symmetry

MDPI

Article

Transactional Data Anonymization for Privacy and Information Preservation via Disassociation and Local Suppression

Xiangwen Liu [1,*], Xia Feng [2] and Yuquan Zhu [1]

[1] School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China; yqzhu@ujs.edu.cn
[2] School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, China; xiazio@ujs.edu.cn
* Correspondence: liuxw@ujs.edu.cn

Abstract: Ubiquitous devices in IoT-based environments create a large amount of transactional data on daily personal behaviors. Releasing these data across various platforms and applications for data mining can create tremendous opportunities for knowledge-based decision making. However, solid guarantees on the risk of re-identification are required to make these data broadly available. Disassociation is a popular method for transactional data anonymization against re-identification attacks in privacy-preserving data publishing. The anonymization algorithm of disassociation is performed in parallel, suitable for the asymmetric paralleled data process in IoT where the nodes have limited computation power and storage space. However, the anonymization algorithm of disassociation is based on the global recoding mode to achieve transactional data k^m -anonymization, which leads to a loss of combinations of items in transactional datasets, thus decreasing the data quality of the published transactions. To address the issue, we propose a novel vertical partition strategy in this paper. By employing local suppression and global partition, we first eliminate the itemsets which violate k^m-anonymity to construct the first k^m-anonymous record chunk. Then, by the processes of itemset creating and reducing, we recombine the globally partitioned items from the first record chunk to construct remaining k^m-anonymous record chunks. The experiments illustrate that our scheme can retain more association between items in the dataset, which improves the utility of published data.

Keywords: disassociation; k^m-anonymity; privacy preservation; transactional data publishing

Citation: Liu, X.; Feng, X.; Zhu, Y. Transactional Data Anonymization for Privacy and Information Preservation via Disassociation and Local Suppression. *Symmetry* **2022**, *14*, 472. https://doi.org/10.3390/sym14030472

Academic Editor: Kuo-Hui Yeh

Received: 30 January 2022
Accepted: 23 February 2022
Published: 25 February 2022

1. Introduction

In the age of IoT, large-scale data on human behavior are generated from the interaction between an IoT device and a human or devices that provide simple data, such as sensing. For example, in smart homes, interaction data can be generated when we monitor older people's interactions with their surroundings during activities of daily living for creating assisted living experiences [1]. Like market basket data, medical treatment records, and click-stream data, human behavior data are common and usually organized as transactional data (set-valued data). Publishing and sharing transactional data for statistical analysis, prediction, or critical decisions in various applications of different areas are pivotal to advances in knowledge-based services and new scientific discoveries. However, transactional data often contain detailed information about individuals. If a transactional record in a dataset is so specific that not many people can match it, there is a chance that, with the help of background knowledge, an adversary could uniquely identify the victim's record and their sensitive information. Recent research [2–4] has demonstrated such re-identification attacks in movie rating data, credit card data, and spatiotemporal positions. Therefore, there is an urgent demand for privacy-preserving transactional data publishing.

Data anonymization is the process that modifies the original dataset to impose a level of privacy on it to protect individuals' privacy. Disassociation [5], proposed by Terrovitis et al.,

is a popular anonymization technique developed for transactional data to protect privacy against re-identification attacks. Terrovitis et al. apply a more flexible privacy principle of k^m-anonymity [6], instead of traditional k-anonymity [7], on sparse multidimensional transactional data due to the curse of high dimensionality [8]. k^m-anonymity guarantees that an adversary, who has up to m items as partial knowledge of a record, cannot distinguish any record from other k-1 records. Furthermore, unlike other common methods, such as generalization [9] and perturbation [10], disassociation does not replace specific items with more general ones or add noise to the published dataset; instead, it cuts off the associations among infrequent items by separating them into different data chunks. Figure 1 shows an example of disassociation on a transactional dataset. The records in the original dataset in Figure 1a are first horizontally partitioned to two clusters, T_1 and T_2, with a maximum cluster size of five. Then, each cluster is vertically partitioned into several 3^2-anonymous record chunks and a term chunk in Figure 1b. The disassociation technique preserves the frequent items (i.e., gathering together the items satisfying k^m-anonymity into one record chunk) for frequent itemset discovery and aggregate analysis while hiding the infrequent itemsets against privacy breach (i.e., separating items of the itemset violating k^m-anonymity into different record chunks or the term chunk). This divide-and-conquer strategy enables data anonymization available in the IoT-based environment. In the IoT system composed of one central server and multiple IoT nodes, the clusters generated by horizontal partition can be asymmetrically allocated to IoT nodes according to the computing power of nodes, and the nodes perform the vertical partition algorithm on clusters in parallel to achieve k^m-anonymous data chunks.

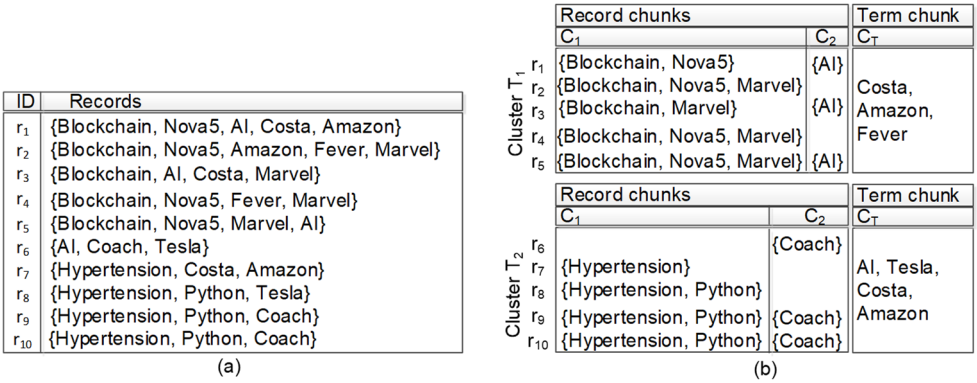

Figure 1. Example of disassociation. (**a**) Original transactional data. (**b**) Disassociated data.

Although data analysts can analyze the disassociated transactional dataset by reconstructing all possible associations between items of data chunks, it is time-consuming and hard to realize [11]. Therefore, it is essential to preserve more instances of items in data chunks to attain more associations between items, which has not yet been discussed by recent researches related to disassociation. We study the anonymization process in disassociation for this purpose. The weaknesses of the anonymization process in data utility are as follows. First, the vertical partition algorithm of disassociation, called VERPART, employs the global recoding mode [6] to partition items into different data chunks. Specifically, it checks each item in the original dataset and removes all instances of the item that violates k^m-anonymity to construct k^m-anonymous data chunks, which would lose the accurate association between the removed items with items reserved in the data chunk. Moreover, constructing k^m-anonymous data chunks by checking items one by one in vertical partition can lead to more information loss because the maximum number of items in a record chunk cannot be guaranteed due to the random selection of an item when there are more than one item with the same number of instances in a cluster. We give a detailed analysis on VERPART in Section 4.

To address the above challenges, this paper aims to design an improved disassociation scheme to preserve the data quality of anonymized datasets. Our contributions are as follows.

- We design a novel anonymization scheme for the vertical partition of transactional data. Our scheme is composed of problematic itemset identification and record chunk construction.
- Our proposed scheme employs local suppression and global partition to create the first k^m-anonymous record chunk for preserving more combinations of items.
- Based on a real transactional dataset, comprehensive experiments are conducted. The results demonstrate that our scheme can preserve data utility more effectively than previous work.

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 introduces the preliminary concepts about disassociation. Section 4 formulates the problem. We present the details of our anonymization scheme in Section 5. Section 6 shows the results and analysis of the experiments. Finally, we conclude this paper in Section 7.

2. Related Work

Many research works have been conducted to study the privacy protection techniques in data life cycle management [12–14]. As one of the research hot spots, the anonymization techniques are not only widely used to protect personal privacy in data publishing, but also to protect location privacy in location-based services [15] and personal privacy [16] in social networks [17]. This section summarizes the anonymization models, methods, and related algorithms for transactional data publishing.

2.1. On Transactional Data Anonymization

Compared with structured data with the characteristics of the fixed number of attributes and distinct discrimination of quasi-identifiers and sensitive attributes, transactional data are high-dimensional, sparse, and lack quasi-identifier attributes. Thus, the anonymization principles for structured data are not suitable for transactional data anonymization.

In privacy models for transactional data anonymization, complete k-anonymity [9], k^m-anonymity, and (h, k, p)-coherence [18] are extensions of k-anonymity. The complete k-anonymity model requires at least k records with identical items in a dataset to prevent identity disclosure by the adversary with the knowledge of any combination of items. Xue et al. [19] propose Gray-TSP, a generalization-based algorithm that maps each transactional record to a bitmap to reduce information loss and meet the complete k-anonymity principle. An anonymization system named PTA [20] groups similar transactional records and replaces each group with its centre point to achieve a complete k-anonymity of the transactional dataset. k^m-anonymity is more flexible than complete k-anonymity. It requires that any existing combination of up to m items appears at least k times. The Apriori algorithm [21] uses a bottom-up strategy based on global generalization to enforce k^m-anonymity. The local recoding algorithm (LRA) [22] is based on Apriori. It uses hierarchy-based generalization to enforce k^m-anonymity, as the vertical partition algorithm (VPA) [22] does. Loukides et al. [23] present constraint-based anonymization of transactions (COAT) to specify fine-grained privacy and utility constraints for less distortion on k^m-anonymous transactions. A clustering-based anonymizer (CBA) [24] aims to achieve k^m-anonymity of the transactional dataset by generalization and suppression so that the predefined utility policy can be satisfied. Disassociation [5] can be categorized as the bucketization technique without distinguishing sensitive values from non-sensitive ones. It horizontally and vertically partitions datasets to non-overlapping groups of k^m-anonymity and preserves the result accurately. Loukides et al. [25] proposed a disassociation-based approach to anonymize datasets about diagnosis codes, which can obtain better data quality than CBA. Complete k- and k^m-anonymity protect identity information. The (h, k, p)-coherence model protects the privacy of individuals from sensitive attribute disclosure and identity disclosure. It requires that any itemset with a length up to p is linked to at least k records and that the occurrence probability of a sensitive item in the records containing an itemset of at most p size is not higher than h. An anonymized algorithm, Greedy, is proposed in [18] to achieve (h, k, p)-coherence of a transactional dataset.

In contrast to the above models, the ρ-uncertainty model [26] provides privacy protection from attribute disclosure. It restricts the possibility of an individual correlating with any sensitive item less than the threshold ρ. Related anonymization techniques include the global generalization-based algorithm (Suppression control) and suppression-based algorithms (TD control) in [26], a partial suppression through divide and conquer approach proposed by Jia et al. [27], and personalized ρ^m- and (ε, σ)-ρ^m-uncertainty [28] for relaxed privacy guarantee of transactional datasets.

The anonymity models that we introduce above are under the uniform assumption that either the attacker has background knowledge of itemsets with a specific size, or any item can be the sensitive item to be protected. The assumption is so restrictive that excessive data transformation occurs. Given that, in reality, only some specific itemsets can cause the re-identification attack and some specific items are sensitive, researchers specify fine-grained and flexible privacy requirements. PS-rule-based anonymity [29] is the general version respecting the above models, where two sets of items, i.e., antecedent and consequent, contain those that lead to identity disclosure and sensitive information disclosure, respectively. The itemsets to be anonymized and sensitive items to be protected are specified by data publishers. The corresponding algorithm RBAT in [29] is better than the Apriori algorithm [21] in terms of data utility. Tree- and sample-based anonymity principles [30] are two generalization approaches which use PS-rule protection, and the latter is more scalable than the former. Privacy-constrained anonymity [31] provides a more flexible privacy principle. The anonymization algorithm, referred to as UGACLIP in [31], is only employed to anonymize the itemsets known by an attacker. So, it requires a predictable notion of the background knowledge of the attacker, which is challenging to obtain.

2.2. On Bucketization Technique

As a primary anonymization technique for privacy-preserving data publishing, bucketization partitions datasets into non-overlapped subsets to de-link the relation between attributes without modifying published data [32]. Anatomy [33] is the first proposed bucketization technique to protect sensitive information in relational datasets, where quasi-identifiers and sensitive values are first separated into two tables, and each table is divided into buckets. Every quasi-identifier bucket is associated with its sensitive value bucket through one common attribute, the bucket identifier. Wang et al. [34] propose a flexible and effective bucketization scheme with personalized privacy settings about sensitive values and different sizes of buckets to get better data quality. Multi-sensitive bucketization (MSB) [35] discusses the privacy-preserving problem of releasing data with multi-sensitive attributes. Based on anatomy, Liu et al. [16] design a linear time algorithm for the l-diverse dataset in the published social graph. A relational data anonymization scheme using anatomization, called SLPPA [32], performs the table and group divisions to achieve the $(\alpha, \beta, \gamma, \delta)$-privacy requirement. Slicing [36] is another bucketization-based approach that first vertically partitions attributes into columns and then horizontally divides tuples into buckets to meet l-diversity. T-closeness slicing is designed in [37] for publishing transactional data. A hybrid anonymization approach integrating both anatomisation and slicing is adopted to publish data with multiple sensitive attributes [38]. Disassociation is a bucketization technique designed for high-dimensional set-valued data anonymization. First, the horizontal partition is performed to cluster similar records into one group. Next, the vertical partition separates each group into non-overlapped subsets satisfying k^m-anonymity. Disassociation is used in [25,39] to anonymize electronic health data for better data inquiry, statistics, and analysis results.

2.3. On Suppression Technique

In the anonymization of sparse multidimensional data, such as transactions or trajectories, the instances of one attribute (one attribute represents an item in transactional data or a spatial-temporal point in trajectory data) are entirely or partly deleted, by global or local suppression, to achieve the predefined privacy principle and preserve truthful

and undeformed instances of attributes. Global suppression, employed in [18,26] for transactional data anonymization and in [40–44] for trajectory data anonymization, can incur more information loss than local suppression which only deletes the necessary instances of an attribute. In [27] and [28], improved local suppression algorithms for ρ-uncertainty anonymity of the transactional datasets are proposed, which can achieve higher data quality than global suppression [26]. The literature on trajectory data anonymization [42–44] demonstrates that local suppression can obtain better anonymity gains.

2.4. Summary of Related Work

There are various algorithms for transactional data anonymization. These algorithms can be categorized according to the attack form, privacy principles, and data transformation strategy, as shown in Table 1. Techniques based on generalization or perturbation often modify the attribute values, reducing data utility for analysis [32]. Anatomy [33] and slicing [36] are often used to anonymize relational data with a fixed number of attributes in anatomization-based techniques, while disassociation is tailored for transactional data anonymization. In recent disassociation-based works [11,25,39,45–49], some employ disassociation to anonymize electronic health data [25,39], some consider improving the horizontal partition algorithm for data quality [11,48], and others focus on the attribute disclosure risk that the disassociated dataset may suffer from [45–47], where they evaluate the privacy breach [45] and provide solutions to attribute disclosure [46,47]. These related works did not consider the information loss caused by the anonymization process (i.e., the vertical partition algorithm), which is the original intention of the study on disassociation in this paper. We employ local suppression to construct the first record chunk in the vertical partition process to preserve more associations between items in a record chunk for higher data quality.

Table 1. Anonymity models, algorithms, and methods respecting transactional data against identity and attribute attacks.

Attack	Anonymity Model	Algorithm	Method
Re-identification attack	Complete k-anonymity [9]	Gray-TSP [19]	Generalization
		PTA [20]	Generalization
	k^m-anonymity [5]	Apriori [21]	Generalization
		LRA [22]	Generalization
		VPA [22]	Generalization
		Disassociation [5]	Anatomization
		COAT [23]	Generalization and suppression
		CBA [24]	Generalization and suppression
	Privacy constrained anonymity [31]	UGACLIP [31]	Generalization and suppression
Attribute linkage attack	ρ-uncertainty [26]	Suppression control [26]	Suppression
		TD control [26]	Suppression and generalization
		Partial suppression [27]	Suppression
	personalized ρ^m-uncertainty [28]	SUPPRESSOR [28]	Suppression
	$(\varepsilon, \sigma)-\rho^m$-uncertainty [28]	SAMPLESUPPRESSOR [28]	Suppression
Re-identification and attribute linkage attack	(h, k, p)-coherence [18]	Greedy [18]	Suppression
	PS-rule based anonymity [29]	RBAT [29]	generalization
		Tree-based anonymization [30]	generalization
		Sample-based anonymization [30]	generalization

3. Preliminary Concepts

Let $\mathbb{T}$ be a transactional dataset with $|\mathbb{T}|$ records $t_1, t_2, \ldots, t_{|\mathbb{T}|}$, $\mathbb{D}$ and $|\mathbb{D}|$ be the domain and domain size of all possible items (e.g., purchased items, query terms, etc.) in $\mathbb{T}$.

A record, t_i $(1 \leq i \leq |\mathbb{T}|)$, associated with a specific individual, is a non-empty subset of $\mathbb{D}$. For an itemset $I \subseteq \mathbb{D}$, $\mathbb{T}(I)$ represents the records containing I in dataset $\mathbb{T}$ and the support of I in $\mathbb{T}$, and $S(I, \mathbb{T})$ is the number of the records containing I in $\mathbb{T}$.

A prior review of certain basic concepts is conducted in this section to clear the main idea of transactional data anonymization by disassociation.

Definition 1. *(Horizontal Partition and Cluster). A horizontal partition creates several subsets of transactional dataset $\mathbb{T}$, such that every record in $\mathbb{T}$ belongs to exactly a subset. A subset of records is called a cluster. Let there be l clusters $T_1, T_2, \ldots, T_l$, then $\cup_{i=1}^{l} T_i = \mathbb{T}$, and $T_{l_1} \cap T_{l_2} = \varnothing$, where $1 \leq l_1 \neq l_2 \leq l$.*

The horizontal partition is firstly performed to bring together similar records composed of many common items and a few uncommon ones into one cluster, which achieves the anonymity guarantee with reduced information loss in the process of vertical partition on each cluster in the next step, the vertical partition. Moreover, the vertical partition on each cluster can be carried out independently and even in parallel, which makes the anonymization efficient.

Definition 2. *(Vertical Partition and Chunk). A vertical partition separates a cluster T into several record chunks $C_1, C_2, \ldots, C_n$ and one term chunk C_T. Let D, D_i, and D_T be the corresponding domain of items respecting cluster T, each record chunk $C_i (1 \leq i \leq n)$, and the term chunk C_T. There are $D_p \cap D_q \cap D_T = \varnothing$ $(1 \leq p \neq q \leq n)$ and $D = D_1 \cup D_2 \ldots \cup D_n \cup D_T$ $(1 \leq i \leq n)$. Each record chunk $C_i = \{t \cap D_i | for\ every\ record\ t \in T\}$ is the collection of records, where duplicate records are allowed and $1 \leq i \leq n$. The term chunk $C_T = D_T$ is a set of items.*

Note that the number of record chunks is arbitrary (i.e., $n \geq 0$), and the term chunk may be empty. The vertical partition is the core process of anonymization in disassociation, following the principle of k^m-anonymity to publish a privacy preservation dataset.

Definition 3. *(k^m-Anonymity). For any itemset I with length up to m in a transactional dataset $\mathbb{T}$, if the number of records containing I in $\mathbb{T}$ is not less than k, i.e., $S(I, \mathbb{T}) \geq k$, $\mathbb{T}$ is said to be a k^m-anonymity of the transactional dataset.*

Definition 4. *(Disassociation for k^m-anonymous transactional dataset). A published transactional dataset is a k^m-anonymity of the disassociated dataset if and only if each record chunk in every cluster of the dataset, after performing horizontal and vertical partition sequentially, is k^m-anonymous.*

4. Problem Definition

Disassociation is an anonymization technique designed for transactional data. It employs k^m-anonymity, a more flexible privacy principle than k-anonymity, to anonymize data. There have been many research works related to disassociation in recent years. However, the information loss caused by the anonymity algorithm in disassociation has never been mentioned in these researches yet. Therefore, it is necessary to carry out the study on improving the anonymity algorithm for anonymized data quality. For this reason, we first discuss the information loss caused by VERPART and then introduce our solution to data in this section.

4.1. Analysis on Vertical Partition

4.1.1. Implementation Process of VERPART

Given a cluster T, D represents the item domain of T. The VERPART algorithm separates the records of cluster T into chunks by the following steps.

(1) Sort the items in D, in descending order.
(2) Remove the items with less than k instances in T, from D, into the term chunk C_T.

(3) Construct the domain of items D_i respecting each record chunk $C_i (1 \leq i \leq n)$ by adding the remaining items in D into D_i one by one by descending order, to examine whether the instances of the new combinations of any up to m items appear at least k times.

Figure 1b is the result dataset after performing VERPART on each cluster generated by the horizontal partition of the original dataset in Figure 1a. However, the disassociated dataset, after the implementation of VERPART, still has deficiencies in data quality.

4.1.2. Data Quality Analysis

We propose two cases in which the algorithm reduces data quality.

Case 1: VERPART cannot guarantee the maximum size of the domain of items respecting each record chunk, which degrades the data quality.

To explain Case 1, we give an example in Figure 2 to show the vertical partition process on cluster T (Figure 2a) by algorithm VERPART for 2^2-anonymity. Item f is first moved into the term chunk because $S(\{f\}, T) = 1$ and then items a and e are sequentially put into the item domain of the first record chunk C_1, shown as Figure 2b. Next, one of the two items c or d can be randomly selected as the candidate due to the same number of instances in T according to the VERPART algorithm. Suppose that item d is selected to add into the item domain of C_1, the disassociated dataset T_1' is shown as Figure 2c. However, if item c is selected, the result T_2' is shown as Figure 2d.

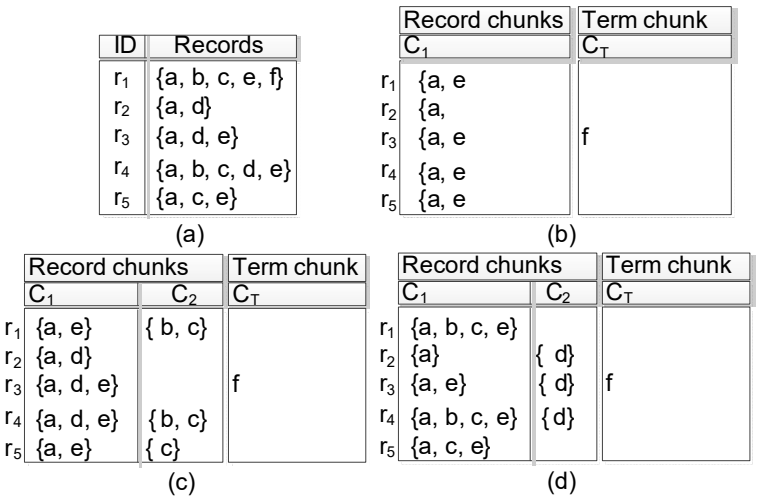

Figure 2. Example for analysis on implementing VERPART. (**a**) Original cluster T. (**b**) Intermediate process on vertical partition. (**c**) One result cluster T_1' after vertical partition. (**d**) The other result cluster T_2' after vertical partition.

As shown in Figure 2, choosing one item randomly from several candidates with the same number of instances in a cluster generates different result dataset. The data utilities of these two datasets are different. The data quality of dataset T_2' in Figure 2d is superior to T_1' in Figure 2c for fewer items are separated from record chunk C_1 in T_2' compared with T_1', which means more correlations of items are preserved.

Case 2: What is applied in Algorithm VERPART is a global recoding of disassociation to separate itemsets violating k^m-anonymity for transaction data anonymization.

For example, consider the records of a cluster T in Figure 3a with $k = 2$ and $m = 2$. Itemset $\{a, e\}$ is problematic because $S(\{a, e\}, T) = 1$. To eliminate the problematic itemset $\{a, e\}$, we employ VERPART and achieve the result cluster T_1', as shown as Figure 3b. In cluster T_1', the accurate associations between item e and the other items in cluster T are broken, which seriously degrades the accuracy of correlation analysis. Notice that if we only remove the instance of item a in record r_1, shown in Figure 3c, the problematic itemset $\{a, e\}$ can be eliminated as well. The remaining itemsets associating item a still keep

2^2-anonymity in T_2' in Figure 3c. Intuitively, combining global vertical partition with local recoding (such as local suppression [43]) helps to improve data utility.

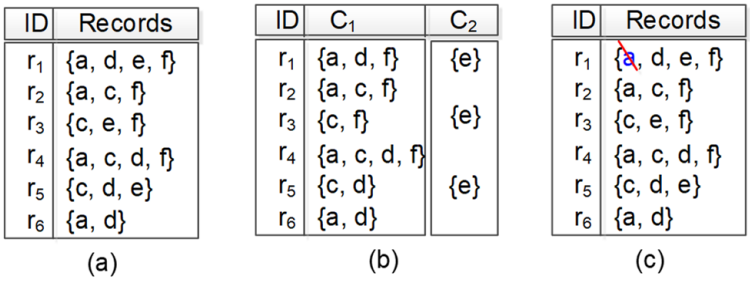

Figure 3. Example for local suppression. (**a**) Original cluster T. (**b**) The result cluster T_1' after vertical partition. (**c**) The result cluster T_2' after local suppression of item a.

4.2. Solution Statement

To address the above problems, we present a novel vertical partition scheme based on the anonymization techniques, i.e., disassociation and local suppression (DLS), to preserve data utility while protecting the privacy of disassociated data.

Figure 4 shows the whole anonymization framework of our DLS scheme, where each cluster is anonymized after the horizontal partition and then follows data refining. Our work mainly focuses on DLS. It performs the following procedures to conserve utility and privacy of the published data.

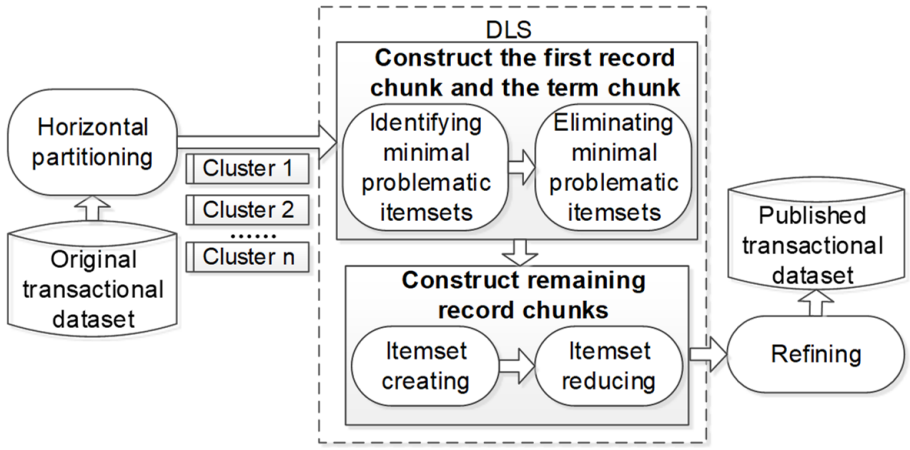

Figure 4. Architecture of our anonymization scheme.

(i) DLS first identifies all itemsets violating k^m-anonymity in each cluster and employs local suppression and global partition to eliminate them to construct the first k^m-anonymous record chunk and the term chunk.

(ii) Then, DLS recombines the globally partitioned items by enumerating all the itemsets meeting with k^m-anonymity requirement and iteratively choosing the itemset with maximum size based on a greedy strategy as the domain of the new record chunk, to generate the remaining record chunks.

Definition 5. *(Transactional data anonymization by disassociation and local suppression). Given the transactional dataset $\mathbb{T}$ and k^m-anonymous privacy requirement, the goal of anonymization of $\mathbb{T}$ is to achieve a sanitized version of $\mathbb{T}$ by using the DLS scheme such that the published dataset not only satisfies k^m-anonymity but also preserves more correlation between original items for data utility.*

5. DLS Scheme and Algorithms

In this section, we first describe the details of our DLS scheme and then present a set of algorithms to implement k^m-anonymous disassociation.

5.1. First Record Chunk Construction

Inspired by the approach proposed in the literature [43], we identify all itemsets violating k^m-anonymity and eliminate them by local suppression and global partition, to create the first record chunk.

5.1.1. Minimal Problematic Itemset Identification

If an adversary, holding the background knowledge of items with a length up to m, can link the target individual to his record with the one-to-one relationship or high probability, the adversary has successfully launched an identity linkage attack, and the privacy breach occurs. Given an itemset I with $S(I,T) > 0$ in a cluster T, itemset I is regarded as a problematic itemset respecting k^m-anonymity requirement if the number of records containing I in T is less than k.

Definition 6. *(Problematic Itemset). Let D be the domain of items respecting a cluster T of a transactional dataset, and I be a subset of D with $S(I,T) > 0$. For any I in T, if the number of records containing I in T is less than the preset threshold k, i.e., $S(I,T) < k$, I is called a problematic itemset respecting T.*

To achieve transactional data k^m-anonymization, it is sufficient for us to remove all problematic itemsets from the dataset. However, it is not feasible to enumerate all possible problematic itemsets and then eliminate them for the number of problematic itemsets is huge [40,43]. To that end, we take a much more efficient way similar to work [40,43] for k^m-anonymity. We only identify and eliminate those which are not super itemsets of any problematic itemset, named minimal problematic itemsets. When considering a problematic itemset I in cluster T, any super itemset of I, denoted by I', is still problematic for $S(I',T) \leq S(I,T) < k$. So, identifying and eliminating minimal problematic itemsets is much more efficient than that of problematic ones.

Definition 7. *(Minimal problematic itemset (MPI)). Itemset I is a minimal problematic itemset if each sub-itemset of I is not a problematic itemset.*

We prove that eliminating all MPI respecting a cluster T can guarantee the elimination of all privacy breach violating k^m-anonymity.

Theorem 1. *A transactional dataset $\mathbb{T}$ is said a k^m-anonymous dataset if there is no MPI in any cluster T of $\mathbb{T}$.*

Proof of Theorem 1. Suppose transactional dataset $\mathbb{T}$ is a k^m-anonymous dataset even if there is an MPI in a cluster T of $\mathbb{T}$. According to Definition 3 and Definition 4, there exists at least a record chunk in T such that T must contain problematic itemsets. According to Definition 7, a problematic itemset either is an MPI or contains an MPI, which stands in contradiction to the initial assumption. Therefore, $\mathbb{T}$ is a k^m-anonymous dataset. □

5.1.2. Minimal Problematic Itemset Elimination

To eliminate an MPI $I(|I| > 1)$ in cluster T, local suppression is employed to delete partial instances of items in I from T. The example for Case 2 in Section 3 has shown the main idea of local suppression, where more original instances of items are preserved in the cluster. However, the performance of local suppression may generate new MPI, which leads to expensive computational cost on identifying newly generated MPI and no guarantee on limited iterations for eliminating MPI [43]. We redefine the notion of valid

local suppression for transactional data anonymization to avoid generating MPI when performing local suppression.

Definition 8. *(Valid local suppression). Given a cluster T to be k^m-anonymized, an MPI I respecting k^m-anonymity in T, and $T(I)$ the records containing I in T, the item domain of I and $T(I)$ are denoted as D_I and $D_{T(I)}$, respectively, and T_r represents the records in T, except $T(I)$, i.e., $T_r = T - T(I)$. For all the non-problematic itemsets $\mathbb{I}$ composed of an item $v \in D_I$ and any other at most $m-1$ items in $D_{T(I)} - D_I$, if the number of any itemset $I^* \in \mathbb{I}$ in T_r is equal to 0 or no less than k, i.e., $S(I^*, T_r) = 0$ or $S(I^*, T_r) \geq k (\forall I^* \in \mathbb{I})$, MPI I can be eliminated by the valid local suppression of item v from cluster T, and item v is called a valid item for local suppression respecting I in cluster T.*

We review the example given in Section 3 and Case 2, shown in Figure 3. Since itemset $\{a, e\}$ is an MPI with $k = 2$ and $m = 2$ respecting cluster T in Figure 3a, we need to eliminate $\{a, e\}$. Items a and e are candidate items to be locally suppressed. Only record r_1 contains $\{a, e\}$ in cluster T, i.e., $T(a, e) = \{r_1\}$. So, $\{a, e\}$ can be eliminated by deleting item a or e in r_1. If item a is deleted as shown in Figure 3b, the number of itemsets $\{a, d\}$ and $\{a, f\}$, both of which initially meet with k^m-anonymity, will be reduced by 1 in T. To avoid generating new MPI in T, it is necessary to check the number of the remaining instances of $\{a, d\}$ and $\{a, f\}$. Note that $T(\{a, d\}) = T(\{a, f\}) = 2$ in Figure 3b. So, MPI $\{a, e\}$ can be eliminated by the valid local suppression of item a from cluster T and item a is a valid item for local suppression respecting MPI $\{a, e\}$ in T. However, item e is not a valid item. The reason is if we delete item e from r_1 in T, the number of itemsets $\{d, e\}$ and $\{e, f\}$ is reduced to 1, both of which become new MPI.

If more than one valid item can be locally suppressed when eliminating an MPI I in a cluster T, choosing a *good* item $v \in I$ for *better* result is required.

Generally, suppressing item v improves privacy protection and decreases data utility. To find a good item v for the sub-optimal trade-off between privacy protection and data utility, we define the function of the anonymity gain metric $AG(v, I)_T$ as:

$$AG(v, I)_T = \frac{PG(v, I)_T}{IL(v, I)_T} \tag{1}$$

where $PG(v, I)_T$ is the number of MPI that are eliminated when we eliminate an MPI I by deleting the item v in I from all records containing I in cluster T (representing the privacy protection gain), and $IL(v, I)_T$ is the number of instances of v in all records containing I in cluster T (representing the information loss). More $PG(v, I)_T$ and less $IL(v, I)_T$ achieve more $AG(v, I)_T$, which leads to a better anonymization effect.

If none of the items in an MPI I is the valid item for local suppression in a cluster T, we select the item v with the best $AG(v, I)_T$ in T and employ the method of item partition to separate all instances of v from T to eliminate I. Item partition is a global coding method for anonymization and no new MPI generates after performing of each item partition. Note that $PG(v, I)_T$ here represents the number of MPI that is eliminated by partitioning v from T and $IL(v, I)_T$, i.e., the number of instances of v in T.

5.2. Remaining Record Chunk Construction

5.2.1. Itemset Creating

Itemset creating generates all itemsets that are item domains of the sets of all possible k^m-anonymous sub-records of a cluster. We enumerate all itemsets meeting with the conditions: (*i*) each itemset is a subset of globally separated items; (*ii*) each itemset is a non-problematic itemset. Specifically, beginning with the itemsets with size $i = 2$, we iteratively generate all the itemsets with size i by self-join between itemsets with size $i - 1$. Within each iteration, a pruning strategy that deletes the itemsets containing MPI from the newly generated candidate itemsets with size i is used, according to the apriori property

proposed below, to obtain the item domains of all possible k^m-anonymous data chunks with domain size i and take them to create candidate itemsets with size $i+1$.

Theorem 2. All the subsets of an itemset do not contain MPI if the itemset is not a problematic itemset in a cluster.

Proof of Theorem 2. Suppose I', a subset of itemset I contains an MPI I_p, even if the itemset I is not a problematic itemset in cluster T. (*i*) When $I_p = I'$, we have $0 < S(I, T) \leq S(I', T) < k$. So, I must be a problematic itemset according to Definition 6, which contradicts the initial assumption. (*ii*) When $I_p \subset I$, let $I^* = I' - I_p$, we have $0 < S(I,T) \leq S(I',T) = S(I_p \cup I^*, T) \leq \min(S(I_p,T), S(I^*,T)) < k$ for $S(I_p,T) < k$. So, itemset I must be a problematic itemset according to Definition 6, which contradicts the initial assumption. □

5.2.2. Itemset Reducing

Itemset reducing uses a greedy strategy to generate the item domains of the remaining record chunks. Specifically, we take all the itemsets obtained in the previous step as the initial objects to be processed. The specific procedure is as follows:

(1) randomly select an itemset with the maximum size as the current itemset;
(2) delete the items that also exist both in the current itemset and the remaining itemsets;
(3) repeat steps (1) and (2) until the remaining itemsets are all empty;
(4) construct the remaining record chunks by projecting all the current itemsets created in each iteration, which compose the collection of item domains, to the correspondent cluster to be anonymized.

5.3. Anonymization Algorithms

We developed a set of algorithms for the design scheme DLS to perform the anonymization process. The specific implementation is as follows.

5.3.1. First Record Chunk and Term Chunk Construction

We first identify and eliminate all MPI respecting a cluster to generate the first record chunk and the term chunk.

1. MPI Identification

MPI identification (MPII) algorithm is detailed in Algorithm 1. In Algorithm 1, Ca_i, U_i, and P_i represent all candidate MPI, non-problematic itemsets and MPI with size i, respectively. The MPII algorithm first puts all items in cluster T into Ca_1 (line 1). Then, for each itemset I in $Ca_i (1 \leq i \leq m)$, the number of I in cluster T is computed. If I satisfies k-anonymity respecting T, i.e., $S(I,T) \geq k$, I is added to U_i for creating Ca_{i+1}, the candidate set of MPI with size $i+1$, otherwise, and I is added to P_i (lines 4–13). Next, the candidate set of MPI with length $i+1$, Ca_{i+1} is generated in two steps. First, a self-join of U_i (denoted as $U_i \bowtie U_i$) is conducted (line 15). Second, all the super itemsets of the identified MPI are deleted from Ca_{i+1} (lines 16–20). The MPII algorithm outputs all MPI P respecting cluster T.

Algorithm 1 MPII algorithm.

Input: Cluster T, thresholds k and m
Output: The set of MPI P
1: $Ca_1 \leftarrow$ *all distinct items in* T, $P_1 \leftarrow U_1 \leftarrow \varnothing$;
2: $I \leftarrow 1$;
3: **while** $i \leq m$ *and* $Ca_i \neq \varnothing$ **do**
4: **for** each itemset I in Ca_i **do**
5: Scan T once to compute $|S(I,T)|$, for any $I \in T$;
6: **if** $|S(I,T)| > 0$ **then**
7: **if** $|S(I,T)| < k$ **then**
8: Add I to P_i;
9: **else**
10: Add I to U_i;
11: **end if**
12: **end if**
13: **end for**
14: i++;
15: Create candidate problematic itemsets Ca_i by $U_{i-1} \bowtie U_{i-1}$;
16: **for** each itemset $I' \in C_i$ **do**
17: **if** I' is a super set of any itemset in P_{i-1} **then**
18: Remove I' from Ca_i;
19: **end if**
20: **end for**
21: **end while**
22: **return** $P = P_1 \cup P_2 \cup \ldots \cup P_m$;

2. MPI Elimination

The MPI elimination (MPIE) algorithm is performed to remove all identified MPI P from cluster T by local suppression and item partition. MPIE algorithm is presented in Algorithm 2. Firstly, all MPI of length 1 are put into C_T to form the term chunk about cluster T (line 1) and remove them from P (line 2). Then, we update T and its domain D by removing all instances of items in C_T from T and items in C_T from D, respectively (lines 4–5). The updated T and D are, respectively, denoted by D'_1 and C'_1. Next, Algorithm 3 *item_disasso*(C'_1, D'_1, P) is performed to eliminate all MPI in P by locally suppressing or globally partitioning items in C'_1 and returns D_1, the domain of items of the first created record chunk (line 6). By projecting cluster T to D_1, MPIE gets the first record chunk C_1 (line 7). Finally, MPIE removes items in D_1 from D'_1. The remaining items D_r is the item domain of remaining record chunks (line 8). Algorithm 2 outputs C_1, C_T, and D_r (line 9).

Algorithm 2 MPIE algorithm.

Input: Cluster T, the MPI set P, thresholds k and m
Output: The first record chunk C_1, the term chunk C_T respecting cluster T and items D_r excluded from C_1 *and* C_T
1: $C_T \leftarrow \cup\, I$, *where* $I \in P$ *and* $|I| = 1$;
2: $P \leftarrow P - C_T$;
3: $D \leftarrow$ *all distinct items in* T;
4: $D'_1 \leftarrow D - D_T$;
5: $C'_1 = \{D'_1 \cap r \mid$ *for every* $r \in T$;
6: $D_1 = item\_disasso(C'_1, D'_1, P)$;
7: Create the first record chunk C_1 by projecting T to D_1;
8: $D_r = D'_1 - D_1$;
9: **return** C_1, C_T *and* D_r;

To construct the first record chunk, Algorithm 3 first puts all MPI into set P' to retain all original MPIs of cluster T (line 1). Then, for every MPI I_P, Algorithm 3 checks whether there are effective locally suppressed items in I_P by calling Algorithm 4 $Check\_local\_sup(v, P', I_P, C'_1)$.

For any item $v \in I_P$, if v can be locally suppressed, Algorithm 3 puts v *and* I_P into V_L (lines 3–7) and calculates the anonymity gain of each item in I_P by Formula (1) (line 8). Next, the item with the highest anonymity gain v' is selected for eliminating I'_P, the MPI that v' belongs to (line 11). Algorithm 3 deletes all the instances of v' in C'_1 if v' and I'_p are not in V_L. Otherwise, Algorithm 3 deletes the instances of v' from the records containing I'_P in C'_1 (lines 12–17). Finally, the item domain D'_1 respecting records C'_1 and the set of MPI P' are updated (lines 18–20). Algorithm 3 performs the above process iteratively until all MPI are eliminated. Algorithm 3 returns the item domain D_1 of the first record chunk.

Algorithm 3 *item_disasso*$(C'_1,\ D'_1,\ P)$.

1: $P' \leftarrow P$;
2: **while** $P' \neq \varnothing$ **do**
3: **for** each $I_P \in P'$ **do**
4: **for** each $v \in I_P$ **do**
5: **if** *Check_local_sup*(v, P', I_P, C'_1) **then**
6: $V_L \leftarrow (v, I_P)$;
7: **end if**
8: Calculate $AG(v, I_P)_{C'_1}$ by *Formula* (1);
9: **end for**
10: **end for**
11: $v' \leftarrow$ *the item with the highest AG*;
12: **if** $(v', I'_P) \notin V_L$ **then**
13: Remove all instances of v' in C'_1;
14: $D'_1 = D'_1 \setminus \{v'\}$;
15: **else**
16: Remove instances of v' from $C'_1(I'_P)$;
17: **end if**
18: Update P' by deleting all the eliminated MPI when removal of v', from P';
19: **end while**
20: $D_1 \leftarrow D'_1$;
21: **return** D_1;

Algorithm 4 checks whether an item $v \in I_P$ can be locally suppressed or not to eliminate MPI I_P, i.e., whether at least a new MPI will be generated after deleting the instances of item v from the records $C'_1(I_P)$. According to Definition 8, the items that are contained in record sets both $C'_1(I_P)$ and $C'_1(v) - C'_1(I_P)$ are all possible items composing the new generated candidate MPI. So, they are put into set W (line 1). However, the items in set P', which together with item v constitute MPI, are deleted from set W (line 2). Then, for each new candidate MPI I in candidate MPI set S, the number of I is computed (lines 3–4). If a new MPI I in S is generated, i.e., $0 < |I| < k$, which means item v cannot be a valid item for local suppression of MPI I_P respecting C'_1, Algorithm 4 returns false. Otherwise, Algorithm 4 returns true (lines 5–10).

Algorithm 4 *Check_local_sup*(v, p', I_p, C'_1).

1: $W \leftarrow$ *distinct item w such that* $w \in C'_1(I_P) \wedge w \in \left(C'_1(v) - C'_1(I_P)\right)$;
2: Remove all items, except v, in $P'(v)$ from W;
3: $S \leftarrow$ all possible itemsets containing v with length of at most m generated from W;
4: Scan $C'_1(v) - C'_1(I_P)$ once to compute $|I|$ for each itemset $I \in S$;
5: **for** each $I \in S$ with $|I| > 0$ **do**
6: **if** $|I| < k$ **then**
7: **return** *false*;
8: **end if**
9: **end for**
10: **return** *true*;

5.3.2. Remaining Record Chunk Construction

We perform the remaining record chunk construction (RRCC) algorithm to construct other record chunks about cluster T based on the strategy of itemset creating and reducing. Note that the domain of items of the remaining record chunks is the globally partitioned items D_r generated in Algorithm 2.

RRCC algorithm presents the details of constructing remaining record chunks, shown in Algorithm 5. In Algorithm 5, D_i^C represents all non-problematic itemsets with size i, i.e., all possible i-dimensional item domains respecting remaining record chunks, and the collection of all D_i^C is denoted by D^C. We first put the MPI composed of items in D_r into P^* for the following pruning operation (line 1). Then, we construct the set of *1*-dimensional item domains D_1^C, where each item domain is an itemset composed of every distinct item in D_r (line 2), and put D_1^C into D^C to initialize it (line 3). Then, we create item domains of i ($i > 1$) in two steps. In the first step, a self-join of D_{i-1}^C 1 ($i > 1$) is conducted to generate candidate i-dimensional item domains D_i^C (Line 7). In the second step, according to Theorem 1, we prune the itemsets from D_i^C, which are in P^* as well (line 8), and merge the remaining non-problematic itemsets in D_i^C into D^C (lines 9–11). The above process of item domain creating is performed iteratively until there is no new created itemset. Next, the process of item domain reducing is performed as follows. In each iteration (lines 13–22), we first select the itemset with the maximum size in D^C, denoted by I_{cur} (line 14); delete the items contained in I_{cur} from all the itemsets in D^C and the resulting empty sets (lines 15–20); and then save I_{cur} to D_r^c, which represents the set of the item domains of remaining record chunks (line 21). Finally, we project cluster T to every itemset I in D_r^c to generate the remaining record chunks and output them (lines 23–24).

Algorithm 5 RRCC algorithm.

Input: Cluster T, all MPI P, item domain of remaining record chunks D_r
Output: Remaining record chunks $C_2, \ldots, C_n$ *of* T
1: $P^* = \{I | I \in P \wedge d \in D_r,$ *for any item d in an itemset* $I\}$;
2: $D_1^C = D_1^C \cup \{d\},$ *for any item* $d \in D_r$;
3: $D^C = D^C \cup D_1^C$;
4: $i = 1$;
5: **while** $|D_i^C| > 1$ **do**
6: $i++$;
7: $D_i^C = D_{i-1}^C \bowtie D_{i-1}^C$;
8: Delete all itemsets in P^* from D_i^C;
9: **if** $D_i^C \neq \varnothing$ **then**
10: $D^C = D^C \cup D_i^C$;
11: **end if**
12: **end while**
13: **while** $D^C \neq \varnothing$ **do**
14: $I_{cur} \leftarrow$ *the itemset with maximum size in* D^C;
15: **for** each $I \in D^C$ **do**
16: $I = I - I_{cur}$;
17: **if** $I == \varnothing$ **then**
18: Delete I from D^C;
19: **end if**
20: **end for**
21: $D_r^C = D_r^C \cup I_{cur}$;
22: end while
23: Generate remaining record chunks $C_2, \ldots, C_n$ by projecting T to I, for each $I \in D_r^C$;
24: **return** remaining record chunks $C_2, \ldots, C_n$;

6. Experiments

In this section, a series of experiments are conducted to assess our proposed scheme DLS in terms of privacy preservation and data utility.

6.1. Experimental Data and Setup

For our experiments, we use a real-world dataset BMS-WebView-1 (referred to as *WV2)*, detailed in [50], to evaluate the performance of our scheme. *WV2* contains click-stream data from an e-commerce web site over several months. The dataset is one of the most popular public datasets and commonly used as the benchmark in the data mining community. *WV2* is comprized of 77512 transactions, whose maximum and average size are 161 and 5.0, respectively, with a domain size of 3340. The experiments are performed on a machine with Intel (R) Core (TM) i3-3240 CPU @ 3.40 GHz, 4 GB RAM, and the algorithms are implemented in C++.

We first assess the privacy disclosure risk of the raw dataset under different protection levels by calculating the proportion of the MPI with variant lengths in the total itemsets with the same length. Next, to comprehensively study the effectiveness of our anonymization scheme, we evaluate the effect of parameters k and m on utility loss and implement the disassociation algorithm in [5] to compare our scheme DLS in data utility. Two utility metrics, i.e., average itempair number ratio (ANR) and average relative error (ARE), are employed to evaluate data utility. The query workload used to compute ARE is the queries of retrieving frequent itemsets in *WV*2. The top 20% itemsets with length 2 in clusters are retrieved for simplicity. The default values of k and m are 10 and 2, respectively, unless specific remark.

6.2. Necessity of Privacy Preservation

We execute the MPII algorithm over clusters to get MPI set by fixing $m = 5$ and setting the value of k to 2, 3, 4, 5, 6, and 11. Then, we calculate the proportion of MPI in the same length itemsets in each cluster, respectively, and average the values. The result is shown in Figure 5.

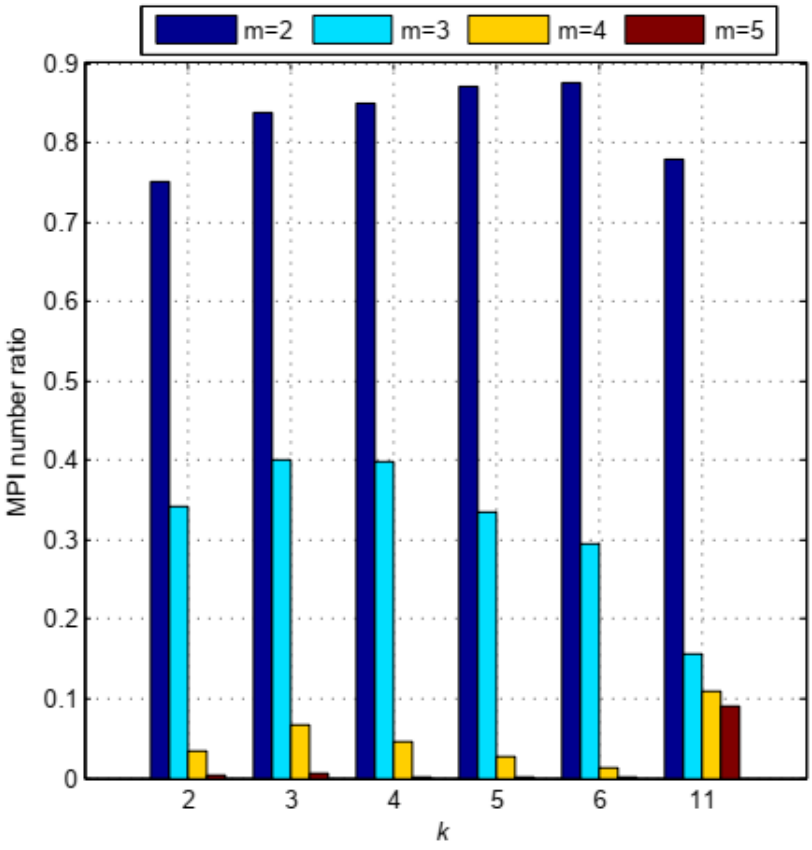

Figure 5. Privacy disclosure risk of the raw dataset under different protection levels.

Figure 5 shows that about 75% of all sets of two items appear in one record ($k = 2$), which means that adversaries, holding these sets of items, can uniquely re-identify target individuals. Moreover, over 75% of itempairs are problematic itemsets no matter how much privacy protection requirement is needed, the proportion of which is more than twice as those of length 3. There is a small proportion of MPI of length 4 and 5. These can be used as the prior knowledge of anonymization for parameter settings to achieve better anonymity gain.

6.3. Data Utility

We first introduce ANR and ARE. Then, we study the effect of k and m on k^m-anonymity under the metric ARE. The experiments compare data quality between our scheme DLS and disassociation (Disa, for short) in [5] in the end.

6.3.1. Measuring Utility

ANR calculates the average ratio of the number of accurate itempairs in each anonymized cluster to the itemsets of length 2 in the correspondent original cluster. More retained itempairs improve the query precision of the reconstructed dataset, which brings higher data utility. ANR is given by

$$\text{ANR} = \frac{\sum_{T \in \mathbb{T}} \frac{\sum_{rc \in C} |I_{rc}|}{|I_T|}}{Num_T} \tag{2}$$

where C represents all record chunks generated after anonymizing a cluster T in the original transactional dataset $\mathbb{T}$; Num_T is the number of clusters in $\mathbb{T}$ after horizontal partition; and $|I_{rc}|$ and $|I_T|$ are the number of all itemsets with size 2 in a record chunk $rc \in C$ and cluster T, respectively. Note that I_T here represents the itempairs composed of items excluding minimal problematic items in cluster T.

ARE is one of the general metrics to measure data utility. ARE calculates the average ratio of the number of accurate itempair instances in each anonymized cluster to the itemset instances of length 2 in the corresponding original cluster. We design a count(*) query Q on the original and anonymized clusters, where the top 20% frequent itemsets $\mathbb{I}$ of length 2 in each original cluster are selected. ARE is calculated as

$$\text{ARE}_Q = \frac{\sum_{T \in \mathbb{T}} \frac{\sum_{I \in \mathbb{I}} \frac{|T(I)| - \sum_{rc \in C} |rc(I)|}{|T(I)|}}{|\mathbb{I}|}}{Num_T} \tag{3}$$

where C represents all record chunks generated after anonymizing a cluster T in the original transactional dataset $\mathbb{T}$; Num_T is the number of clusters in $\mathbb{T}$ after horizontal partition; rc is a record chunk contained in C; and $|rc(I)|$ and $|T(I)|$ are the number of records containing a top 20% frequent itemset $I \in \mathbb{I}$ in the anonymized record chunks C and the corresponding original cluster T, respectively, in the transactional dataset $\mathbb{T}$. Similar to ANR, itemset I is composed of items excluding minimal problematic items in cluster T.

6.3.2. Results

Figure 6 shows the effects of parameters k and m on utility loss. We first vary the parameter m from 2 to 5 while fixing $k = 10$, on the top frequent itemsets with length 2 to assess the effect of m under ARE, as demonstrated in Figure 6a. ARE scores gradually increase with m. The reason is that with an increase in m, the number of MPI increases too; thus, more items are locally suppressed or globally partitioned from clusters. As a result, the number of instances of itemsets is smaller, which causes high ARE. Figure 6b shows the impact of k on the utility loss while fixing $m = 2$. When k is small, the ARE is low for all sets of itempairs because more items in the cluster can be preserved to create candidate itempairs. As the value of k increases, ARE scores increase and then decrease since, if the value of k is large, items that can be preserved in the record chunks are all frequent items, which degrades the number of problematic itempairs. Figure 6a,b also show that

the top 10% frequent itempairs have the lowest ARE while the top 30% frequent itempairs have the highest ARE. This is because there are more itempairs violating k^m-anonymity in the frequent itempairs when the frequent itempair domain increases, which reduces the accuracy of the instances of itempairs in the anonymized record chunks.

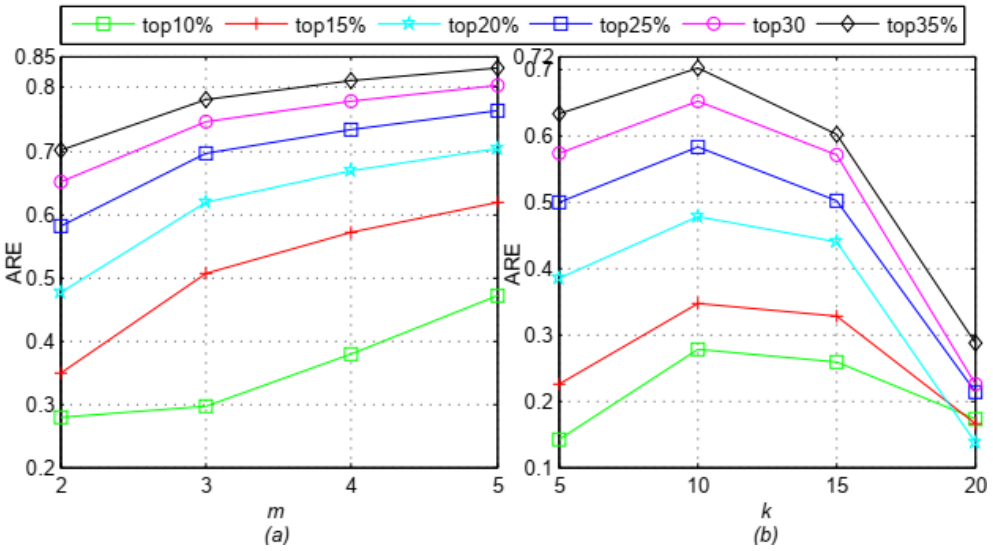

Figure 6. Effects of parameters k and m on data utility. (**a**) ARE evaluation versus m. (**b**) ARE evaluation versus k.

Figure 7 presents the results of ANR in terms of parameters k and m, and the frequent item domain size. On the same conditions in subgraphs of Figure 7, our scheme has higher proportion of itempairs than Disa, which implies better data quality of our scheme. In Figure 7a, the average proportion of preserved itempairs in anonymized clusters increases with the value of k, which indicates that the reduction extent of itempairs in the anonymized dataset is lower than the original dataset when there is an increase in the value of k. Thus, the number of items preserved in the record chunks of a cluster declines against k. Figure 7b shows that the average proportion of itempairs in the anonymized cluster decreases against parameter m. A larger value of m indicates that more problematic itemsets are generated, which means that more items are partitioned into different record chunks to meet the privacy requirement consequently. Figure 7c suggests that the ANRs are constant with the frequent item domain size. The reason is that all the items in record chunks of the anonymized clusters belong to the top 30% frequent items.

We test the ARE of our scheme and Disa by varying k, m, the number of frequent itempairs, and the number of frequent items. Figure 8 shows that our scheme has lower ARE than Disa. In other words, local suppression has smaller impact on co-appearance of items than global partition. Increasing k in Figure 8a leads to more minimal problematic items excluded to the item domain of frequent itemsets, which preserves more frequent itempairs. As a result, lower ARE is generated. In Figure 8b, we observe that the ARE improves with m. Increasing m causes more MPI so that more transactional records need to be anonymized, thus in higher ARE. Then, in Figure 8c, ARE increases gradually with the number of frequent itemsets to be queried. This is because, with the increase in the frequent itempair domain, more itempairs violating k^m-anonymity exist in the frequent itempairs, which decreases the accuracy of the instances of itempairs to be queried in the anonymized record chunks. Finally, the ARE, as presented in Figure 8d, is invariant with the frequent item domain size. All the items composing item domain of record chunks are the top 30% frequent items, so the number of all instances of itempairs is invariable.

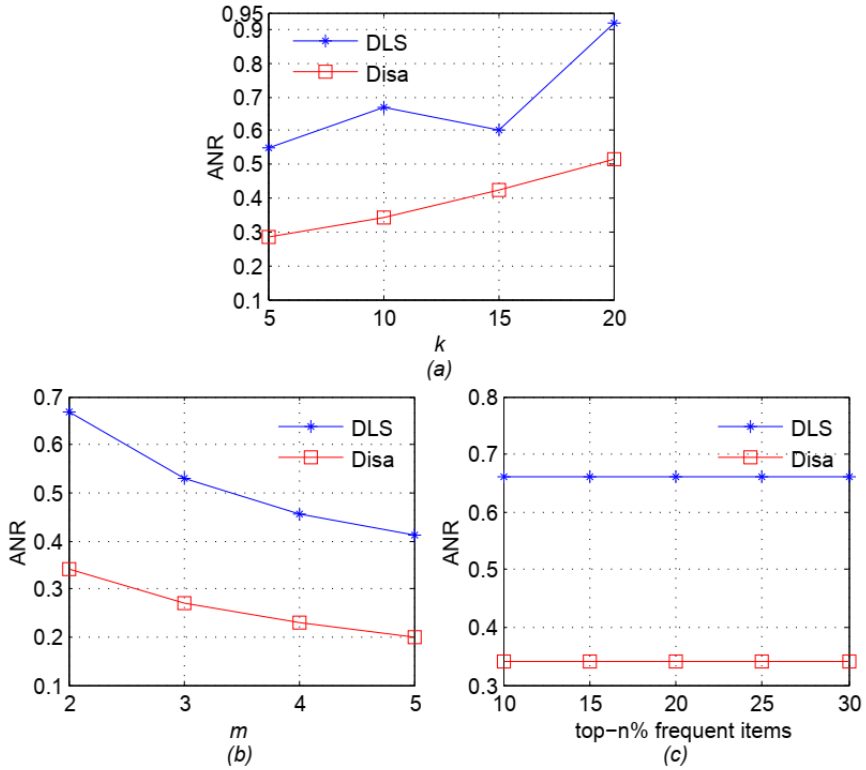

Figure 7. Comparison with Disa with respect to ANR. (**a**) Varying k. (**b**) Varying m. (**c**) Varying frequent item domain size.

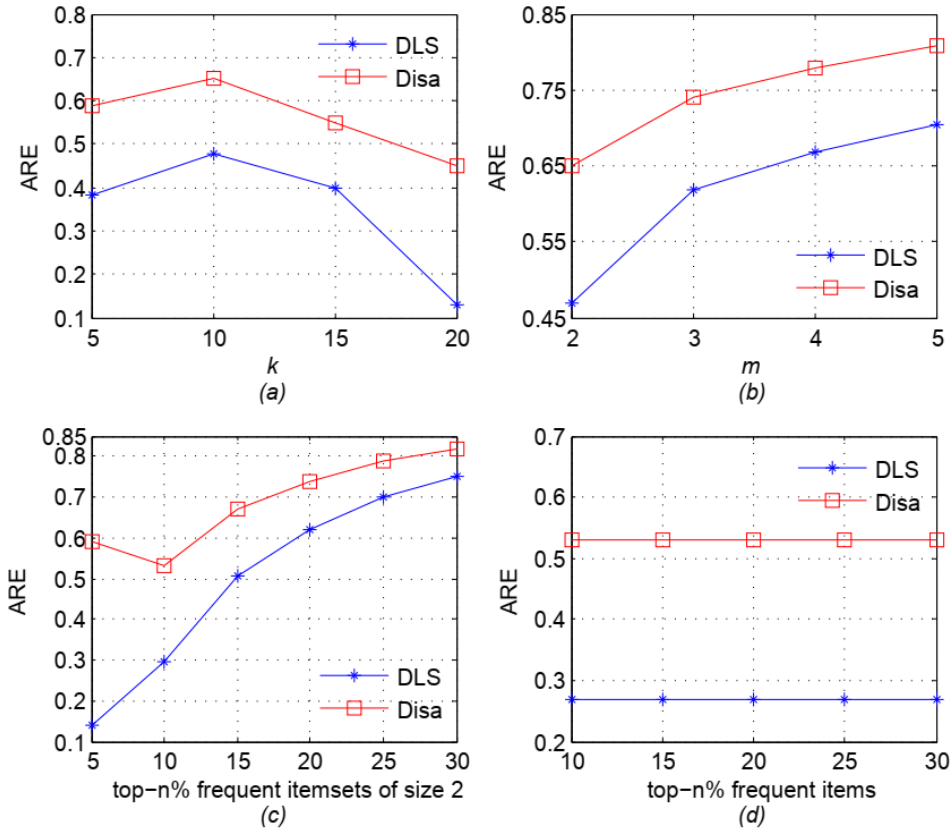

Figure 8. Comparison with Disa with respect to ARE. (**a**) Varying k. (**b**) Varying m. (**c**) Varying frequent itempair numbers. (**d**) Varying frequent item domain size.

6.4. Summary

We compare our improved scheme DLS with the disassociation method Disa in terms of the data utility of published data in the experiment. From experimental results, our scheme can retain more itempairs (measured by ANR) and itempair instances (measured by are) than Disa. Specifically, DLS preserves 1.8 times the number of itempairs as Disa on average in Figure 7. Figure 8 shows that the ARE scores of DLS are, on average, 31.04 percent better than Disa. This indicates that local suppression employed by DLS can retain items initially partitioned from the first record chunk in Disa at the cost of deleting some instances of items in the first record chunk. So, there are more itempairs and itempair instances in the anonymized record chunks. Our scheme reduces the support for itempairs but does not completely eliminate them.

7. Conclusions and Future Work

Disassociation overcomes the limitation of loading the whole dataset into a single task node to perform data anonymization while providing an efficient way for data anonymization in which multiple nodes undertake computing tasks in parallel. To preserve more valuable information, we improve the vertical partition algorithm of disassociation in this paper. We design and implement the related anonymization scheme DLS, which employs both local suppression and global partition to remove rare items or items that participate in rare combinations while preserving more combinations of original items to reduce the information loss of published datasets. The experimental results demonstrate that our scheme can improve the data quality of the anonymized transactional dataset.

This paper proposes an improved scheme to realize the vertical partition process for preserving more combinations of items as well as personal privacy. Our scheme can be applied to other scenarios for the protection of privacy information, such as personalized privacy preservation [51] and the scenario of releasing data with attributes of multiple types (e.g., the data contains both transactions and demographics) [52].

The proposed scheme is limited in the following aspects. First, it is hard to set the value of parameters k and m to achieve the "best" disassociated datasets, i.e., the "best" trade-off between privacy protection and data quality. Second, our scheme is based on disassociation, so the data quality of our disassociated dataset is also affected by the horizontal partition process of disassociation. The reason is that the horizontal partition clusters the records by using a naive similarity function without considering the associations of items in the dataset. Moreover, the disassociated dataset may be subject to privacy breaches if there is a cover problem in the record chunks. The cover problem occurs when there are one or more items in a record chunk where each of the records containing the items in the record chunk is identical with the domain of the record chunk.

In future work, we intend to employ utility constraints in the process of horizontal partition. The predefined utility constraint set contains specific itemsets satisfying intended analysis requirements, which can limit the amount of data disassociation. Moreover, the related algorithm needs to be considered as well. The other future research is evaluating privacy breaches in the disassociated dataset. We need to redefine the cover problem to evaluate the privacy breach in our anonymized datasets due to the difference between our DLS scheme and disassociation.

Author Contributions: Conceptualization, X.L.; methodology, X.L.; software, X.L.; validation, X.L.; formal analysis, X.L.; investigation, X.L.; resources, X.L.; data curation, X.L.; writing—original draft preparation, X.L.; writing—review and editing, X.L.; visualization, X.L.; supervision, X.L.; project administration, Y.Z.; funding acquisition, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61902157 and 62002139; in part by the Graduate Student Scientific Research Innovation Project of Jiangsu Province under Grant KYCX17_1809; in part by the China Postdoctoral Science Foundation under Grant 2019M661753 and 2019M651738; in part by Natural Science Foundation of Jiangsu Province under Grant BK20200886; and in part by Scientific Research Project of Jiangsu University for University Student under Grant 20AB0077.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Thakur, N.; Han, C.Y. An Ambient Intelligence-Based Human Behavior Monitoring Framework for Ubiquitous Environments. *Information* **2021**, *12*, 81. [CrossRef]
2. Narayanan, A.; Shmatikov, V. Robust de-anonymization of large sparse datasets. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP '08), Oakland, CA, USA, 18–21 May 2008; pp. 111–125.
3. Montjoye, Y.-A.; Radaelli, L.; Singh, V.K.; Pentland, A. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science* **2015**, *347*, 536–539. [CrossRef]
4. Gao, J.; Sun, L.; Cai, M. Quantifying privacy vulnerability of individual mobility traces: A case study of license plate recognition data. *Transp. Res. Pt. C-Emerg. Technol.* **2019**, *104*, 78–94. [CrossRef]
5. Terrovitis, M.; Liagouris, J.; Mamoulis, N.; Skiadopoulos, S. Privacy preservation by disassociation. In Proceedings of the VLDB Endowment, Istanbul, Turkey, 27–31 August 2012; pp. 944–955.
6. Terrovitis, M.; Mamoulis, N.; Kalnis, P. Privacy preserving anonymization of set-valued data. In Proceedings of the VLDB Endowment, Auckland, New Zealand, 24–30 August 2008; pp. 115–125.
7. Sweeney, L. *k*-Anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
8. Aggarwal, C.C. On *k*-anonymity and the curse of dimensionality. In Proceedings of the 31st International Conference on Very large Data Bases (VLDB'05), Trondheim, Norway, 30 August–2 September 2005; pp. 901–909.
9. He, Y.; Naughton, J. Anonymization of set-valued data via top-down, local generalization. In Proceedings of the VLDB Endowment, Lyon, France, 24–28 August 2009; pp. 934–945.
10. Soria-Comas, J.; Domingo-Ferrer, J.; Sánchez, D.; Martínez, S. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *VLDB J.* **2014**, *23*, 771–794. [CrossRef]
11. Awad, N.; Couchot, J.-F.; Bouna, B.A.; Philippe, L. Publishing anonymized set-valued data via disassociation towards analysis. *Future Internet* **2020**, *12*, 71. [CrossRef]
12. Wu, H.; Wang, L.; Xue, G. Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 589–602. [CrossRef]
13. Wu, H.; Wang, L.; Xue, G.; Tang, J.; Yang, D. Enabling data trustworthiness and user privacy in mobile crowdsensing. *IEEE/ACM Trans. Netw.* **2019**, *27*, 2294–2307. [CrossRef]
14. Feng, X.; Wang, L. PAU: Privacy assessment method with uncertainty consideration for cloud-based vehicular networks. *Future Gener. Comput. Syst.* **2019**, *96*, 368–375. [CrossRef]
15. Cui, J.; Wen, J.; Han, S.; Zhong, H. Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network. *IEEE Internet Things J.* **2018**, *5*, 3491–3498. [CrossRef]
16. Liu, Q.; Wang, G.; Li, F.; Yang, S.; Wu, J. Preserving privacy with probabilistic indistinguishability in weighted social networks. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 1417–1429. [CrossRef]
17. Jiang, L.; Shi, L.; Liu, L.; Yao, J.; Yuan, B.; Zheng, Y. An efficient evolutionary user interest community discovery model in dynamic social networks for Internet of people. *IEEE Internet Things J.* **2019**, *6*, 9226–9236. [CrossRef]
18. Xu, Y.; Wang, K.; Fu, A.W.; Yu, P.S. Anonymizing transaction databases for publication. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08), Las Vegas, NV, USA, 24–27 August 2008; pp. 767–775.
19. Xue, M.; Karras, P.; Raïssi, C.; Vaidya, J.; Tan, K.-L. Anonymizing set-valued data by nonreciprocal recoding. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12), Beijing, China, 12–16 August 2012; pp. 1050–1058.
20. Lin, C.; Liu, Q.; Fournier-Viger, P.; Hong, T.-P. PTA: An efficient system for transaction database anonymization. *IEEE Access* **2016**, *4*, 6467–6479. [CrossRef]
21. Puri, V.; Sachdeva, S.; Kaur, P. Privacy preserving publication of relational and transaction data: Survey on the anonymization of patient data. *Comput. Sci. Rev.* **2019**, *32*, 45–61. [CrossRef]

22. Terrovitis, M.; Mamoulis, N.; Kalnis, P. Local and global recoding methods for anonymizing set-valued data. *VLDB J.* **2011**, *20*, 83–106. [CrossRef]
23. Loukides, G.; Gkoulalas-Divanis, A.; Malin, B. COAT: Constraint-based anonymization of transactions. *Knowl. Inf. Syst.* **2011**, *28*, 251–282. [CrossRef]
24. Loukides, G.; Gkoulalas-Divanis, A. Utility-aware anonymization of diagnosis codes. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 60–70. [CrossRef]
25. Loukides, G.; Liagouris, J.; Gkoulalas-Divanis, A.; Terrovitis, M. Disassociation for electronic health record privacy. *J. Biomed. Inform.* **2014**, *50*, 46–61. [CrossRef]
26. Cao, J.; Karras, P.; Raïssi, C.; Tan, K.-L. ρ-uncertainty: Inference-proof transaction anonymization. In Proceedings of the VLDB Endowment, Singapore, 13–17 September 2010; pp. 1033–1044. [CrossRef]
27. Jia, X.; Pan, C.; Xu, X.; Zhu, K.Q.; Lo, E. ρ-uncertainty anonymization by partial suppression. In Proceedings of the International Conference on Database Systems for Advanced Applications, Bali, Indonesia, 21–24 April 2014; pp. 188–202.
28. Nakagawa, T.; Arai, H.; Nakagawa, H. Personalized anonymization for set-valued data by partial suppression. *Trans. Data Priv.* **2018**, *11*, 219–237.
29. Loukides, G.; Gkoulalas-Divanis, A.; Shao, J. Anonymizing transaction data to eliminate sensitive inferences. In Proceedings of the 21st International Conference on Database and Expert Systems Applications: Part I (DEXA'10), Bilbao, Spain, 30 August–3 September 2010; pp. 400–415.
30. Loukides, G.; Gkoulalas-Divanis, A.; Shao, J. Efficient and flexible anonymization of transaction data. *Knowl. Inf. Syst.* **2013**, *36*, 153–210. [CrossRef]
31. Loukides, G.; Gkoulalas-Divanis, A.; Malin, B. Anonymization of electronic medical records for validating genome-wide association studies. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 7898–7903. [CrossRef]
32. Yao, L.; Chen, Z.; Wang, X.; Liu, D.; Wu, G. Sensitive label privacy preservation with anatomization for data publishing. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 904–917. [CrossRef]
33. Xiao, X.; Tao, Y. Anatomy: Simple and effective privacy preservation. In Proceedings of the VLDB Endowment, Seoul, Korea, 12–15 September 2006; pp. 139–150.
34. Wang, K.; Wang, P.; Fu, A.; Wong, R.C.-W. Generalized bucketization scheme for flexible privacy settings. *Inf. Sci.* **2016**, *348*, 377–393. [CrossRef]
35. Yang, X.; Wang, Y.; Wang, B. Privacy preserving approaches for multiple sensitive attributes in data publishing. *Chin. J. Comput.* **2008**, *31*, 574–587. [CrossRef]
36. Li, T.; Li, N.; Zhang, J.; Molloy, I. Slicing: A new approach to privacy preserving data publishing. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 561–574. [CrossRef]
37. Wang, M.; Jiang, Z.; Zhang, Y. *T*-closeness slicing: A new privacy preserving approach for transactional data publishing. *INFORMS J. Comput.* **2018**, *30*, 438–453. [CrossRef]
38. Susan, V.S.; Christopher, T. Anatomisation with slicing: A new privacy preservation approach for multiple sensitive attributes. *SpringerPlus* **2016**, *5*, 964. [CrossRef]
39. Loukides, G.; Liagouris, J.; Gkoulalas-Divanis, A.; Terrovitis, M. Utility-constrained electronic health record data publishing through generalization and disassociation. In *Medical Data Privacy Handbook*; Gkoulalas-Divanis, A., Loukides, G., Eds.; Springer: Cham, Switzerland, 2015; pp. 149–177.
40. Mohammed, N.; Fung, B.C.M.; Debbabi, M. *Preserving Privacy and Utility in RFID Data Publishing*; Technical Report 6850; Concordia University: Montreal, QC, Canada, 2010.
41. Al-Hussaeni, K.; Fung, B.C.M.; Cheung, W.K. Privacy-preserving trajectory stream publishing. *Data Knowl. Eng.* **2014**, *94*, 89–109. [CrossRef]
42. Terrovitis, M.; Poulis, G.; Mamoulis, N.; Skiadopoulos, S. Local suppression and splitting techniques for privacy preserving publication of trajectories. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1466–1479. [CrossRef]
43. Chen, R.; Fung, B.C.M.; Mohammed, N.; Desai, B.C.; Wang, K. Privacy-preserving trajectory data publishing by local suppression. *Inf. Sci.* **2013**, *231*, 83–97. [CrossRef]
44. Komishani, E.G.; Abadi, M.; Deldar, F. PPTD: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression. *Knowl.-Based Syst.* **2016**, *94*, 43–59. [CrossRef]
45. Sara, B.; Al Bouna, B.; Mohamed, N.; Christophe, G. On the evaluation of the privacy breach in disassociated set-valued datasets. In Proceedings of the 13th International Joint Conference on e-Business and Telecommunications, Lisbon, Portugal, 26–28 July 2016; pp. 318–326.
46. Awad, N.; Al Bouna, B.; Couchot, J.F.; Philippe, L. Safe disassociation of set-valued datasets. *J. Intell. Inf. Syst.* **2019**, *53*, 547–562. [CrossRef]
47. Puri, V.; Kaur, P.; Sachdeva, S. Effective removal of privacy breaches in disassociated transactional datasets. *Arab. J. Sci. Eng.* **2020**, *45*, 3257–3272. [CrossRef]
48. Awad, N.; Couchot, J.F.; Al Bouna, B.; Philippe, L. Ant-Driven Clustering for Utility-Aware Disassociation of Set-Valued Datasets. In Proceedings of the 23rd International Database Applications and Engineering Symposium, Athens, Greece, 10–12 June 2019; pp. 1–9.

49. Bewong, M.; Liu, J.; Liu, L.; Li, J.; Choo1, K.-K.R. A relative privacy model for effective privacy preservation in transactional data. *Concurr. Comput.-Pract. Exp.* **2019**, *31*, e4923. [CrossRef]
50. Zheng, Z.; Kohavi, R.; Mason, L. Real world performance of association rule algorithms. In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01), San Francisco, CA, USA, 26–29 August 2001; pp. 401–406.
51. Xiao, X.; Tao, Y. Personalized privacy preservation. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Chicago, IL, USA, 27–29 June 2006; pp. 229–240.
52. Wang, L.-E.; Li, X. A graph-based multifold model for anonymizing data with attributes of multiple types. *Comput. Secur.* **2017**, *72*, 122–135. [CrossRef]