The proof is described in Appendix B.

## VI. EVALUATION

We compare our proposed S2M and S2Mb with FRAPP (DET), FRAPP (RAN), PRO, MDN, and MASK . All experiments were conducted on an Intel Xeon CPU E5-2687W v2 workstation with 128 GB of RAM.

### *A. Evaluation Settings*

*1) Existing Studies:* In the papers that originally proposed them, these schemes do not use differential privacy as a privacy metric. Therefore, we first describe how we calculated the constraints to satisfy $\epsilon$-differential privacy for each method.

**FRAPP (DET):** This scheme uses a probability matrix in which probability of unchanging categories are $p$ and other elements are $(1-p)/(F-1)$. The following equation

$$\max\left(\frac{p}{(1-p)/(F-1)}, \frac{(1-p)/(F-1)}{p}\right) \leq e^\epsilon \tag{31}$$

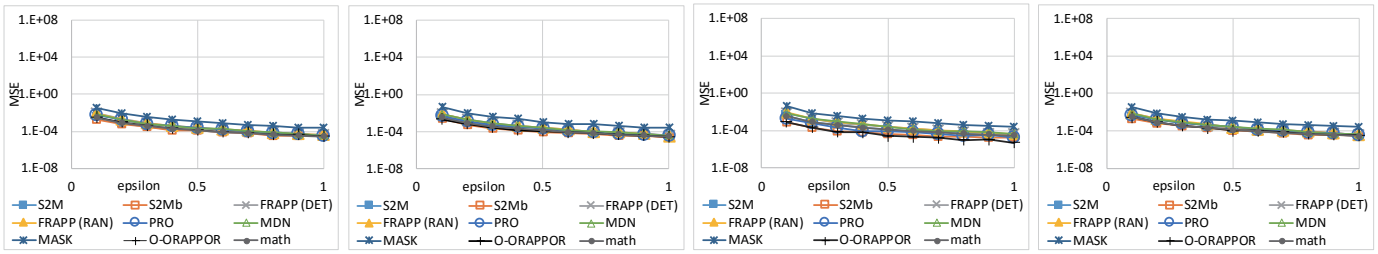
should hold to satisfy $\epsilon$-differential privacy. Therefore, we set

$$p = e^\epsilon/(F - 1 - e^\epsilon). \tag{32}$$

**FRAPP (RAN):** This scheme uses the probability matrix in which probability of unchanging categories are $p+r$ and other elements are $(1-p-r)/(F-1)$, where $r$ is a random variable uniformly distributed between $[-p/2, p/2]$. Let $t$ represent participant $u$'s true category. The probability that participant $u$ sends $t$ to the aggregator is $p$, and the probability that participant $u$ sends $f$ (s.t., $f \neq t$) to the aggregator is

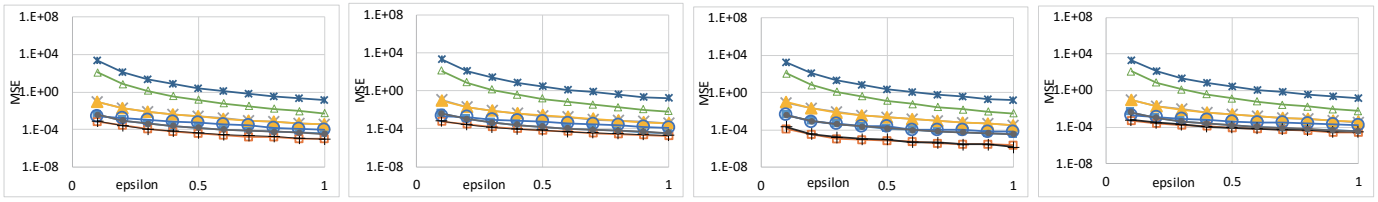
$$\int_{r=-p/2}^{p/2} \left(\frac{1-p-r}{F-1}\right)/p = (1-p)/(F-1).$$

Therefore, (31) should be satisfied for $\epsilon$-differential privacy. Therefore, we set $p$ using (32).



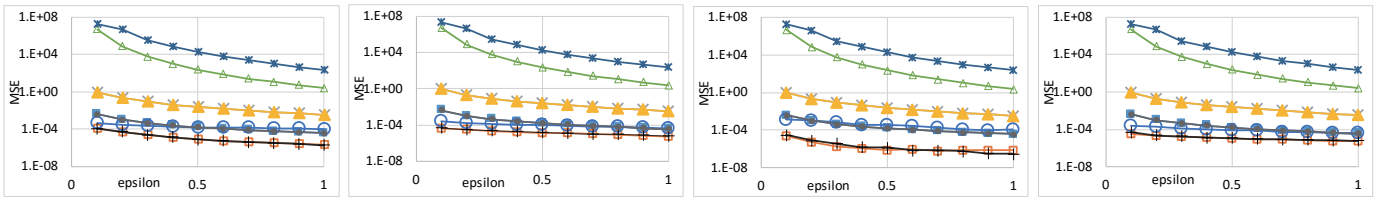

(a) Uniform (b) Normal (c) Peak (d) Random

$G = \{10\}$




(e) Uniform (f) Normal (g) Peak (h) Random

$G = \{10, 10\}$




(i) Uniform (j) Normal (k) Peak (l) Random

$G = \{10, 10, 10\}$

Fig. 2. Anonymization and reconstruction results of MSE of each data distribution

**PRO:** Because this scheme uses the same probability matrix as FRAPP (DET). Therefore, we set $p$ using (32).

**MDN:** This scheme uses the probability matrix calculated by $M' = (((M_1 \otimes M_1) \otimes M_3) \ldots \otimes M_g)$, where $g$ is the number of attributes to be collected and $M_i$ is a probability matrix of each attribute. In each $M_i$, the probability of unchanging categories of $M_i$ are set to $p_i$ and other elements are $(1 - p_i)/(F_i - 1)$, where $F_i$ denotes the number of categories of attribute $i$. The following equation

$$\max\left(\frac{\prod_{i=1}^{g} p_i}{\prod_{i=1}^{g}(1-p_i)/(F_i-1)}, \frac{\prod_{i=1}^{g}(1-p_i)/(F_i-1)}{\prod_{i=1}^{g} p_i}\right) \leq e^{\epsilon}$$

should hold to satisfy $\epsilon$-differential privacy. Therefore, we calculate each $p_i$ by the equation

$$p_i = e^{\epsilon/g}/(F_i - 1 - e^{\epsilon/g}). \tag{33}$$

**MASK:** The original scheme is used for Boolean databases. In the modified scheme, categorical attributes are converted into $M_b = \sum_i F_i$ Boolean attributes, where $F_i$ denotes the number of categories of attribute $i$. There are $M_b$ bits, and each bit flips with probability $1-p$ for anonymization. Because the original bits of each participant have exactly $g$ 1s, the equation

$$\max\left(\frac{p^{2g}}{(1-p)^{2g}}, \frac{(1-p)^{2g}}{p^{2g}}\right) \leq e^{\epsilon}$$

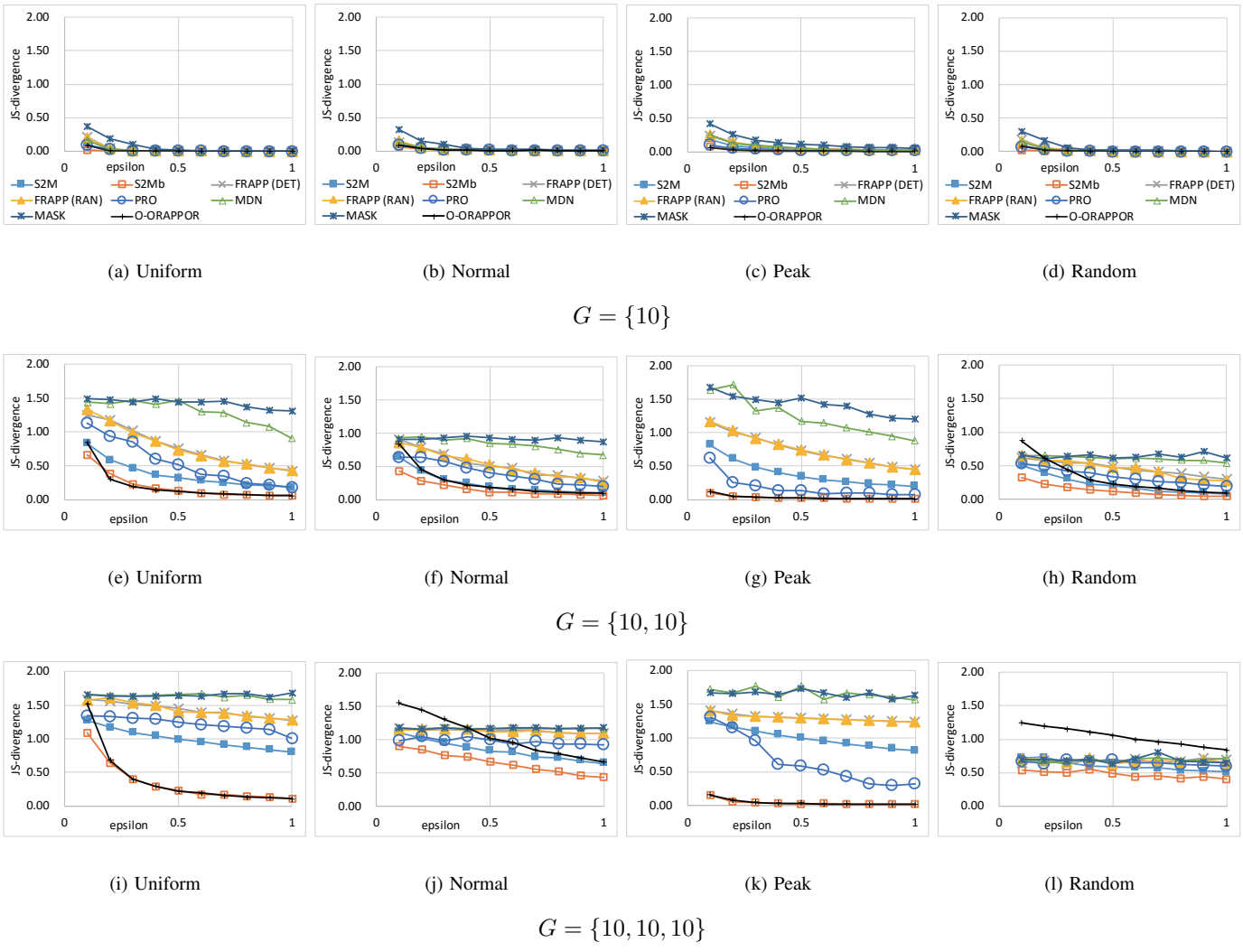
should hold to satisfy $\epsilon$-differential privacy. Therefore, we calculate $p$ by the following equation

$$p = 1 - 1/(1 + e^{\epsilon/(2g)}).$$

**O-RAPPOR:** O-RAPPOR originally uses $\epsilon$-differential privacy.

*2) Multiple attributes and value of $\epsilon$:* Let $G$ denote the set of each domain size of multiple attributes. For example, assume that radiation level and location are collected by each participant, and assume that the number of categories of radiation level is 3 (e.g., Low, Middle, and High) and the number of categories of location is 50 (e.g., each location represents one of the states of the United States). In this case, $G = \{3, 50\}$.

Many studies that use differential privacy set $\epsilon$ from 0.1 to 1. Therefore, in this paper, we set $\epsilon$ from 0.1 to 1.




(a) Uniform (b) Normal (c) Peak (d) Random

$G = \{10\}$

(e) Uniform (f) Normal (g) Peak (h) Random

$G = \{10, 10\}$

(i) Uniform (j) Normal (k) Peak (l) Random

$G = \{10, 10, 10\}$

Fig. 3. Anonymization and reconstruction results of JS divergence of each data distribution

### *B. Different typical distribution*

First, we evaluated the MSE and the JS divergence using synthetic datasets. To determine how the results were affected by different distributions of the data for true category, we conducted experiments using several distributions.

*1) Simulation setting:* We used four distributions: Normal, Uniform, Peak, and Random. In the Uniform, all $x_i$ were set to $N/F$. In the Normal, the values of $x_i$ followed the normal distribution. In the Peak, all participants had the same true category, i.e., $x_a = N$ for a certain $a$ and $x_i = 0$ (s.t. $i \neq a$). In the Random, each value of $x_i$ was determined randomly, satisfying $\sum_i x_i = N$.

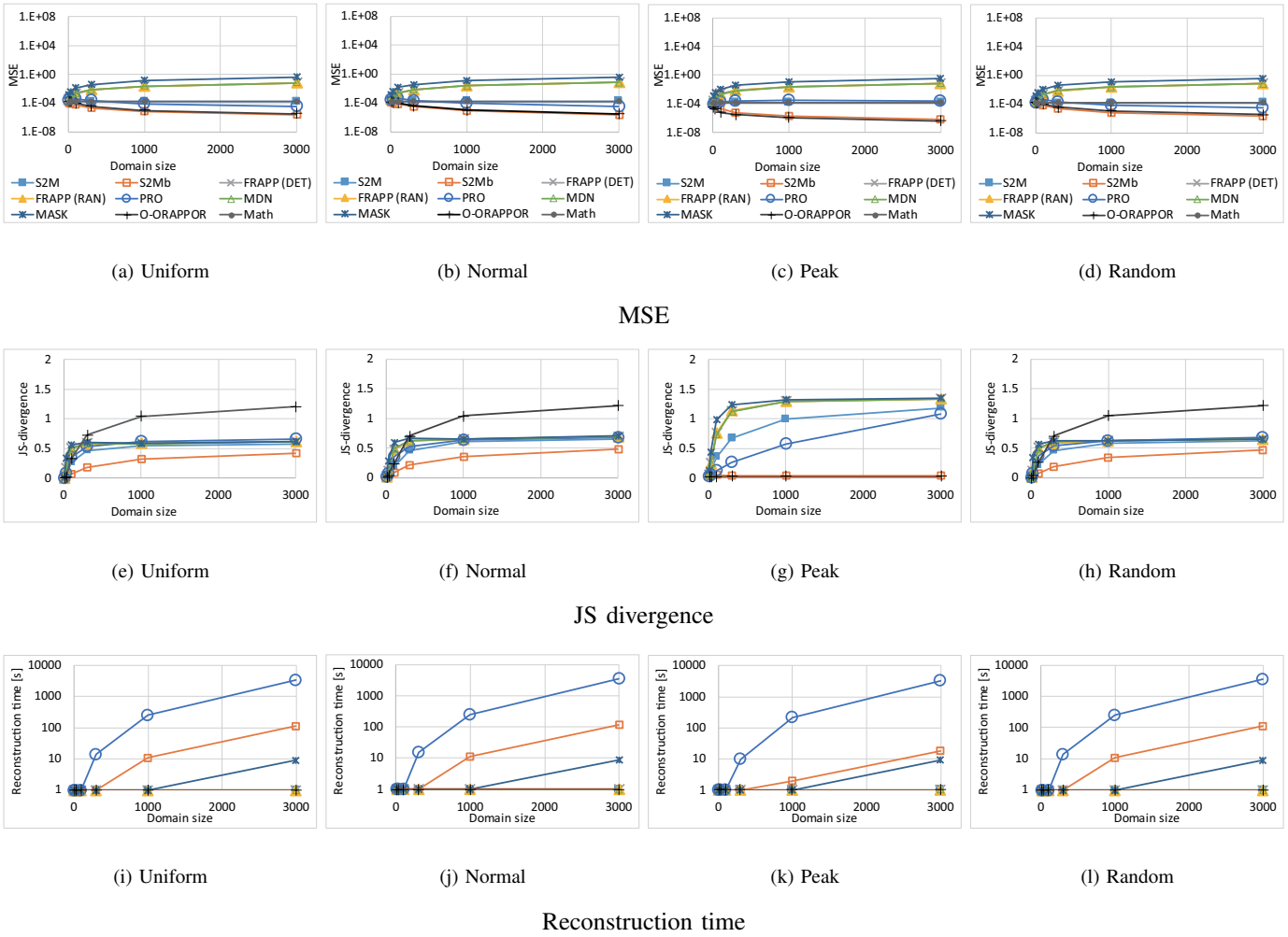
When we conducted experiments with multiple attributes, we let the distribution of each attribute value follow the specified distribution.

*2) Results of MSE:* In the first experiment, we set $N = 100,000$ and set $G = \{10\}$, $\{10, 10\}$, and $\{10, 10, 10\}$. The results are shown in Figure 2. Every scheme was executed 10 times, and its average MSEs are shown. The legend "Math" represents the expected MSE value of our methods calculated based on (27). We know from the figures that the expected MSE value is almost the same as that of S2M, and the MSE of S2Mb is always less than the expected MSE.

Figures 2(a)-2(d) represent the results of $G = \{10\}$. Although there was little difference between schemes, the MSEs of S2Mb and O-RAPPOR were the lowest among them. All MSEs decreased as $\epsilon$ increased. This is because a large $\epsilon$ means little privacy, and the probability that each participant sends her/his true category without being changed to the aggregator is getting larger.

Figures 2(e)-2(h) and Figures 2(i)-2(l) show the results of $G = \{10, 10\}$ and $G = \{10, 10, 10\}$, respectively. When $G$ was set to $\{10, 10\}$ or $\{10, 10, 10\}$, the performances of S2Mb and O-RAPPOR was visibly better than that of other schemes. We know from the data shown in Figure 2 that the larger the problem, the more the performances of S2Mb and O-RAPPOR exceeds those of other schemes.

The MSE calculates the average value of $(x_i/N - \widehat{x_i}/N)^2$. When the domain size is increasing, the value of $(x_i/N - \widehat{x_i}/N)^2$ tends to be decreasing because the value of $x_i$




(a) Uniform (b) Normal (c) Peak (d) Random

MSE

(e) Uniform (f) Normal (g) Peak (h) Random

JS divergence

(i) Uniform (j) Normal (k) Peak (l) Random

Reconstruction time

Fig. 4. Anonymization and reconstruction results with varying the domain size

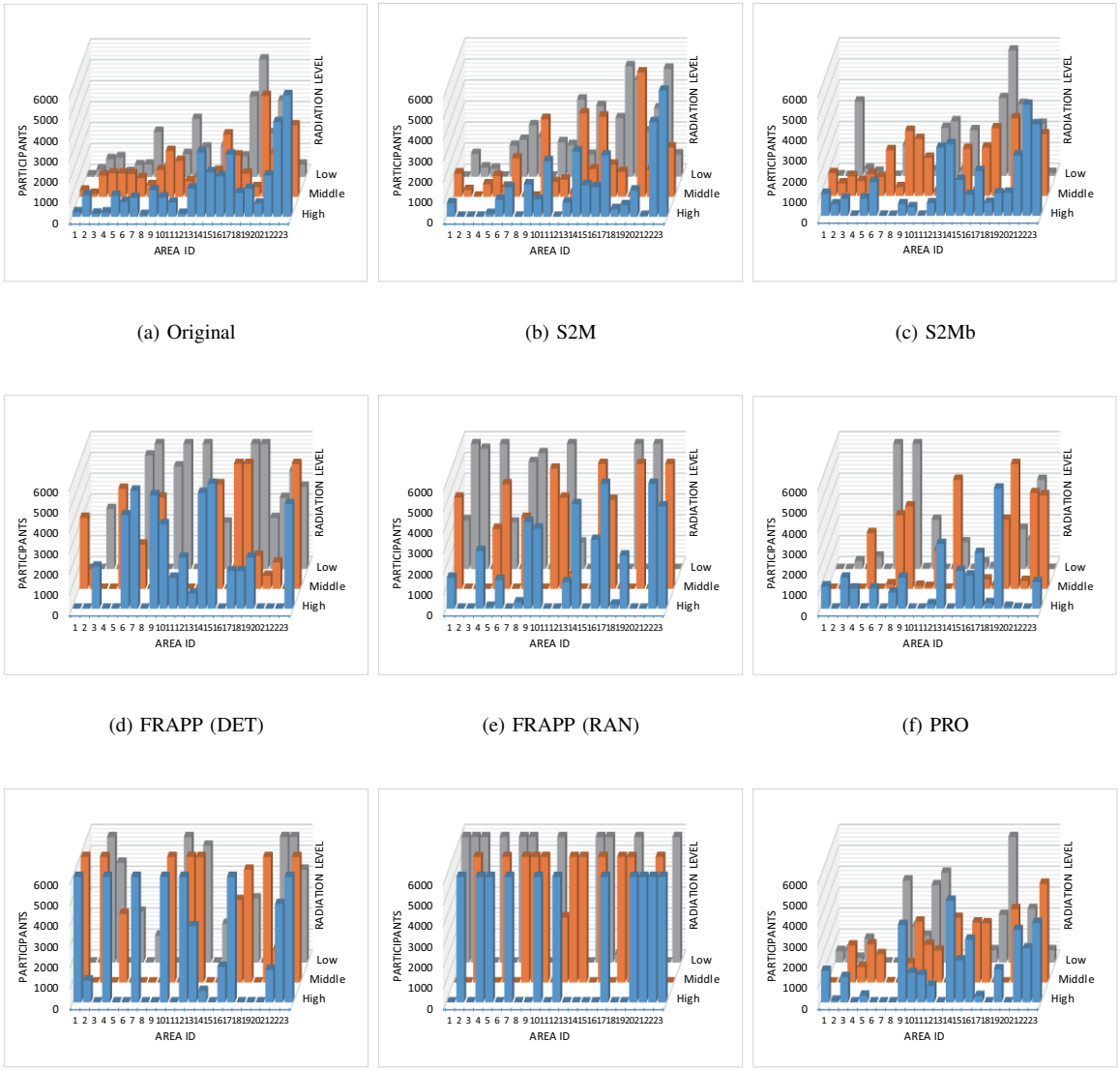
decreases. Therefore, the MSE decreases as the domain size increases in Figure 2.

*3) Results of JS divergence:* The results of the JS divergence are shown in Figure 3. The same as the results of the MSE, the results of S2Mb outperform the existing studies. O-RAPPOR uses the probability simplex algorithm [27] to make the estimated values positive because intermediate values can be negative. By using the probability simplex algorithm, the negative intermediate values become zeros. When many values of the original distribution are small positive values, many values of the estimated distribution of O-RAPPOR can be zeros. When we use MSE as a utility metric, the difference between small positive values and zeros is not as much of a problem. On the contrary, when we use JS divergence as a utility metric, the difference has a big impact on the JS divergence because MSE is low when the rough shapes of the two distributions are similar, whereas JS divergence is low when the detailed shapes of the two distributions are similar. Because S2Mb can maintain the small difference between the small positive values of the original distribution, S2Mb outperforms the existing methods.

The reason why the JS divergence of S2Mb and that of O-RAPPOR are almost the same in the peak distribution is that the many values of the peak distribution are zeros and the estimated values of O-RAPPOR are also zeros. On the contrary, because many values of other distributions are not zeros, the JS divergence of O-RAPPOR is large in other distributions.

Although we do not introduce the method of calculating the expected value of JS divergence, we know from the figure that the JS divergence tends to decrease as the MSE decreases. Unlike the results of the MSE, the results of the JS divergence of S2M are different according to underlying distributions. However, in all cases, the results of S2Mb realized the highest accuracy.

*4) Results of changing the domain size:* We then conducted an experiment in which we changed the domain size of an attribute from 10 to 3,000. We set $\epsilon$ to 0.5. The results are shown in Figure 4. When the domain size is increasing, the MSE tends to be decreasing. Therefore, the MSEs of S2M, S2Mb, and O-RAPPOR decrease as the domain size increases. On the contrary, the MSEs of FRAPP (DET), FRAPP (RAN), MDN, and MASK increase as the domain size increases because these methods are easily influenced by the domain




(a) Original (b) S2M (c) S2Mb

(d) FRAPP (DET) (e) FRAPP (RAN) (f) PRO

(g) MDN (h) MASK (i) O-ORAPPOR

Fig. 5. Results of radiation simulations

size.

With regard to JS divergence, the values of all methods increase as the domain size increases. Although the JS divergence of S2Mb and that of O-RAPPOR are almost the same in the peak distribution, S2Mb outperforms existing methods, including O-RAPPOR, in other distributions.

When the domain size was large, the reconstruction time for all methods increased. We then conducted an experiment in which we changed the number of participants $(N)$ from 1,000 to 1,000,000. We had similar results for different values of $N$.

We see that the reconstruction times of PRO and S2Mb largely depend on the domain size. With regards to S2Mb, the number of iterations was not determined in advance. The iteration shown in lines 6 to 14 in Algorithm 3 is terminated when the difference between the sum of $\widehat{x}_i[\sharp]$ and the sum of $\widehat{x}_i[\sharp + 1]$. The number of iterations seems to depend on the domain size. However, even if the domain size is 3,000 and the number of participants is 1,000,000, the calculation time is only 100 seconds.

### C. Smartphone radiation threat detection

Mobile crowdsensing, which is one kind of anonymized data collection, can help to detect and locate radiation threats in a city [20]. In this simulation, we assume that mobile phones are equipped with GPS and radiation monitoring systems.

*1) Simulation setting:* The main part of Tokyo, Japan, is administered in 23 special wards that have a total population of about 9,200,000. We assumed that 100,000 participants, about 1.1% of the population, would join the crowdsensing. Each participant's smartphone senses a radiation level that is categorized into one of three categories (High, Middle, and

Low) and senses its location, which is categorized into one of the 23 special wards. Then, each participant calculates her or his true category based on the category definitions, anonymizes the true category, and sends the disguised category to the aggregator. We set $\epsilon$ to 0.5. The ratio of the three radiation levels in each special ward is randomly determined in advance.

*2) Results:* The original data distribution is shown in Figure 5(a), and the reconstructed results of the server in each scheme are shown in Figures 5(b)-5(h).

As illustrated in Figure 5, MDN, MASK, FRAPP (DET), FRAPP (RAN), and PRO cannot accurately reconstruct the true data distributions. S2M and O-RAPPOR can reconstruct with higher precision, but several bins of the histograms differ greatly from the original bins. On the other hand, S2Mb can determine the true distribution almost perfectly, although one bin (Area ID is 3 and the radiation level is low) has an error. The values of MSE and the JS divergence of S2Mb are $8.32 \times 10^{-5}$ and 0.074, whereas those of O-RAPPOR are $1.25 \times 10^{-4}$ and 0.225. Given that many values of the estimated distribution of O-RAPPOR are zeros, the value of JS divergence of O-RAPPOR is large.

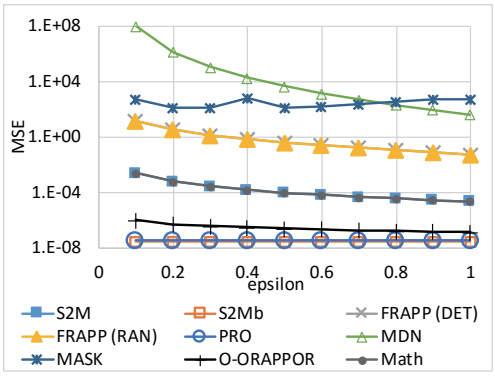
### D. A Real Public Dataset

We evaluated the MSE and the JS divergence by a real dataset, the Localization dataset [28].

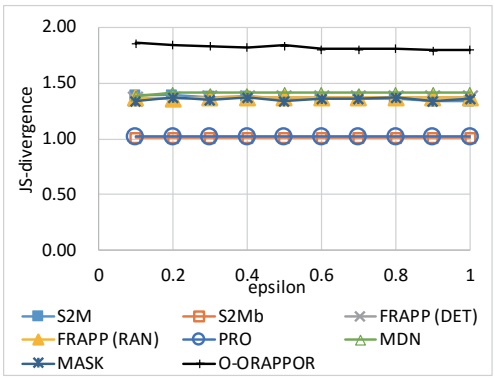
*1) Description of dataset:* The Localization dataset consists of 8 attributes and has 164,860 records. We extracted 3 attributes: x coordinate, y coordinate, and activity (e.g., walking, falling, sitting). The activity has 11 categories. Because x and y coordinates are numerical values, we divided each of them into 50 categories in advance. In the resulting database, $G = \{50, 50, 11\}$.

*2) Results:* Figure 6 shows the results. Every scheme is executed 10 times, and the average MSEs and the JS divergence.

The results are similar to the results of the combination of the three random distributions shown in Section VI.B because the ratios of 11 activities can be treated as a random distribution. The reason why the result of O-RAPPOR is worse with regard to JS divergence is that many of the values of the estimated distribution of O-RAPPOR are equal to zero. Based on the results, we can say that S2Mb can better maintain the small difference between the small positive values of the original distribution than other existing studies.




(a) MSE




(b) JS divergence

Fig. 6. Results of the Localization dataset

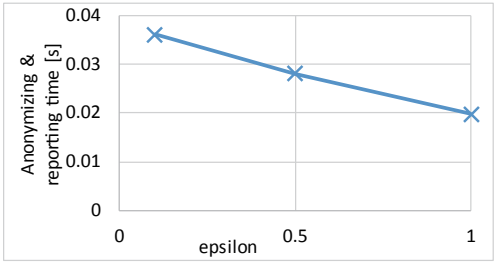



Fig. 7. Time required for smartphones.

### E. Real Smartphone Applications

We implemented our node protocol as a smartphone application for Android to verify the feasibility of the protocols. We measured the time it took for a smartphone to anonymize its sensed data and send the disguised data. Because our target is a crowdsensing system, the calculation cost of the randomisation algorithm conducted in smartphones should be light.

*1) Implementation and evaluation setting:* We implemented an Android smartphone application that has functions of sensing the surrounding noise level and location, calculating the category ID from the sensed data, anonymizing the category ID, and sending the disguised category ID to the aggregator. The noise level was categorized into one of 10 categories, and GPS information was represented as 10 categories of longitude and 10 categories of latitude.

We used three smartphones that installed the application, and the smartphones sensed the noise level and their location at the University of Electro-Communications campus in Tokyo. We measured the time it took for the smartphones to anonymize the sensed data and send the disguised data to the aggregator. We ignored the time it took to sense the noise level and location. The aggregator collected 1,000 data in total from the smartphones with each value of $\epsilon$.

*2) Results:* The results are shown in Figure 7. The smaller the $\epsilon$, the longer the time required for anonymization and

sending the data to the aggregator. This is because a smaller $\epsilon$ means a larger $s$—that is, when the value of $\epsilon$ is small, the node sends many category IDs to the aggregator.

However, it took less than 0.1 seconds in any of the cases. We can say based on Figure 7 that our node protocol is efficient for smartphones. Therefore, participants do not need to worry about the battery life of their smartphones.

## VII. DISCUSSION

### A. Different privacy levels among participants

We describe how to estimate the true distribution when the values of $\epsilon$ are different between participants.

We divide participants into groups based on the value of $\epsilon$, which is determined based on each participant's required privacy. Each group has the same value of the privacy parameter $\epsilon$. Let $\delta$ denote the number of groups, let $S_1, S_2, \ldots, S_\delta$ denote each group, and let $N_k$ denote the number of participants of a group $S_k$.

We use $x_{k,i}$ to represent the number of participants whose true category is $H_i$ in participant group $S_k$. Let $\widehat{x_{k,i}}$ denote the estimated number of participants in category $i$ in participant group $S_k$. Although details are omitted due to space limitation, we have

$$\widehat{x_i} = N \times \sum_{k=1}^{\delta} \left(\frac{1}{E[\sigma_k^2]} \times \frac{\widehat{x_{k,i}}}{N_k}\right) \sum_{k=1}^{\delta} \frac{1}{E[\sigma_k^2]}, \qquad (34)$$

where $E[\sigma_k^2]$ represents the estimated MSE calculated from (27) of participant group $S_k$.

### B. Sensing multiple times at a node

If node IDs can be hidden from the aggregator completely, nodes can report the anonymized sensed data with satisfying $\epsilon$-differential privacy at multiple times.

However, if this assumption is not realized, the resulting privacy level decreases when a node reports the anonymized sensed data at multiple times. To put it more specifically, when a node participates in crowdsensing that collects the same attributes at $h$ times with the privacy parameter being $\epsilon$, the node privacy is protected by $(h \times \epsilon)$-differential privacy. A larger value of the privacy parameter means a lower protection of privacy. Note that this characteristic is the same as all other existing RR schemes.

Therefore, if the node has a plan to participate in the anonymized data collection that collects the same attributes at $h$ times and the node wants to be protected by $\epsilon$-differential privacy, the node should execute the node protocol with $(\epsilon/h)$-differential privacy.

## VIII. CONCLUSION

RR can realize a privacy-preserving mobile crowdsensing where each participant's mobile phone probabilistically replaces the original category of the data with another category. The replaced category is sent to the aggregator, which attempts to estimate the distribution of the original categories of participants. However, RR schemes require great many samples in order to achieve proper reconstruction. In this paper, we propose S2M and S2Mb schemes, which can supersede existing RR schemes.

By simulations with synthetic and real datasets, we proved that S2Mb scheme can reduce the estimated errors. The larger the problem, the more the performance of S2Mb exceeds those of other schemes.

Future work will include the evaluation of other relevant datasets. We also plan to extend our approach to include the database of trajectories of participants' positions.

## REFERENCES


[1] P.-T. Chen, F. Chen, and Z. Qian, "Road Traffic Congestion Monitoring in Social Media with Hinge-Loss Markov Random Fields," in *Proc. IEEE ICDM*, 2014, pp. 80–89.

[2] S. Agrawal and J. Haritsa, "A Framework for High-Accuracy Privacy-Preserving Mining," in *Proc. IEEE ICDE*, 2005, pp. 193–204.

[3] C. Dwork, "Differential Privacy," in *Automata, Languages and Programming*, 2006, vol. 4052, pp. 1–12.

[4] E. Schubert, A. Zimek, and H.-P. Kriegel, "Generalized Outlier Detection with Flexible Kernel Density Estimates," in *Proc. SIAM SDM*, 2014, pp. 542–550.

[5] R. Bhaskar, S. Laxman, A. Smith, and A. Thakurta, "Discovering frequent patterns in sensitive data," in *Proc. ACM KDD*, 2010, pp. 503–512.

[6] R. Chen, B. C. Fung, B. C. Desai, and N. M. Sossou, "Differentially private transit data publication," in *Proc. ACM KDD*, 2012, pp. 213–221.

[7] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What Can We Learn Privately ?" *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2013.

[8] H. Wu, W. S. Ng, K.-L. Tan, W. Wu, S. Xiang, and M. Xue, "A privacy preserving framework for managing vehicle data in road pricing systems," in *Proc. ACM KDD*, 2013, pp. 1427–1435.

[9] Q. Li and G. Cao, "Efficient and privacy-preserving data aggregation in mobile sensing," in *Proc. IEEE ICNP*, 2012, pp. 1–10.

[10] R. Lu, X. Liang, X. Li, X. Lin, and X. S. Shen, "EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[11] S. Papadopoulos, A. Kiayias, and D. Papadias, "Exact In-Network Aggregation with Integrity and Confidentiality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1760–1773, 2012.

[12] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-Preserving Aggregation of Time-Series Data," in *Proc. NDSS*, 2011.

[13] Q. Li and G. Cao, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, 2013, pp. 60–81.

[14] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proc. ACM SIGMOD*, jun 2010, pp. 735–746.

[15] M. Huai, L. Huang, Y.-e. Sun, and W. Yang, "Efficient Privacy-Preserving Aggregation for Mobile Crowdsensing," in *Proc. IEEE BDCloud*, 2015, pp. 275–280.

[16] Z. Huang and W. Du, "OptRR: Optimizing Randomized Response Schemes for Privacy-Preserving Data Mining," in *Proc. IEEE ICDE*, 2008, pp. 705–714.

[17] S. J. Rizvi and J. R. Haritsa, "Maintaining data privacy in association rule mining," in *Proc. VLDB*, aug 2002, pp. 682–693.

[18] R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving OLAP," in *Proc. ACM SIGMOD*, 2005, pp. 251–262.

[19] M. M. Groat, B. Edwards, J. Horey, W. He, and S. Forrest, "Enhancing privacy in participatory sensing applications with multidimensional data," in *Proc. IEEE PerCom*, 2012, pp. 144–152.

[20] ——, "Application and analysis of multidimensional negative surveys in participatory sensing applications," *Pervasive and Mobile Computing*, vol. 9, no. 9, pp. 372–391, 2013.

[21] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response," in *Proc. ACM CCS*, 2014, pp. 1054–1067.

[22] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete Distribution Estimation under Local Privacy," in *Proc. ICML*, 2016.

[23] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.
[24] R. Chaytor and K. Wang, "Small domain randomization: same privacy, more utility," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 608–618, 2010.
[25] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *Proc. ACM PODS*, 2003, pp. 211–222.
[26] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," in *Proc. ACM SIGMOD*, 2000, pp. 439–450.
[27] W. Wang and M. A. Carreira-Perpiñán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv*, vol. 1309.1541, no. 1, pp. 1–5, 2013.
[28] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, and M. Gams, "An Agent-Based Approach to Care in Independent Living," in *International Joint Conference on Ambient Intelligence*, 2010, pp. 177–186.

## APPENDIX A
## PROOF OF THEOREM V.1

*Proof.* By differentiating (27) with respect to $p$, we get

$$\frac{\partial E[\sigma^2]}{\partial p} = \frac{2(F-1)^2(s-F)sN}{F(pF-s)^3}.$$

From this equation, we know that the expected MSE decreases with the decreasing $p$ when $p < s/F$ and that it decreases with the increasing $p$ when $s/F \leq p$.

On the other hand, from (24), the combination of $s$ and $p$ should satisfy the following equation:

$$\begin{cases} \dfrac{(1-p)s}{p(F-s)} \leq e^\epsilon & (p < s/F) \quad (35a) \\ \dfrac{p(F-s)}{(1-p)s} \leq e^\epsilon & (\text{otherwise}) \quad (35b) \end{cases}$$

The following will be studied in two cases: where $p < s/F$ and where $s/F \leq p$.

**Case 1:** $p < s/F$

The left-hand side of (35a) increases with the decreasing $p$ when $p < s/F$. Because the expected MSE decreases with the decreasing $p$ when $p < s/F$, we can minimize the expected MSE when $(1-p)s/(p(F-s))$ is equal to $e^\epsilon$ while satisfying the required privacy level. By solving the equation

$$\frac{(1-p)s}{p(F-s)} = e^\epsilon,$$

we get

$$p = \frac{s}{e^\epsilon F + s - e^\epsilon s}.$$

By substituting this equation into (27), we obtain the following:

$$E[\sigma^2] = \frac{N(F-1)}{(1-e^\epsilon)^2 F(F-s)s} \cdot (e^{2\epsilon}(F-s-1)(F-s) + 2e^\epsilon(F-s)s + (s-1)s).$$

By differentiating this equation with respect to $s$, we get

$$\frac{(F-1)^2(-e^{2\epsilon}(F-s)^2 + s^2)N}{(1-e^\epsilon)^2 F(F-s)^2 s^2}.$$

Therefore, when

$$s = \frac{e^\epsilon F}{1+e^\epsilon},$$

the expected MSE is minimized.

**Case 2:** $s/F \leq p$

The left-hand side of (35b) increases with the increasing $p$ when $s/F \leq p$. Because the expected MSE decreases with the increasing $p$ when $s/F \leq p$, we can minimize the expected MSE when $p(F-s)/((1-p)s)$ is equal to $e^\epsilon$ while satisfying the required privacy level. By solving the equation

$$\frac{p(F-s)}{(1-p)s} = e^\epsilon,$$

We get

$$p = \frac{e^\epsilon s}{F - s + e^\epsilon s}.$$

By substituting this equation into (27), we obtain the following:

$$E[\sigma^2] = \frac{N(F-1)}{(1-e^\epsilon)^2 F(F-s)s} \cdot (-F + F^2 + s - 2sF + 2e^\epsilon(F-s)s + e^{2\epsilon}(s-1)s + s^2).$$

By differentiating this equation with respect to $s$, we get

$$-\frac{(F-1)^2((F-s)^2 - e^{2\epsilon}s^2)N}{(1-e^\epsilon)^2 F(F-s)^2 s^2}.$$

Therefore, when

$$s = \frac{F}{1+e^\epsilon},$$

the expected MSE is minimized.

From the two cases above, we have

$$(s = \frac{F}{1+e^\epsilon},\ p = \frac{e^\epsilon s}{F - s + e^\epsilon s}), \quad \text{or} \quad (36)$$

$$(s = \frac{e^\epsilon F}{1+e^\epsilon},\ p = \frac{s}{e^\epsilon F + s - e^\epsilon s}), \quad (37)$$

for optimizing parameters.

We can use either (36) or (37), but in this paper, we adopt (36) because we consider that a smaller $s$ is preferable in terms of data transmission cost from the node to the aggregator.

Finally, because $s$ should be a natural number greater than or equal to 1, we have (28). □

## APPENDIX B
## PROOF OF THEOREM V.3

*Proof.* Similar to the analysis of the MSE, let us introduce $d_i$, which represents the difference between $w_i$ and $\widehat{w_i}$. Let $\zeta_i$ denote the difference between $\widehat{X_i}$ and $X_i$. The value of $\zeta_i$ is calculated by

$$\zeta_i = d_i/(p-q). \quad (38)$$

The JS divergence is calculated as follows. First, we get $\boldsymbol{R}(i)$ for each $i$.

$$\boldsymbol{R}(i) = \frac{1}{2}\left(\frac{X_i}{N} + \frac{X_i + \zeta_i}{N}\right). \quad (39)$$

Then we get

$$\text{JS divergence} = \frac{1}{2}KL(\boldsymbol{P_X}, \boldsymbol{R}) + \frac{1}{2}KL(\widehat{\boldsymbol{P_X}}, \boldsymbol{R})$$
$$= \sum_i \frac{X_i}{N}\log\frac{2X_i}{(2X_i+\zeta_i)} + \sum_i \frac{X_i+\zeta_i}{N}\log(1 + \frac{\zeta_i}{2X_i+\zeta_i}).$$

By differentiating this equation with respect to $\zeta_i$, we get

$$\log(1 + \frac{\zeta_i}{\zeta_i + 2X_i})/N. \quad (40)$$

From (40), we know that the JS divergence increases along with the increasing $\zeta_i$ when $\zeta_i \geq 0$, and the JS divergence decreases with the increasing $\zeta_i$ when $\zeta_i < 0$. That is, the JS divergence decreases with the decreasing $\zeta_i^2$.

Therefore, the parameters that minimize the expected value of $\sum_i \zeta_i^2$ can minimize the JS divergence. Because the optimized parameters for MSE minimizes $\sum_i \zeta_i^2$, these parameters can also minimize the JS divergence. □

**Yuichi Sei** received his Ph.D. degree in Information Science and Technology, from the University of Tokyo, Japan, in 2009. From 2009 to 2012 he was working at Mitsubishi Research Institute. Since 2013, he has been an assistant professor at the University of Electro-Communications. His research interests include pervasive computing, privacy-preserving data mining, and software engineering.

**Akihiko Ohsuga** (M'11) received a B.S. degree in mathematics from Sophia University in 1981 and a Ph.D. degree in computer science from Waseda University in 1995. He is a professor in the Graduate School of Informatics and Engineering, the University of Electro-Communications (UEC). He is also a visiting professor in National Institute of Informatics (NII). His research interests include agent technologies, web intelligence, and software engineering. He is a member of IEEE Computer Society (IEEE CS), Information Processing Society of Japan (IPSJ), Institute of Electronics, Information and Communication Engineers (IEICE), Japanese Society for Artificial Intelligence (JSAI), Japan Society for Software Science and Technology (JSSST), and Institute of Electrical Engineers of Japan (IEEJ). He was a chair of IEEE CS Japan Chapter. He was a member of the board of directors of JSAI and JSSST. He received the 1986 Paper Award from IPSJ.