# Privacy-Preserving Bin-Packing With Differential Privacy

Li, Tianyu ; Erkin, Zekeriya; Lagendijk, Reginald L.

**DOI**
10.1109/OJSP.2022.3153231

**Publication date**
2022

**Document Version**
Final published version

**Published in**
IEEE Open Journal of Signal Processing

**Citation (APA)**
Li, T., Erkin, Z., & Lagendijk, R. L. (2022). Privacy-Preserving Bin-Packing With Differential Privacy. *IEEE Open Journal of Signal Processing*, *3*, 94-106. Article 9721160. https://doi.org/10.1109/OJSP.2022.3153231

IEEE Open Journal of Signal Processing

Received 3 November 2021; revised 10 February 2022; accepted 11 February 2022. Date of publication 24 February 2022; date of current version 16 March 2022. The review of this article was arranged by Associate Editor Arsenia Chorti.

*Digital Object Identifier 10.1109/OJSP.2022.3153231*

# Privacy-Preserving Bin-Packing With Differential Privacy

**TIANYU LI (Graduate Student Member, IEEE), ZEKERIYA ERKIN (Senior Member, IEEE), AND REGINALD L. LAGENDIJK (Fellow, IEEE)**

Cyber Security Group, Delft University of Technology, 2628 XE Delft, The Netherlands

CORRESPONDING AUTHOR: TIANYU LI (e-mail: tianyu.li@tudelft.nl)

This work was supported by the Dutch Research Council (NWO) under Project 439.18.453B through the Research Programme of Duurzame Living Labs fase 2. This publication is part of the Project Spark! Living Lab.

**ABSTRACT** With the emerging of e-commerce, package theft is at a high level: It is reported that 1.7 million packages are stolen or lost every day in the U.S. in 2020, which costs $25 million every day for the lost packages and the service. Information leakage during transportation is an important reason for theft since thieves can identify which truck is the target that contains the valuable products. In this paper, we address the privacy and security issues in bin-packing, which is an algorithm used in delivery centers to determine which packages should be loaded together to a certain truck. Data such as the weight of the packages is needed when assigning items into trucks, which can be called bins. However, the information is sensitive and can be used to identify the contents in the package. To provide security and privacy during bin-packing, we propose two different privacy-preserving data publishing methods. Both approaches use differential privacy (DP) to hide the existence of any specific package to prevent it from being identified by malicious users. The first approach combines differential privacy with k-anonymity, and the other one applies clustering before differential privacy. Our extensive analyses and experimental results clearly show that our proposed approaches have better privacy guarantees, better efficiency, and better performance than the existing works that use either differential privacy or k-anonymity.

**INDEX TERMS** Bin-packing, data anonymization, differential privacy, k-anonymity.

## I. INTRODUCTION

Today, data plays an important role in our modern society. Many services such as transportation, supply chain logistics and healthcare are heavily dependent on data. On the one hand, more data improve the quality of services and even enable personalized ones. On the other hand, the collected data pose a serious threat in terms of privacy violations since the collected data are mostly privacy-sensitive or commercially valuable [1]. Considering container management systems for the transportation of goods, in the largest ports around the world, thousands of containers per day are being transported [2]. Trucks bring containers in and out, and while doing so, it is commercially important to use the container space as much as possible. To utilize the container space efficiently, different companies share the trucks to transport their products, and optimization algorithms are proposed to arrange the packages in containers [3], [4]. While doing so, it is also important to protect the commercially sensitive package data since such data can be obtained by malicious entities, resulting in the theft of certain products from the ports [5], [6]. As reported in a survey with 2000 respondents who have shopped online in the last 12 months [7], 43% of them experienced package stolen in 2020. Among them, 64% had more than one packages stolen. Also, it is mentioned that information leakage is an important reason for truck theft, and thieves know which truck is the target that contains the valuable products [5]. In some cases, only the targeted products are stolen [8].

There are different processes during the transportation of packages that may leak information. In this paper, we address the privacy and security issues in bin-packing. The information of packages is needed when assigning items into bins. However, the information is sensitive and can be used to identify the contents in the package. Thieves can infer an iPhone or

a MacBook in the package with a specific weight and volume since it always has the same weight and volume.

To protect data privacy and simultaneously use the optimization algorithm for better container management, the authors in [9] proposed a method to solve the bin-packing problem under privacy-preservation. In that work, $k$-anonymity, which is a well-known technique for data anonymization [10], is used to publish anonymous container data. The authors use two $k$-anonymous algorithms: k-Optimize [11] and Flash [12], to publish data in a privacy-preserving manner. For every record in the dataset, there are $k-1$ same other records in the same dataset so that the record is indistinguishable. The authors use stochastic programming and robust optimization to address the uncertainty introduced by the $k$-anonymous published data that are fed to the optimizer. The authors clearly point out the trade-off between privacy guarantees and accuracy. However, the work completes computation in the order of minutes to hours for 25 or 50 items, which is with low efficiency. Meanwhile, the work is sensitive to the homogeneity attack since attackers can know the sensitive information if all the $k$ tuples of quasi-identifiers share the same value in the sensitive attribute. Also, it is sensitive to the background knowledge attack since attackers can know the sensitive information based on some background knowledge. For example, there are $k$ same packages, but the attacker knows the destination of the targeted package, and only one out of the $k$ packages is heading to the targeted destination [13].

Besides approaches using $k$-anonymity, there are different privacy-preserving optimization methods, such as [14]–[16], in which only the optimization process is privacy-preserving. In these works, the optimizer knows the original information of packages and containers, which raises privacy risks in that the optimizer can be malicious by misusing the data or leaking information to other malicious users.

In this paper, we address the bin-packing problem as in [9]. We assume that data is firstly anonymized and then fed to the optimizer as also suggested in [9]. However, unlike that work that relies on $k$-anonymity, we are focusing on Differential Privacy (DP) [17], [18] for two reasons: 1) to provide better privacy protection and 2) to achieve better efficiency in terms of run-time such that our proposals can be considered feasible in practice. We propose two algorithms based on DP:

- *Differential privacy with k-anonymity:* We first generate a lattice including all the possible generalization results of the input dataset with a given hierarchy, and then use the exponential mechanism [19] to output a specific generalization according to the utility. This method adds noise to the mapping function, which involves sampling, suppression and generalization selection. This method can reach a low value of $\epsilon$ for differential privacy and show low uncertainty based on the preset generalization hierarchy. However, the sampling and suppression result in only a proportion of data being processed.
- *Differential privacy with clustering:* We first cluster the data based on the number of occurrences, and then add Laplacian noise [18] to each cluster. This method directly adds noise to the data, resulting in a shorter run-time but introducing more noise, which has an impact on the performance.

Our security analysis and experimental results clearly show that our proposed methods provide better privacy and security guarantees than the previous work by comparing the probability of identifying the targeted package. The experiments show that the run-time of our proposed methods is significantly low, 0.1 seconds for 50 packages, while the previous work [9] needs several minutes or hours for anonymization. Also, the proposed methods achieve a comparable packing performance to the previous work [9].

The rest of the paper is organized as follows. In Section II, we explain the preliminaries including differential privacy and $k$-anonymity. In Section III, we present related works about the existing privacy-preserving data publishing methods and optimization methods. Then Section IV shows our two differential privacy-based data publishing methods followed by the security analysis in Section V and experimental results and analysis in Section VI. Finally, we give the conclusion and discussions in VII.

## II. PRELIMINARIES

### A. DIFFERENTIAL PRIVACY

Two datasets $D$ and $D'$ are neighbouring datasets if they only differ in one or zero rows of data, and an algorithm $\mathcal{A}$ satisfies $\epsilon$-differential privacy ($\epsilon$-DP) if and only if for neighbouring datasets $D$, $D'$ and any set $O \subseteq \text{range}(\mathcal{A})$ [17], [18]:

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in O]. \tag{1}$$

However, the guarantee is so strong that it is very hard to be implemented, and it is excessive in many situations [20]. To make it more practical, parameter $\delta$ serves as a small error factor in the equation. $\mathcal{A}$ satisfies $(\epsilon, \delta)$-differential privacy if:

$$\Pr[\mathcal{A}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in O] + \delta. \tag{2}$$

Based on the definition, the Laplace Mechanism and the Exponential Mechanism are two widely used mechanisms that satisfy differential privacy.

#### 1) THE LAPLACE MECHANISM

It is the most general mechanism for differential privacy, and it adds Laplace noise [18]. To add the noise, the mechanism applied Laplace distribution which is centred at zero with a scale parameter $b$:

$$Lap(x \mid \mu = 0,\ b) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right). \tag{3}$$

We use $Lap(b)$ to denote density $Lap(x \mid \mu = 0,\ b)$. Then for the query $f : \mathcal{D}^N \to \mathbb{R}^k$, a randomized algorithm $\mathcal{A}$ satisfies $\epsilon$-differential privacy if $\epsilon > 0$, k is the dimension of the

dataset, and $y_i$ is the noise added to dimension $i$:

$$\mathcal{A}(D,\ f,\ \epsilon) = f(D) + (y_1\ \dot{s}\ y_k)$$
$$\text{and } y_i \sim Lap\left(\frac{\Delta f}{\epsilon}\right). \tag{4}$$

In (4), $\Delta f$ is the sensitivity for the query $f : \mathcal{D}^N \to \mathbb{R}^k$, and the $l_1$-*sensitivity* ($\Delta f$) is defined as:

$$\Delta f = \max_{X,X' \in \mathcal{D}^N : ||X - X'||_1 \leq 1} ||f(X) - f(X')||_1. \tag{5}$$

#### 2) THE EXPONENTIAL MECHANISM

The exponential mechanism [19] is a technique for designing algorithms with differential privacy. In the exponential mechanism, a utility function $u : \mathcal{D}^N \times \mathcal{R} \to \mathbb{R}$ is defined to access the utility of each element input $n \in \mathcal{R}$, where $\mathcal{D}$ is the domain and $\mathcal{R}$ is a range. Then a measure $\mu$ is used to assign a large probability of elements with a large utility.

With the utility function $u$, we calculate the *sensitivity* ($\Delta u$) of the utility function as:

$$\Delta u = \max_{n \in \mathcal{R}} \max_{X,X' \in \mathcal{D}^N : ||X - X'||_1 \leq 1} |u(X, n) - u(X,' n)|\ , \tag{6}$$

and the output probability of the exponential mechanism is defined as:

$$\Pr[\mathcal{A}^{\epsilon'}_{u,\Delta u}(X) = t \in \mathcal{R}] = \frac{\exp\left(\epsilon' \cdot u(X,t)\right) \cdot \mu(t)}{\int_{\mathcal{R}} \exp\left(\epsilon' \cdot u(X,n)\right) \cdot \mu(n) dn} \tag{7}$$

which satisfies $\epsilon$-DP (where $\epsilon = 2\epsilon' \Delta u$).

### B. BIN-PACKING PROBLEM

The *bin-packing* problem is an NP-hard optimization problem [21]. A real example is how to load packages into a minimum number of containers while avoiding overloading nor oversizing. The problem can be considered with different dimensions: *weight* and *volume* (*height*, *width* and *length*), which means that the problem can be with 1-D (*weight* or *volume*), 2-D (*weight* and *volume*) or 4-D (*weight*, *height*, *width* and *length*).

In this paper, we formulate the bin-packing problem as proposed in [22]. Considering 1-D bin-packing problem, for $n$ items (or packages), we load them into the minimum number of bins (or containers). $w_j$ is the weight of item $j \in N$, where $N = \{0, 1, 2, \ldots, n\}$, and all the bins have capacity $c$. We define the decision variables $y_i$ and $x_{i,j}$ as follows:

$$y_i = \begin{cases} 1 & \text{if bin } i \text{ is used,} \\ 0 & \text{if bin } i \text{ is not used,} \end{cases} \tag{8}$$

$$x_{i,j} = \begin{cases} 1 & \text{if item } j \text{ is loaded in bin } i, \\ 0 & \text{if item } j \text{ is not loaded in bin } i. \end{cases} \tag{9}$$

Given $y_i$ and $x_{i,j}$, as shown in (8) and (9), the formulation of the 1-D bin-packing problem is:

$$\min \sum_{i \in N} y_i \tag{10}$$


**FIGURE 1.** Framework overview.

$$\text{s.t.} \sum_{j \in N} w_j x_{i,j} \leq c y_j \quad \forall i \in N, \tag{11}$$

$$\sum_{i \in N} x_{i,j} = 1 \quad \forall i, j \in N. \tag{12}$$

In (10), the objective is to minimize the number of bins, and the two constraints ensure that every bin is not overloaded and one item can only be loaded into one bin.

### C. THE FRAMEWORK FOR BIN-PACKING

Fig. 1 shows the framework used in this paper. The framework was proposed in [9], including two modules: the data publishing module and the optimizer module. In the data publishing module, we apply anonymization methods to the private dataset and publish the differentially private (DP) dataset to the public. Then the optimizer module gets data from the public and applies optimization to solve the bin-packing problem using the anonymous data. The whole framework is privacy-preserving since the optimization is based on anonymous data.

## III. RELATED WORK

Data anonymization is a technique to achieve privacy protection in data mining. The idea is to analyze data without revealing users' sensitive information [23]. Among many approaches, data perturbation methods [24], [25] attracted significant attention in recent years. By applying data perturbation, a certain amount of noise is added to the raw dataset to achieve data anonymization. The noise decreases the utility of the dataset while preserving users' privacy by adding uncertainty to the dataset. Two widely used methods are $k$-anonymity [10] and differential privacy [17], [18], which are based on data generalization and adding random noise.

The concept of $k$-anonymity was introduced by Samarati and Sweeney in 1998 [10]. A dataset is $k$-anonymous if, for each individual in the dataset, there are at least $k-1$ other individuals which show the same value. There are a variety of $k$-anonymous algorithms for data anonymization. For example, Datafly [26] is a heuristic $k$-anonymous algorithm, which generalizes the quasi-identifiers showing the most distinct values. Mondrian [27] is another modern $k$-anonymous algorithm proposed by LeFevre *et al.* By using *kd-tree*, Mondrian splits the dataset and reconstruct it with equivalence classes whose size is at least $k$. Also, Emam *et al.* [28] proposed OLA, which achieves $k$-anonymity by using a pre-defined *generalization hierarchy* with generalization rules for each attribute.

In 2019, Hoogervorst *et al.* [9] applied $k$-anonymity to the bin-packing problem to publish the weights of packages. The

authors used full domain generalization and partition-based single-dimensional recoding to generalize the data. Also, two $k$-anonymous algorithms: k-Optimize [11] and Flash [12], are evaluated. However, $k$-anonymity is sensitive to the homogeneity attack and the background knowledge attack [13]. Meanwhile, $k$-anonymity brings uncertainty for the optimization, so the authors also applied stochastic programming and robust optimization to improve the performance of bin-packing. As far as we know, this is the only literature which applied anonymization techniques to the input data for bin-packing instead of proposing a privacy-preserving optimizer.

Different from $k$-anonymity, differential privacy aims to hide the existence of any single row of data in the dataset. Differential privacy can be applied to either add noise to the output of a certain query (such as the optimization in [14]) or add noise to the dataset [24], [29]–[32]. The work of [31] and [32] consider the trajectory data release using differential privacy. Hyukki Lee and Yon Dohn Chung [24] released the medical micro-data in a differentially private way. They applied generalization, suppression and insertion to add noise to the data. Moreover, they used the exponential mechanism to maximize the utility of the output dataset. The CASTLEGUARD [30] applied the Laplace mechanism to the numerical data to get a differentially private dataset, but the output is noisy and sparse with a low value of $\epsilon$. Also, Holohan *et al.* [29] applied $k$-anonymity to part of the attributes and differential privacy to the rest. Similar to the work of CASTLEGUARD, they also applied the Laplace mechanism to the numerical data. Besides, they gave a confidence interval for the perturbation. In our work, we used this method in Section IV-B.

Overall, from the literature, there are two main techniques for data anonymization: $k$-anonymity and differential privacy. However, $k$-anonymity based approaches are sensitive to background knowledge attacks and need a long run time (several minutes or hours) to find the optimal. Meanwhile, existing differential privacy based methods introduce large noise to the dataset for bin-packing problems, which can influence the performance. To achieve better efficiency, better privacy guarantees (compared to $k$-anonymity solutions) and better performance for bin-packing (compared to the existing differential privacy solutions), we propose two different approaches by (1) combining the use of $k$-anonymity and differential privacy and (2) applying clustering with differential privacy.

## IV. DATA ANONYMIZATION USING DIFFERENTIAL PRIVACY

In this section, we present two data anonymization methods based on differential privacy with different approaches and strengths. The first method combines differential privacy and $k$-anonymity using preset generalization hierarchy and the differentially private node selection method, which shows better privacy guarantee but lower efficiency. The second method adds Laplace noise to the data in each cluster, which works more efficiently since all items are considered each time, but it is with a lower privacy guarantee.

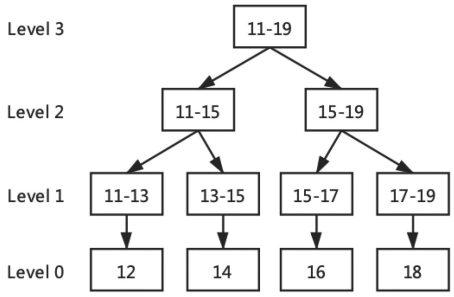


**FIGURE 2.** Example full-domain generalization.

### A. DIFFERENTIAL PRIVACY WITH K-ANONYMITY

This subsection shows a data anonymization method that combines differential privacy with $k$-anonymity. We firstly generalize data based on the full-domain generalization method and construct a differential private mapping function based on a $k$-anonymous algorithm OLA [28]. Then we prove that the new $k$-anonymization method satisfies differential privacy.

#### 1) FULL-DOMAIN GENERALIZATION

Full-domain generalization [33] is a widely used method for recoding [10]. For different quasi-identifier attributes $Q_i$, a generalization function $\phi_i$ is defined as $\phi_i : D_{Q_i} \rightarrow D_{G_i}$, and $D_{Q_i} \leq_D D_{G_i}$, which means that $D_{G_i}$ is generalized from $D_{Q_i}$. $D_{Q_i}$ is the original dataset and $D_{G_i}$ is the generalized dataset. For each value $q \in D_{Q_i}$, $\phi_i$ maps it to $g \in D_{G_i}$, and we can get that $g \in \gamma^+(q)$ (which means that $g$ is a generalization of $q$), or $g = q$. In a full-domain generalization, all values $q$ for all attributes $Q_i$ are replaced by $\phi_i(q)$.

In Fig. 2, we give an example of the possible generalization of four values {12, 14, 16, 18}. For the value **12**, we can generalize it into "11-13" or "11-15" or "11-19" or remain as "12". By generalization, we add some uncertainty to the data, which can decrease the utility but better protect privacy. The generalization is independent of data distribution, and instead, it is determined by the attribute. Also, with the generalization, the generalized value of different inputs may be the same, such as "12" and "14" may both output "11-15". The full-domain generalization method is used for $k$-anonymity since it reduces utility (with more generalization) to achieve $k$-anonymity. However, the generalization can only be used for 1-D bin-packing problems, since it is not possible to generalize a 2-D tuple in a same way. In this paper, we combined OLA and differential privacy to show a solution.

#### 2) LATTICE-BASED STRUCTURE

Firstly, we define different levels to show how much an attribute is generalized. As shown in Fig. 2, level zero means that no generalization is applied, and level three means that the data is fully generalized. Based on the definition, we use a lattice-based structure to decide how many generalizations should be applied when using the full-domain generalization. The structure is proposed in a $k$-anonymous algorithm

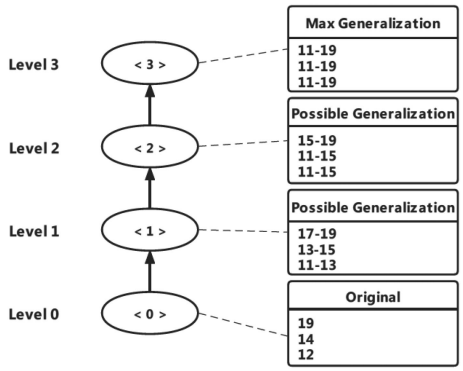


**FIGURE 3.** Example lattice with level 2 for one attribute.

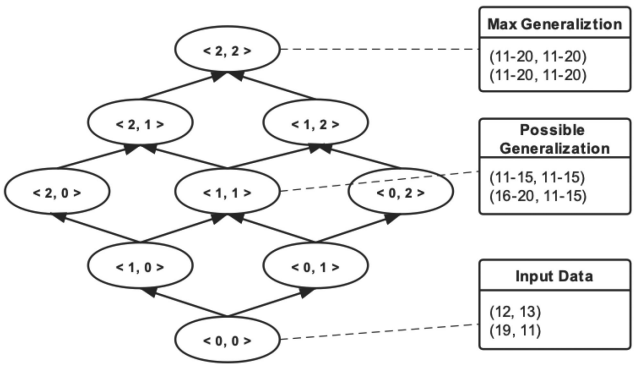


**FIGURE 4.** Example lattice with level 2 for two attributes.

OLA [28]. Fig. 3 gives an example when there is only one attribute, and $< 0 >, < 1 >, < 2 >$ are all the nodes in the lattice.

Then we expand it to be with two attributes as shown in Fig. 4. Each node indicates a different generalization of an attribute. The lattice becomes larger with a deeper full-domain generalization hierarchy or more attributes.

### 3) APPLICATION OF DIFFERENTIAL PRIVACY

Li *et al.* [20] give the idea of differential privacy under-sampling $(\beta, \epsilon, \delta)$-DPS where $\beta$ is the sampling factor, $\epsilon$ is the privacy budget, and $\delta$ is the small error factor for differential privacy. The sampling means that every record is only with probability $\beta$ being selected from the original dataset, otherwise it is removed. For an algorithm $\mathcal{A}$, if $\mathcal{A}^\beta$ is $\epsilon$-DP, $\mathcal{A}$ satisfies $(\beta, \epsilon, \delta)$-DPS. $\mathcal{A}^\beta$ means that the dataset is firstly sampled with probability $\beta$, and a smaller $\beta$ results in a smaller $\epsilon$. The same paper also proves that if the mapping function $\mathcal{A}_m$ of a $k$-anonymization algorithm satisfies $\epsilon_1$-DP, the $k$-anonymization algorithm satisfies $(\beta, \epsilon, \delta)$-DPS where

$$\epsilon \geq -\ln(1-\beta) + \epsilon_1, \tag{13}$$

$$\delta = d(k, \beta, \epsilon - \epsilon_1) = \max_{n:n \geq \lceil \frac{k}{\gamma} \rceil} \sum_{j > \gamma n}^{n} f(j; n, \beta),$$

**Algorithm 1:** Differential Privacy with $k$-Anonymity.

**Input:** Input dataset $D_{in}$, privacy budget $\epsilon_1$
**Output:** Differentially private dataset $D_{out}$

1: Apply the $\beta$ sampling to $D_{in}$, and get $D'_{in}$
2: Construct the lattice generalizations for attributes of $D'_{in}$
3: Calculate the utility of each node by (15)
4: Compute the probability for every node to be selected as the output (using the exponential mechanism with $\epsilon_1$)
5: Randomly pick a node $n_i$ according to the probability
6: Generalize the dataset $D_{out}$ for $n_i$
7: Suppress the records which do not satisfy $k$-anonymity
8: **return** $D_{out}$
9: *Note: sensitivity $\Delta u$ can be calculated anywhere and the algorithm satisfies $2\epsilon_1 \Delta u$-DP.*

$$\gamma = \frac{(e^{\epsilon - \epsilon_1} - 1 + \beta)}{e^{\epsilon - \epsilon_1}}, \tag{14}$$

in which $f(j; n, \beta)$ returns the probability of achieving $j$ successes in $n$ trials and the probability of a successful trial is $\beta$.

Based on the definition of $\epsilon$-DP $k$-anonymization algorithm, we present Algorithm 1. We firstly apply the sampling, which means that every record is with a probability $\beta$ being selected from the original dataset. In the second step, we generate the lattice based on the generalization hierarchies. Then in step 3 in Algorithm 1, we calculate the utility of each node using the utility function in (15) with consideration of privacy and information loss. Intuitively, we want the algorithm with a higher privacy guarantee and lower information loss. For the privacy part $sup(D, n)$, we consider $k$-anonymity in terms of the proportion of the suppressed data. For the information loss part $gen(D, n)$, we consider how many levels have been generalized.

$$u(D, n) = sup(D, n) \cdot gen(D, n) \tag{15}$$

where

$$sup(D, n) = \frac{|D_{k-anonymity}|}{|D_{raw}|} \in [0, 1] \tag{16}$$

$$gen(D, n) = 1 - \frac{1}{N_A} \sum_{i=1}^{N_A} \frac{n_{A_i}}{|FDG_{A_i}|} \in [0, 1]. \tag{17}$$

Equation (15) shows the trade-off between the information loss ($gen(D, n)$) and the privacy concern ($sup(D, n)$). Ideally, the output node is with the highest utility value. In (16), we choose the remaining proportion of the dataset to ensure that a higher value of $sup(D, n)$ represents better privacy guarantee. In (17), $N_A$ is the number of attributes, $n_{A_i}$ is the generalized level, and $|FDG_{A_i}|$ is the fully generalized level.

Based on the utility function and (7), we can calculate the output probability of the exponential mechanism in step 4 in Algorithm 1 as shown in (18). With the output probability for each node, a node is selected as the output node.

$$\Pr[\mathcal{A}^{\epsilon'}_{u,\Delta u}(D) = t \in \mathcal{R}] = \frac{\exp(\epsilon' \cdot u(D,t)) \cdot \mu(t)}{\int_{\mathcal{R}} \exp(\epsilon' \cdot u(D,n)) \cdot \mu(n) dn}. \tag{18}$$

Equation (18) satisfies $\epsilon$-differential privacy (where $\epsilon = 2\epsilon'\Delta u$), and the *sensitivity* ($\Delta u$) of the utility function is:

$$\Delta u = \max_{n \in \mathcal{R}} \max_{D, D' \in \mathcal{D}^N : ||D - D'||_1 \leq 1} |u(D,n) - u(D,'n)|. \tag{19}$$

The sensitivity shows the maximum change of the value of the utility function if we change only one row of data in the dataset. For the utility function in (15),

$$\begin{aligned} \Delta u &= \max_{n \in \mathcal{R}} \max_{D, D' \in \mathcal{D}^N : ||D - D'||_1 \leq 1} |u(D,n) - u(D,'n)| \\ &= \max_{n \in \mathcal{R}} \max_{D, D' \in \mathcal{D}^N : ||D - D'||_1 \leq 1} |sup(D,n) - sup(D,'n)| \\ &\quad \cdot gen(D,n) \\ &\leq |sup(D,n) - \left(sup(D,n) + \frac{k}{|D|}\right)| = \frac{k}{|D|}. \end{aligned} \tag{20}$$

With the equations, the mapping function satisfies $\epsilon_1$-DP with the exponential mechanism, so Algorithm 1 satisfies $(\beta, \epsilon, \delta)$-DPS as in (13).

#### 4) DISCUSSIONS

The proposed method can be expanded to be used for different data anonymization tasks with both categorical and numerical data. Also, the method can be applied to datasets with different dimensions. The proposed method can be used as a general scheme, but we only consider it for the bin-packing use case in this paper.

Meanwhile, the complexity of the approach is influenced by the number of attributes and records of a dataset. When the number of attributes increases, the lattice will increase exponentially, resulting in a long run time.

### B. DIFFERENTIAL PRIVACY WITH CLUSTERING

This subsection shows another anonymization method in which we adopt clustering before applying the Laplace mechanism, as shown in Algorithm 2.

Section III shows that we can add noise to the raw dataset to satisfy differential privacy, which is used in [29] and [30]. In the bin-packing problem, all the attributes are numerical, so we can add Laplace noise to the value of each attribute ($v_i$) as:

$$v'_i = v_i + Lap\left(\frac{\Delta v_i}{\epsilon}\right) \tag{21}$$

where $\Delta v_i = \max(v_i) - \min(v_i)$. Here the sensitivity is defined as the difference between the largest and lowest possible weight. If the weight is anonymous among this range, it is anonymous among all the packages. By adding Laplace noise, the output $v'_i$ satisfies $\epsilon$-DP. However, sometimes customers do not want to change the value of their products. For example, the *weight* is 10kg and the *volume* is 1m$^3$, and we publish it as 12kg and 0.8m$^3$. However, for express or logistics, the price is based on weight and volume. It can cause a problem if the differentially private value is not close to the accurate one. Considering this problem, the published dataset is only used to optimize the bin-packing problem, such as how to load packages into a minimum number of containers. Also, we introduce a confidence $c \in [0, 1]$, and Holohan *et al.* [29] show that the probability

$$\mathcal{P}(v_i \in [v'_i - r_c, v'_i + r_c]) = c \tag{22}$$

$$\text{where } r_c = -\frac{\Delta v_i}{\epsilon} \ln(1 - c). \tag{23}$$

By applying that, we can publish an interval instead of a single value. With the confidence $c$, we can control the probability of whether the accurate value is in the interval.

Equation (21) shows that the noise is influenced by outliers, such as the extremely large or heavy packages. In order to reduce the influence of outliers, we adopt clustering before applying differential privacy.

Here the clustering is based on the proportion of occurrence. For example, we can divide the input dataset into five parts by 5%, 30%, 30%, 30% and 5%. By applying the clustering, we can ease the problem of outliers, but it only satisfies differential privacy within each cluster. Most data are anonymous among the 30% records, which show similar weights or volumes.

To some extent, the clustering method extends the restriction of differential privacy. The proposed method anonymizes any single record among its cluster instead of the whole dataset. It is a trade-off between utility and privacy. There are thousands of packages in real use, and being anonymous among its cluster, which is with hundreds of packages, is still secure, as shown in Section V.

In Algorithm 2, the sensitivity is calculated for each cluster with complexity $O(n_c)$, and the noise is added to the weight of each package with complexity is $O(n_p)$, so the complexity for Algorithm 2 is $O(n_c) + O(n_p)$ where $n_c$ is the number of clusters and $n_p$ is the number of packages.

Also, the differential privacy with clustering method can be expanded to different data anonymization tasks, but it is restrictive since only numerical data with low dimensions can be considered. With high dimensions, there are a large number of clusters, and only a few records are in each cluster, which makes it infeasible. In this paper, we consider the bin-packing problem, which is a suitable use case for the approach.

## V. SECURITY ANALYSIS

This section analyzes and compares the privacy guarantees provided by the $k$-anonymity method in [9], the DP with $k$-anonymity method and the DP with clustering method in Section IV. As mentioned in Section III, the work of [9] is the only literature which considered privacy in bin-packing.

**Algorithm 2:** Differential Privacy with Clustering.

**Input:** Input dataset for de-identification $D_{in}$
**Output:** Output dataset $D_{out}$
1: Sort $D_{in}$
2: Apply clustering to $D_{in}$ based on the proportion of occurrence
3: Calculate $\Delta v_i$ for each cluster
4: Calculate $v_i'$ by adding Laplace noise to each cluster using (21)
5: Calculate the interval of each $v_i'$ using (22) and (23)
6: Get $D_{out}$ by combining the output of each cluster
7: **return** $D_{out}$

There are other works which applied differential privacy for anonymization such as [29], but the privacy guarantee is the same as our proposed approaches since differential privacy is applied to all of them. For that work, the performance for bin-packing is further compared in Section VI.

In this paper, we assume that the adversary knows the accurate information of one package, and he wants to identify this package from the anonymous output. If the adversary can identify the package, he knows which container the package is loading to, and thus he can track this package. To quantify how well privacy is protected concerning this scenario, we compare the probability that an adversary can identify the correct package from the output dataset. In the work of [9], only $k$-anonymity is considered. Each row of data occurs at least $k$ times in the output dataset. We can calculate the probability of identifying the same package from the output dataset given the information of the target package, as shown in (24). In the scenario, the adversary knows the original weight $a_i$ (such as $a_i = 12$), and he wants to identify which $b_i$ is its output. He firstly finds all possible $b_i$ which show the correct generalization for $a_i$ (such as [10,15]). Based on the definition of $k$-anonymity, there are at least $k$ possible $b_i$ showing the same generalization [10,15], so the probability is at most $1/k$.

$$\Pr[\text{identify correct } b_i \in D_{out} \text{ of } a_{target} \in D_{in}] \leq \frac{1}{k}. \quad (24)$$

In the differential privacy with $k$-anonymity method in Section IV-A, we add uncertainty to the dataset using sampling, generalization and suppression. Compared to the work of [9], this approach applies $\beta$ random sampling and differentially private mapping, which achieves $(\beta, \epsilon, \delta)$-DP. On the one hand, in the output dataset, every single row of data is hidden in a crowd. Based on the definition of differential privacy, the probability of outputting a specific record changes less than $e^{\epsilon}$ if we change any record in the input dataset. On the other hand, this approach applies $k$-anonymity with sampling and differentially private mapping. The $\beta$ sampling adds more uncertainty in that the adversary does not know whether the target package is in the input dataset or not. Even if the adversary gets all the possible $b_i$, he does not know whether the correct data is included. Equation (25) shows the new probability equation and $0 < \beta < 1$.

$$\Pr[\text{identify correct } b_i \in D_{out} \text{ of } a_{target} \in D_{in}] \leq \beta \frac{1}{k}. \quad (25)$$

Besides, the differentially private mapping function provides stronger privacy guarantees. In $k$-anonymous algorithms, the mapping is usually based on the existence of a few values [20]. For example, if the dataset is {1, 2, 3, 5, 7, 9} and $k = 3$, one of the possible generalizations is {[1, 3], [5, 9]}, which shows the existence of "1, 3, 5, 9" in the input dataset. The differentially private mapping does not overly depend on any single record in the input dataset. Each possible generalization can be chosen as the final output concerning their probability from the exponential mechanism [20]. As a result, the mapping function enhances the privacy guarantee, but it cannot be shown in (25).

In the differential privacy with clustering method in Section IV-B, we add Laplace noise to each cluster to hide the existence of any single row of data in each cluster. For example, if we have a dataset $D$: $\{a_0, a_1, \ldots, a_5\}$ with two clusters $C_1$: $\{a_0, a_1, a_2\}$ and $C_2$: $\{a_3, a_4, a_5\}$. The output dataset is $D'$:

$$\begin{aligned}\{&a_0 + \text{Lap}(\delta_1/\epsilon),\ a_1 + \text{Lap}\ (\delta_1/\epsilon),\ a_2 + \text{Lap}(\delta_1/\epsilon),\\ &a_3 + \text{Lap}(\delta_2/\epsilon),\ a_4 + \text{Lap}(\delta_2/\epsilon),\ a_5 + \text{Lap}(\delta_2/\epsilon)\}\end{aligned} \quad (26)$$

where the sensitivity

$$\delta_i = \max_{a_x, a_y \in C_i} |a_x - a_y|. \quad (27)$$

Assume that the adversary knows $x_{target} = x_2$ from $D$ and the output dataset $D'$. He wants to identify $x_2$ from $D'$, so he calculates the difference between the accurate data and the output data, getting:

$$\begin{aligned}\{&\Delta a_0 + \text{Lap}(\delta_1/\epsilon),\ \Delta a_1 + \text{Lap}(\delta_1/\epsilon),\ \Delta a_2 + \text{Lap}(\delta_1/\epsilon),\\ &\Delta a_3 + \text{Lap}(\delta_2/\epsilon),\ \Delta a_4 + \text{Lap}(\delta_2/\epsilon),\ \Delta a_5 + \text{Lap}(\delta_2/\epsilon)\}.\end{aligned} \quad (28)$$

where $\Delta a_i = a_{target} - a_i$.

If the adversary infers that the noise is generated by the Laplace mechanism, he knows the probability density function for Laplace distribution:

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right). \quad (29)$$

Based on the probability density function, the adversary can get the probability equation:

$$\begin{aligned}&\Pr[\text{identify correct } b_i \in D_{out} \text{ of } a_{target} \in D_{in}]\\ &= \frac{f(b_{target} - a_{target}\ |\mu = 0, b = \delta_1/\epsilon)}{\sum_{b_i \in D_{out}} f(b_i - a_{target}\ |\mu = 0, b = \delta_i/\epsilon)}\\ &= \frac{f(\text{Lap}(\delta_1/\epsilon)\ |\mu = 0, b = \delta_1/\epsilon)}{\sum_{b_i \in D_{out}} f(\Delta a_i + \text{Lap}(\delta_i/\epsilon)\ |\mu = 0, b = \delta_i/\epsilon)}.\end{aligned} \quad (30)$$

However, in (30), the adversary cannot get access to the value of $\delta$ and $\epsilon$, so he cannot get the result of the probability. Meanwhile, (30) shows that $\text{Lap}(\delta_i/\epsilon)$ influences the output probability. With a high sensitivity $\delta_i$ or a low $\epsilon$, the variance of the Laplace noise is large. The result of the $\text{Lap}(\delta_i/\epsilon)$

**TABLE 1 Experimental Settings for DP With k-Anonymity**

| setting | $c$ | $l$ | $u$ | $n$ | distribution |
|---|---|---|---|---|---|
| I. 50/L/U | 500 | 0.25 | 0.75 | 50 | 100%: U(125,375) |
| II. 80/L/U | 500 | 0.25 | 0.75 | 80 | 100%: U(125,375) |
| III. 50/S/U | 2500 | 0.05 | 0.15 | 50 | 100%: U(125,375) |
| IV. 80/S/U | 2500 | 0.05 | 0.15 | 80 | 100%: U(125,375) |
| V. 50/L/N | 500 | 0 | 1.00 | 50 | 100%: N(250,100) |
| VI. 50/L/Un | 500 | 0.25 | 0.75 | 50 | 75%: U(125,250), 25%: U(250,375) |
| VII. 200/L | 100000 | 0.2 | 0.35 | 200 | As used in [37] |

**TABLE 2 Experimental Settings for DP With Clustering**

| setting | $c$ | $l$ | $u$ | $n$ | distribution |
|---|---|---|---|---|---|
| I. 25/L/U | 500 | 0.25 | 0.75 | 25 | 100%: U(125,375) |
| II. 50/L/U | 500 | 0.25 | 0.75 | 50 | 100%: U(125,375) |
| III. 25/S/U | 2500 | 0.05 | 0.15 | 25 | 100%: U(125,375) |
| IV. 50/S/U | 2500 | 0.05 | 0.15 | 50 | 100%: U(125,375) |
| V. 25/L/N | 500 | 0 | 1.00 | 25 | 100%: N(250,100) |
| VI. 25/L/Un | 500 | 0.25 | 0.75 | 25 | 75%: U(125,250), 25%: U(250,375) |
| VII. 200/L | 100000 | 0.2 | 0.35 | 200 | As used in [37] |

counts equally or more than $\Delta a_i$, which can hide any record in the cluster.

In conclusion, both our proposed methods show better privacy guarantees, which can lower the probability that a potential attacker identifies targeted packages from the group.

## VI. EXPERIMENTAL EVALUATION

This section shows the experimental evaluation of the proposed methods in Section IV. We use Python to implement both methods on a laptop with Windows 10 Pro, Intel Core i7-10710 U CPU and 16.0 GB RAM. We use Google Or-Tools [34] for optimization. We have compared the performance of our proposed methods to the existing methods using $k$-anonymity [9] or differential privacy [29] with seven different synthetic datasets. BPPLIB [35] has given different benchmarks for bin-packing, such as Falkenauer [36], Scholl [37], and the Randomly Generated Instances [38]. Among them, the datasets are generated following the uniform distribution with a different number of items (n), capacity (c), minimum (l) and maximum (u) values. These datasets have a variety of combinations of these four factors (n, c, l, u) to test the performance of the optimization algorithms for bin-packing. However, this paper focuses on evaluating the proposed anonymization algorithms in terms of the performance for bin-packing, feasibility and run-time, instead of assessing the optimization methods. In the experiments, we consider more distributions such as normal distributions and uniform distributions, but fewer combinations of the four factors. To properly evaluate both proposed approaches, different instance settings are applied, and the settings are further introduced in Tables 1 and 2.

This section first shows the optimization methods for bin-packing and introduces the factors to evaluate the performance. After that, we demonstrate the performance of the proposed methods, in which the instance setting and performance analysis are included. Finally, we compare the performance of our proposed approaches to the existing works.

### A. OPTIMIZATION METHODS

Equation (10) shows how the standard optimization works, and the optimization result is the number of bins needed to load all the items. Note that the bin-packing problem is computationally NP-hard. The optimization method is how the problem is solved, so the optimization methods influence the global performance in terms of run time and whether the optimal is found. There are different optimization methods for bin-packing, such as the work of [3], [4]. In this paper, the performance of the optimization methods is not our focus, and we choose a widely used optimization tool (Google Or-Tools) in all the experiments and set a time limit (1 minute) for optimization.

In the experiments, we can apply the upper bound or the mean value to the standard optimization for the anonymous data. With the upper bound, the optimization for Algorithm 1 is ensured to be feasible for the containers. The optimization for Algorithm 2 is feasible with at least the probability of the confidence $c$ in (22). With the upper bound, the solution is feasible to the containers, but it can also lead to container space waste since the weights can be largely overestimated. With the mean value of the interval, we can avoid the overestimated weights. However, it also increases the risk that the container is overloaded, making the solution infeasible to the constraints.

### B. PERFORMANCE METRICS

To evaluate the performance of the proposed methods, we introduce different factors. Also, to mitigate the influence of the randomness for differential privacy, every experiment is carried out ten times, and the average is used as the result.

*Objective ratio* ($o/o_n$)*:* $o$ is the optimization result using the output data from the proposed algorithms, and $o_n$ is the optimization result using the original data. The optimal objective ratio is 1, since a ratio larger than 1 means more bins are used, and a ratio less than 1 means some bins must have violated the restrictions.

*Feasibility* $f$*:* For each bin $b_i$, the optimization result using the anonymous data can violate the constraints in (10). For example, two anonymous items whose weights show as $\{11.2, 13.6\}$ are loaded to a container with capacity $= 25$, but the accurate weights of these items are $\{12, 15\}$, which violates the constraint. To evaluate how often the violation happens, we use the feasibility value $f$ to represent the proportion of the bins that satisfies all the constraints using the accurate data. If $B = \{b_0, b_1 \ldots, b_m\}$ is the optimization result that uses $m$ bins to load all the items and $D(b_i)$ is the accurate weights of the items in bin $i$, then

$$f = \frac{\sum_{b_i \in B} g(b_i,\ D(b_i))}{|B|}, \tag{31}$$

$$\text{where } g(b_i,\ D(b_i)) = \begin{cases} 0 & \text{if bin } i \text{ violates constraints,} \\ 1 & \text{otherwise.} \end{cases} \tag{32}$$

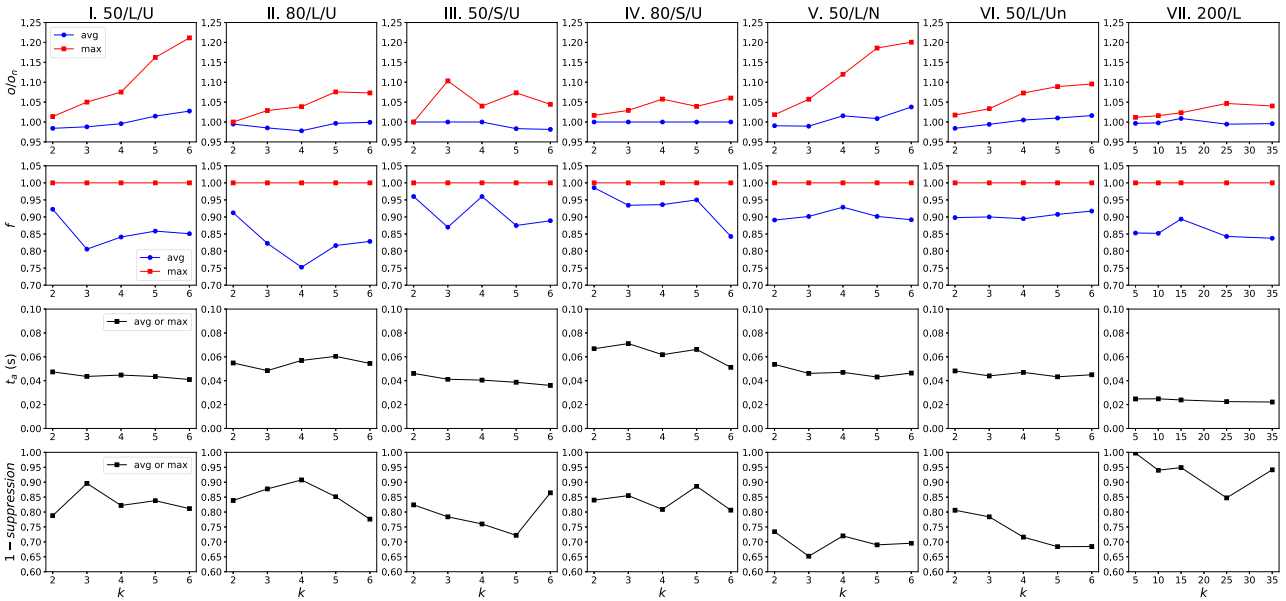


**FIGURE 5.** **Performance of the differential privacy with k-anonymity method using the average or the upper bound of intervals (with $\epsilon' = 3$). The x-axis is $k$, and the y-axis is: the objective ratio $o/o_n$, the feasibility $f$, the anonymization time $t_a$ (s), and the proportion of the remaining data $1 - suppression$.**

*Anonymization time $t_a$:* The run-time to run the proposed methods. We use the anonymization time to evaluate the efficiency of the methods.

*Suppression rate:* We introduce the suppression rate to evaluate how much data is suppressed in the differential privacy with $k$-anonymity method.

### C. PERFORMANCE OF DIFFERENTIAL PRIVACY WITH K-ANONYMITY

#### 1) INSTANCE SETTINGS

Seven different instance settings are evaluated, as shown in Table 1. Similar to the benchmarks in BPPLIB, we consider uniform distribution in instances I to IV with the same distribution as the instances used in the work of [9]. Setting I and II have different numbers of medium and large items with uniform distribution (U). Similarly, we increase the capacity from 500 to 2500 to evaluate the small items in setting III and IV. Also, we add the normal distribution to consider a different distribution. Instance VI is with a combination of two uniformly distributed sub-sets, which is also with the same distribution as used in [9]. It is with 25% large items and 75% small items. Instance VII is generated by [37] with more items (200) and the optimization is hard to be solved. This instance is supposed to show how well different algorithms work on a larger dataset.

In Table 1, $c$ is the capacity of the bins; the weights of all the items are in the range of $[l \cdot c,\ u \cdot c]$; $n$ is the number of items. Due to the suppression by $k$-anonymity, the number of items is larger than the settings for the clustering method in Table 2. In the settings, 'L' means large items, 'U' means uniform distribution, and 'N' means normal distribution.

#### 2) PARAMETER SETTINGS

In the $(\beta, \epsilon, \delta)$-DP with $k$-anonymity method, (13) and (20) show that:

$$\epsilon \geq -\ln(1-\beta) + \epsilon_1 = -\ln(1-\beta) + 2\epsilon' \cdot \frac{k}{|D|}, \qquad (33)$$

where $\beta$ is the sampling rate and $\epsilon_1 = 2\epsilon'\Delta u$ is for the $\epsilon_1$-DP mapping function. In the evaluation, we assume that the instances in Table 1 are **after** the $\beta$ sampling. We choose the number of $k \in [2, 6]$ as the independent variable to evaluate the performance since the value of $\epsilon$ (in (33)) and $\delta$ (in (14)) are both dependent on $k$. Meanwhile, we set $\epsilon' = 3$ to achieve a relatively small value of $\epsilon$. For example, with $k = 4,\ |D| = 40,\ \beta = 0.7$, we can get $\epsilon \approx 1.8$. Due to the randomness of differential privacy, we carry out every experiment ten times and use the average value for evaluation. Also, it is time-consuming to get the optimal solution for an optimization problem, so we set a time limit of 1 minute for the standard optimization.

#### 3) PERFORMANCE ANALYSIS

Fig. 5 shows the performance of the differential privacy with $k$-anonymity method. We use both the average (avg) and the upper bound (max) of the output intervals as the input to the standard optimizer. The average performs better than the upper bound in terms of objective ratio at the cost of feasibility. The weights of items are overestimated with the upper bound, leading to a larger objective ratio ranging from 1.0 to 1.2 ($k = 6$). For the same reason, the optimization results using the upper bound always satisfy all the constraints. On the contrary, the average weights are closer to the real, but the weights can be underestimated, resulting in overloaded bins.

For most settings with the upper bound, the objective ratio increases with a higher value of $k$. With a larger $k$, the exponential mechanism is more likely to choose a node with more generalization to keep a low suppression proportion. With more generalization, the upper bound is more overestimated, which increases the objective ratio. Meanwhile, the objective ratio is more close to 1 with a larger dataset. For instance setting VII, the objective ratio is close to 1 even using the upper bound.

For the flexibility, it is not always equal to 1 if the average bound is applied. The probability of violation is around 10% to 20%. To mitigate this problem, we can set the capacity a bit smaller than the real capacity. Also, in practice, we can drop some products to satisfy the constraints.

The suppression rates are different among different distributions. For uniform distributions, the suppression is around 10% to 25%, which means that only a small proportion of data are suppressed. For the normal distribution in setting V, the suppression rises to around 30% since weights are sparse for the large/small items. For a similar reason, values are sparse for the large items with the nonuniform distribution, resulting in a higher suppression (20% to 30%). When the number of items increases, the suppression rate is only with around 10% even when $k = 35$, which shows its advantages in large datasets.

The suppression also introduces a problem that not all the items are considered for bin-packing. To deal with that, there are three different approaches:

- Keep the items into the next pool and wait for $k$ items with the same range for $k$-anonymity.
- Apply differential privacy directly or apply Algorithm 2 to the suppressed data.
- Consider more about the suppression in the utility function, so the utility function can guarantee that the output is with a low suppression.

Both the low suppression rates and the low objective ratio show that the proposed utility function works well. Also, the run-time for the anonymization algorithm is less than 0.1 seconds to output an anonymous dataset. Equation (33) shows that a smaller $k$ means a smaller $\epsilon$, but this is with limits. When we calculate $\delta$ using (14), if $k$ is small, the value of $\delta$ is large. Dwork *et al.* [39] show that $\delta$ should be smaller than $1/|D|$, where $|D|$ is the number of records in the dataset. The value of $\delta$ is large with a small-scale dataset and a small $k$, but $\delta$ can satisfy it with a large dataset and a suitable $k$. For example, if $|D| = 1000$, $\beta = 0.7$, $k = 40$, $\epsilon_1 = 1$, we can get $\delta \leq 6.8 \times 10^{-4} < 1/|D|$. In real use, there are thousands of items being loaded everyday. We can select the minimum $k$, which satisfies the restriction.

### D. PERFORMANCE OF DIFFERENTIAL PRIVACY WITH CLUSTERING

#### 1) EXPERIMENTAL SETTINGS

Table 2 shows the instance setting, which is similar to the previous method. We only change the number of attributes since no suppression nor sampling is applied here.

In the evaluation, $\epsilon \in [0.5, 1, 2, 3, 4, 5]$ is the independent variable. We evaluate the performance with different confidence factors $c \in [0, 0.5, 0.7, 0.9]$. We use the upper bound for all the intervals as the input to the optimizer. Also, we carry out every experiment ten times and set a time limit of 1 minute for standard optimization.

#### 2) PERFORMANCE ANALYSIS

Fig. 6 shows the performance of the differential privacy with clustering method. The approach with a low confidence factor shows a better objective ratio but lower feasibility. Moreover, all the approaches are robust with different distributions. When $c = 0$, the output data is $\{v_i + Lap(\Delta/\epsilon)\}$, which is also the average of the intervals when $c \neq 0$. When the value of $c$ increases, the intervals become larger, and it is more probable that the accurate data is in the interval. As a result, the increasing upper bounds increase both the objective ratios and the feasibility. Also, the confidence factor can improve the feasibility at a small cost of the objective ratio when $\epsilon$ is small. For example, when $c = 0.7$, the objective ratio is around 1.2, and the feasibility is around 0.9. Although the feasibility is not always equal to 1, we can mitigate it using a smaller capacity than the real capacity. Also, in practice, the trucks can remove some products to meet the constraints.

When $\epsilon$ increases, all the objective ratios are closer to 1, and all the feasibility increases. If $\epsilon$ keeps increasing, both the feasibility and the objective ratio can converge at 1. This shows a trade-off between the privacy concern and the utility for optimization. With a larger $\epsilon$, less noise is added to the accurate data, so the algorithm has a weak privacy guarantee and good utility for the optimization work. Also, the anonymization can be finished within 3 ms.

### E. COMPARISON RESULT

#### 1) EXPERIMENTAL SETTINGS

In this section, we compare the performance of our proposed methods to the differential privacy without clustering approach as used in the work of [29] (in Fig. 7), and the work of [9] (in Fig. 8), which applies two different $k$-anonymous algorithms (k-Optimize [11] and Flash [12]) to achieve privacy-preserving data publishing. The experimental results show that k-Optimize shows the overall best performance [9], so we consider k-Optimize as the comparison method. Meanwhile, we set the minimum interval as **4** (e.g. $10 \rightarrow [8, 14] \rightarrow [8, 22]\ldots$). A smaller minimum interval means a more optimized $k$-anonymous output, but the run-time becomes longer.

The instance setting is the same as the differential privacy with clustering method in Table 2. Considering the randomness from the input dataset, we carry out each comparison experiment **ten** times and use the average as results.

#### 2) PERFORMANCE COMPARISON

Fig. 7 shows the result if only differential privacy is applied with confidence factors. It shows a similar result compared to the proposed differential privacy with clustering method. However, the clustering shows a better objective ratio. The

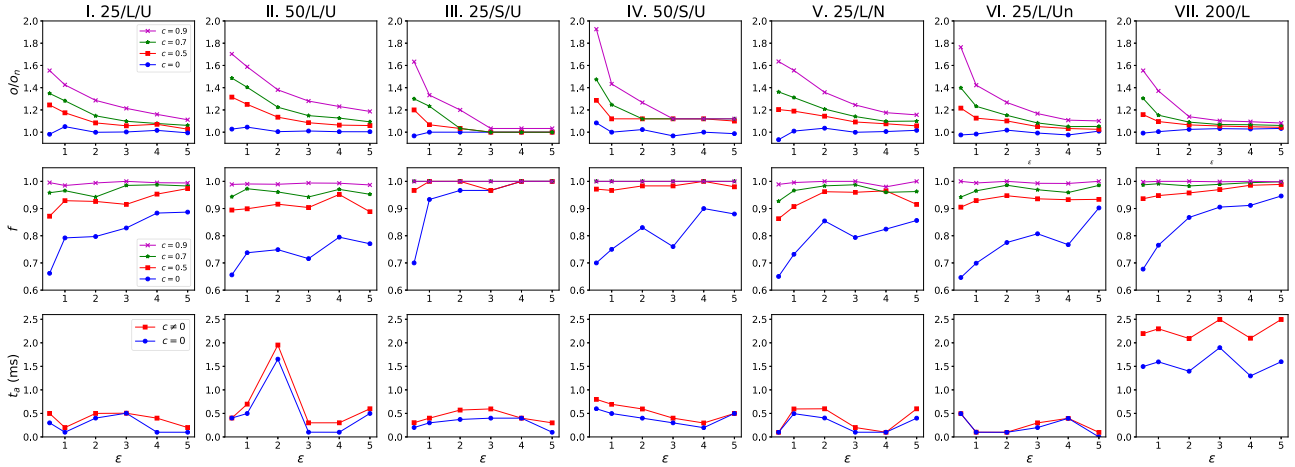


**FIGURE 6.** Performance of the differential privacy with clustering method with different confidence factor $c$. The x-axis is $\epsilon$, and the y-axis is: the objective ratio $o/o_n$, the feasibility $f$, and the anonymization time $t_a$ (ms).

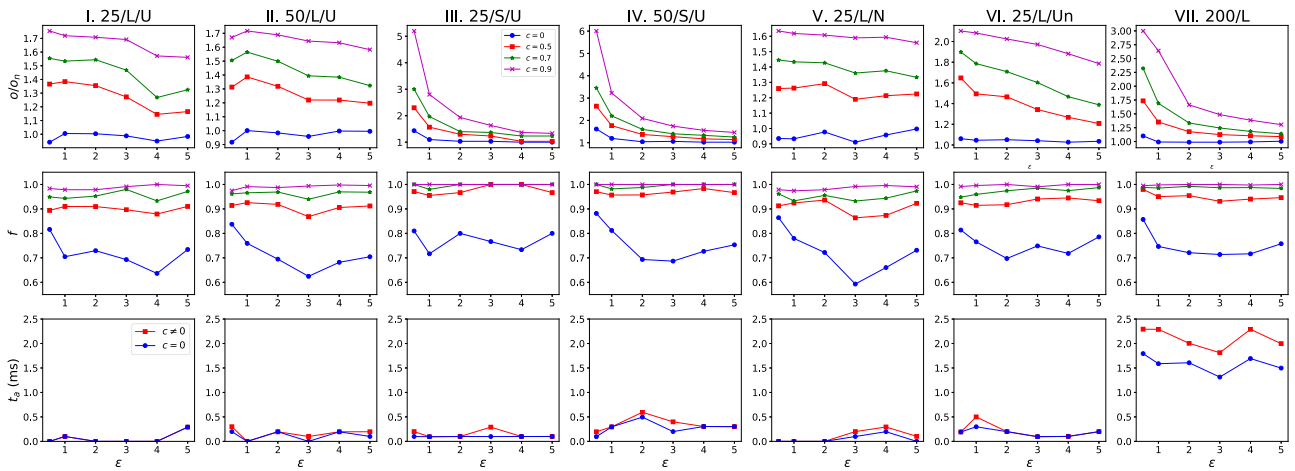


**FIGURE 7.** Performance of the comparison method (differential privacy without clustering). The x-axis is $\epsilon$, and the y-axis is: the objective ratio $o/o_n$, the feasibility $f$, and the anonymization time $t_a$ (ms).

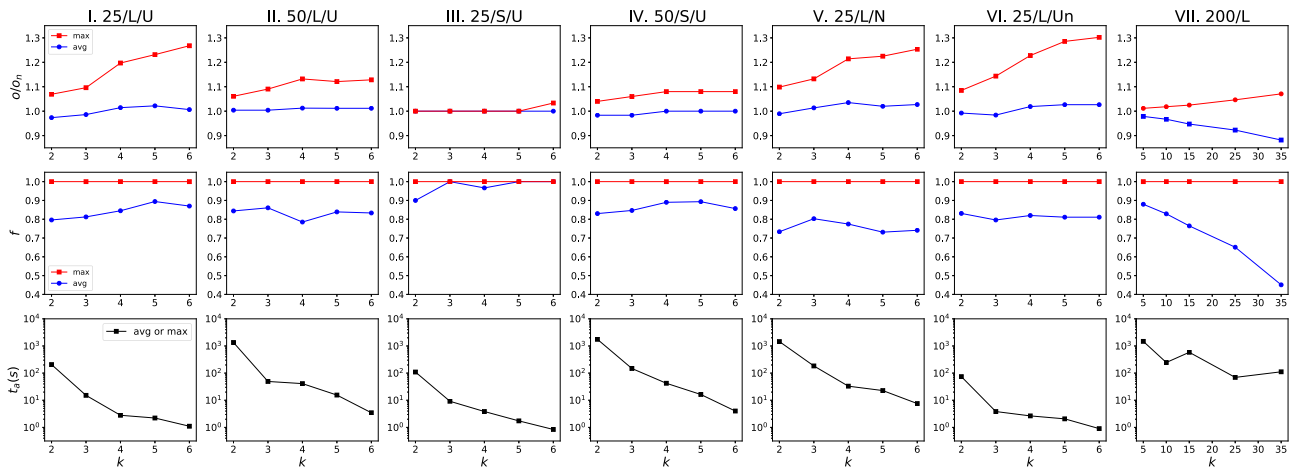


**FIGURE 8.** Performance of the comparison method (k-Optimize with standard optimizer). The x-axis is $k$, and the y-axis is: the objective ratio $o/o_n$, the feasibility $f$, and the anonymization time $t_a$ (s).

objective ratio is around 1.2 when $c = 0.7$ (with clustering), but without clustering, the objective ratio is around 1.4 when $c = 0.7$, and even higher with an increasing number of items. Meanwhile, the feasibility is closer to 1 when clustering is applied. The result shows that the proposed clustering method can improve the original differentially private method in terms of objective ratio and feasibility. Compared to the differential privacy with $k$-anonymity method, our proposed method has better objective ratio (always between 1 and 1.2) and similar flexibility.

Fig. 8 shows the performance of the k-Optimize method with the standard optimizer. We use both the average (avg) and the upper bound (max) of the intervals to show how well it works. The average shows a better objective ratio at the cost of the feasibility. The objective ratio using the upper bound of the k-Optimize output ranges from 1.1 to 1.3 for large items, and is very close to one for small items. Meanwhile, the feasibility of using the average values range from 0.8 to 0.9 for most settings, while it is very close to one for setting III and smaller than 0.8 for setting V. For all settings, a larger $k$ always leads to an increase of the objective ratio since a larger $k$ always means larger intervals in the output of the $k$-anonymous algorithm. The run-time for k-Optimize ranges from $10^0$ to more than $10^3$ seconds with 25 or 50 items.

The differential privacy with $k$-anonymity method and the k-Optimize method have shown very similar objective ratios and feasibility. Meanwhile, the differential privacy with $k$-anonymity method runs much faster than the k-Optimize, which means that we can expand the proposed method to 2-D or 4-D packing problems while k-Optimize can not. However, the proposed method is with suppression, while the k-Optimize considers all the input data. Because of the suppression, the number of rows of the input data is not the same for both methods, resulting in the differential privacy with $k$-anonymity method outperforms the k-Optimize. To better compare these methods, we compare the result of setting I for the proposed method in Fig. 5 to the result of setting II for the proposed method in Fig. 5. The proposed method is with fewer records in the input dataset, but it shows better feasibility and better objective ratio when $k \leq 4$. As a result, the proposed method can show a comparable result to the k-Optimize in terms of objective ratio and feasibility while it is much faster.

Compared to the differential privacy with clustering method, the k-Optimize method also shows a similar result. For example, when $\epsilon = 1$ and $c = 0.5$, the proposed method shows comparable objective ratios and better feasibility than the k-Optimize method ($k = 4$). With larger $\epsilon$ and smaller $k$, the proposed method also shows better objective ratios and feasibility than the $k$-Optimize method. With both the higher privacy guarantee or lower privacy guarantee, the proposed method can outperform or show comparable performance in terms of objective ratio and feasibility. Meanwhile, the proposed method is much faster.

## VII. CONCLUSION AND DISCUSSION

We propose two different privacy-preserving data publishing approaches using differential privacy to solve bin-packing problems under privacy-preserving. By calculating the probability of identifying the correct item, we prove that both proposed methods can provide better privacy guarantees than the previous work using $k$-anonymity. Using differential privacy, each item is supposed to be hidden among a group of items instead of only $k$ items by using $k$-anonymity. Also, we carry out seven different experiments based on different data distributions and a different number of inputs. The results show that our proposed methods are much faster than the $k$-anonymous approach (from $10^3$ s to less than 0.1 s) without any cost of objective ratio or feasibility. And the proposed methods are with better performance (lower objective ratio and higher or similar feasibility) than the approach only applying differential privacy. In conclusion, both proposed methods show advantages in privacy preservation and run-time over previous approaches that only apply $k$-anonymity or differential privacy while showing comparable objective ratio and feasibility. Meanwhile, both proposed methods can be used to solve 2-D or 4-D bin-packing problems, and we leave them as future works.

When we apply privacy-preserving methods, the better privacy guarantee always means the less useful output, so it is important to find the trade-off between these two aspects. In this paper, we use experiments to show the relationship between privacy guarantees ($k$ and $\epsilon$) and performance ($o/o_n$ and $f$). With some performance cost ($10\% - 20\%$ $o/o_n$ and $f$), the proposed methods can provide good privacy guarantees (such as $\epsilon = 1$). A better utility function or a better clustering method can help improve the performance of both proposed methods, and it remains as future works to find how much the utility function and the clustering can influence the performance factors.

## REFERENCES


[1] T. Zhu, G. Li, W. Zhou, and S. Y. Philip, "Differentially private data publishing and analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1619–1638, Apr. 2017.

[2] "Port of rotterdam throughput amounted to 469.4 million tonnes in 2019," Feb. 2020. Accessed: Feb. 18, 2021. [Online]. Available: https://www.portofrotterdam.com/en/news-and-press-releases/port-of-rotterdam-throughput-amounted-to-4694-million-tonnes-in-2019

[3] M. Abdel-Basset, G. Manogaran, L. Abdel-Fatah, and S. Mirjalili, "An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems," *Pers. Ubiquitous Comput.*, vol. 22, no. 5, pp. 1117–1132, 2018.

[4] H. Feng, H. Ni, R. Zhao, and X. Zhu, "An enhanced grasshopper optimization algorithm to the bin packing problem," *J. Control Sci. Eng.*, vol. 2020, 2020, Art. no. nbsp;3894987. [Online]. Available: https://doi.org/10.1155/2020/3894987

[5] A. Van den Engel and E. Prummel, "Organised theft of commercial vehicles and their loads in the European Union," Eur. Parliament. Directorate Gen. Intern. Policies Union, Policy Dept. Struct. Cohesion Policies. Transport Tourism, Brussels, no. PE 379.229, 2007.

[6] M. Essig, M. Hülsmann, E.-M. Kern, and S. Klein-Schmeink, *Supply Chain Safety Management*. Berlin, Germany:Springer, 2013.

[7] "2020 package theft statistics report," Jan. 2021. Accessed: Sep. 21, 2021. [Online]. Available: https://www.crresearch.com/blog/2020-package-theft-statistics-report

[8] ECMT, *Crime in Road Freight Transport*. Gothenburg, Sweden: OECD Publishing, 2002.

[9] R. Hoogervorst, Y. Zhang, G. Tillem, Z. Erkin, and S. Verwer, "Solving bin-packing problems under privacy preservation: Possibilities and trade-offs," *Inf. Sci.*, vol. 500, pp. 203–216, 2019.

[10] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression," SRI Int., SRI Comput. Sci. Lab., Palo Alto, CA, USA, Tech. Rep. SRI-CSL-98-04, 1998.
[11] R. J. Bayardo and R. Agrawal, "Data privacy through optimal K-anonymization," in *Proc. IEEE 21st Int. Conf. Data Eng.*, 2005, pp. 217–228.
[12] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn, "Flash: Efficient, stable and optimal K-anonymity," in *Proc. IEEE Int. Conf. Privacy, Secur., Risk Trust Int. Conf. Soc. Comput.*, 2012, pp. 708–717.
[13] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond K-anonymity," in *Proc. IEEE 22nd Int. Conf. Data Eng.*, 2006, pp. 24–24.
[14] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar, "Differentially private combinatorial optimization," in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms*, 2010, pp. 1106–1125.
[15] C. Zhang, M. Ahmad, and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 3, pp. 565–580, Mar. 2019.
[16] M. Zhang, Y. Chen, and W. Susilo, "PPO-CPQ: A privacy-preserving optimization of clinical pathway query for E-healthcare systems," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10660–10672, Oct. 2020.
[17] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., Berlin, Germany:Springer, 2006, pp. 1–12.
[18] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.*, 2006, pp. 265–284.
[19] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci.*, 2007, vol. 7, pp. 94–103.
[20] N. Li, W. Qardaji, and D. Su, "On sampling, anonymization, and differential privacy or, K-anonymization meets differential privacy," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur.*, 2012, pp. 32–33.
[21] E. C. Man Jr., M. Garey, and D. Johnson, "Approximation algorithms for bin packing: A survey," in *Approximation Algorithms NP-Hard Problems*. 1996, pp. 46–93.
[22] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. Hoboken, NJ, USA: Wiley, 1990.
[23] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 439–450.
[24] H. Lee and Y. D. Chung, "Differentially private release of medical microdata: An efficient and practical approach for preserving informative attribute values," *BMC Med. Inform. Decis. Mak.*, vol. 20, no. 1, pp. 1–15, 2020.
[25] X. Liu, Q. Xie, and L. Wang, "Personalized extended ($\alpha$, k)-anonymity model for privacy-preserving data publishing," *Concurrency Computation: Pract. Experience*, vol. 29, no. 6, 2017, Art. no. e3886.
[26] L. Sweeney, "DataFly: A system for providing anonymity in medical data," in *Database Security XI*, Berlin, Germany:Springer, 1998, pp. 356–381.
[27] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional K-anonymity," in *Proc. 22nd Int. Conf. Data Eng.*, 2006, pp. 25–25.
[28] K. El Emam *et al.*, "A globally optimal K-anonymity method for the de-identification of health data," *J. Amer. Med. Inform. Assoc.*, vol. 16, no. 5, pp. 670–682, 2009.
[29] N. Holohan, S. Antonatos, S. Braghin, and P. M. Aonghusa, "($k$, $\epsilon$)-anonymity: $k$-anonymity with $\epsilon$-differential privacy," Oct. 2017. Accessed: Feb. 19, 2021. [Online]. Available: http://arxiv.org/abs/1710.01615
[30] A. Robinson, F. Brown, N. Hall, A. Jackson, G. Kemp, and M. Leeke, "Castleguard: Anonymised data streams with guaranteed differential privacy," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr.*, 2020, pp. 577–584.
[31] M. Li, L. Zhu, Z. Zhang, and R. Xu, "Achieving differential privacy of trajectory data publishing in participatory sensing," *Inf. Sci.*, vol. 400, pp. 1–13, 2017.
[32] Z. Ma, T. Zhang, X. Liu, X. Li, and K. Ren, "Real-time privacy-preserving data release over vehicle trajectory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8091–8102, Aug. 2019.
[33] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "InCognito: Efficient full-domain K-anonymity," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2005, pp. 49–60.
[34] L. Perron and V. Furnon, Or-tools. Google. Jul. 2019. Accessed: Sep. 21, 2021. [Online]. Available: https://developers.google.com/optimization/
[35] M. Delorme, M. Iori, and S. Martello, "BPPLIB: A library for bin packing and cutting stock problems," *Optim. Lett.*, vol. 12, no. 2, pp. 235–250, 2018.
[36] E. Falkenauer, "A hybrid grouping genetic algorithm for bin packing," *J. Heuristics*, vol. 2, no. 1, pp. 5–30, 1996.
[37] A. Scholl, R. Klein, and C. Jürgens, "Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem," *Comput. Operations Res.*, vol. 24, no. 7, pp. 627–645, 1997.
[38] M. Delorme, M. Iori, and S. Martello, "Bin packing and cutting stock problems: Mathematical models and exact algorithms," *Eur. J. Oper. Res.*, vol. 255, no. 1, pp. 1–20, 2016.
[39] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.

**TIANYU LI** (Graduate Student Member, IEEE) received the B.Eng. degree in information security from Shanghai Jiao Tong University, Shanghai, China, the M.Sc. degree in informatics from the University of Edinburgh, Edinburgh, U.K. He is currently working toward the Ph.D.degree with the Cyber Security Group, Delft University of Technology, Delft, The Netherlands. His research focuses on privacy-preserving techniques using differential privacy and cryptographic tools.

**ZEKERIYA ERKIN** (Senior Member, IEEE) is currently an Associate Professor with the Cyber Security Group, Delft University of Technology, Delft, The Netherlands. His research interests include privacy enhancing technologies, particularly secure data sharing and processing: protecting sensitive data from malicious entities and service providers using cryptographic tools. He is with numerous committees, including IEEE IFS Technical Committee. He is an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE OPEN JOURNAL ON SIGNAL PROCESSING, and *Eurasip Journal on Information Security*.

**REGINALD L. LAGENDIJK** (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1985 and 1990, respectively. He was a Visiting Scientist with the Electronic Image Processing Laboratories, Eastman Kodak Research, Rochester, NY, USA, in 1991, and a Visiting Professor with Microsoft Research and Tsinghua University, Beijing, China, in 2000 and 2003. He was a Consultant with Philips Research Eindhoven from 2002 to 2005. He is a Distinguished Professor with Computing-based Society. He has more than 30 years of research and teaching experience in digital signal processing, including image and video processing, compression, search, watermarking, and digital content protection. His research interests include privacy-protected signal processing, data sharing, algorithm transparency, and meaningful human control over autonomous intelligent systems. He was elected to the Royal Netherlands Academy of Arts and Sciences (KNAW) in 2009. He was also elected Member of the Royal Holland Society of Sciences and Humanities (KHMW), and of the Netherlands Academy of Technology and Innovation (AcTI). He is a Fellow of the IEEE class of 2007, for Contributions to Image Processing. He is currently a Captain of Science (CTO) of the National Innovation Topteam Dutch Digital Delta, figurehead (boegbeeld) of the route Big Data of the Dutch National Research Agenda, Program Coordinator of the national program Digital Society, and a Member of the strategy team of the Dutch AI Coalition (NL AIC).