Preserving Privacy and Utility in RFID Data Publishing

Noman Mohammed, Benjamin C. M. Fung, and Mourad Debbabi

Abstract—Radio Frequency IDentification (RFID) is a technology that helps machines identify objects remotely. The RFID technology has been extensively used in many domains, such as mass transportation and healthcare management systems. The collected RFID data capture the detailed movement information of the tagged objects, offering tremendous opportunities for mining useful knowledge. Yet, publishing the *raw* RFID data for data mining would reveal the specific locations, time, and some other potentially sensitive information of the tagged objects or individuals. In this paper, we study the privacy threats in RFID data publishing and show that traditional anonymization methods are not applicable for RFID data due to its challenging properties: high-dimensional, sparse, and sequential. Our primary contributions are (1) to adopt a new privacy model called LKC-privacy that overcomes these challenges, and (2) to develop an efficient anonymization algorithm to achieve LKC-privacy while preserving the information utility for data mining.

Index Terms—Privacy, security, data mining

1 Introduction

Radio Frequency IDentification (RFID) is a technology for objects automatic identification. Figure 1 depicts an overview of a RFID system, which typically consists of a large number of tags and readers, and a database server. A tag is a small device attached to a moving object. A reader broadcasts a radio signal to the tag, which then transmits its unique identifier back to the reader. Streams of RFID data entries, in the format of (ID, loc, t), are then stored in a RFID database, where ID is the unique identifer of a RFID tag, loc is the location of the reader, and t is the time of detection. The database system answers queries by joining the trajectory data with some object-specific data that describes the object.

Initial applications of RFID focus on tracking items in supply-chain management or baggage in airports. Recently, it has been used to track individuals. Publication of these RFID data threatens individuals' privacy since these raw data provide location information that identifies individuals and, potentially, their sensitive information. Below, we present some real-life applications of publishing RFID data.

Transit company: Transit companies have started to use contactless smart cards or RFID cards, such as the Octopus card in Hong Kong, the OPUS card in Montreal, and the Oyster Travel card in London. In some transit systems, passengers register personal information when they first purchase their cards, so that appropriate fare is charged based on their status. The personal journey data together with the passengers' personal information provide a valuable source of information for data mining with the goal of improving the transportation services. For example, the IT department of STM (Transit Company of Montreal) owns the journey data. They want to share the data internally with their marketing department and externally with other transit companies for analysis purposes.

Hospital: Some hospitals have adopted RFID sensory system to track the positions of patients, doctors, and medical equipment inside the hospital with the goals of minimizing life-threatening medical errors and improving the management of patients and resources [35], [19]. Analyzing RFID data, however, is a non-trivial task. Hospitals often do not have the expertise to perform the analysis themselves but outsource this process and, therefore, require granting a third party access to the patient-specific location and health data.

Most previous work on privacy-preserving RFID technology addressed the threats caused by the physical RFID tags in the *data collection* phase [19], [30]. These techniques do not address the privacy threats in the *data publishing* phase, when a large volume of RFID data is released to a third party for data mining.

In this paper, we study the privacy threats in the data publishing phase and define a practical privacy model to accommodate the special challenges of RFID data. We propose an anonymization algorithm (the data anonymizer in Figure 1) to transform the underlying raw RFID data into a version that is immunized against privacy attacks but still supports effective data mining. Data "publishing" includes sharing the data with specific recipients and releasing the data for public download; the recipient could be an ordinary user who wants to perform legitimate data analysis, or could potentially be an adversary who attempts to associate sensitive information in the published data with a target victim.

- *N. Mohammed, B. C. M. Fung and M. Debbabi are with the Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada.*
E-mail: {no_moham, fung, debbabi}@ciise.concordia.ca

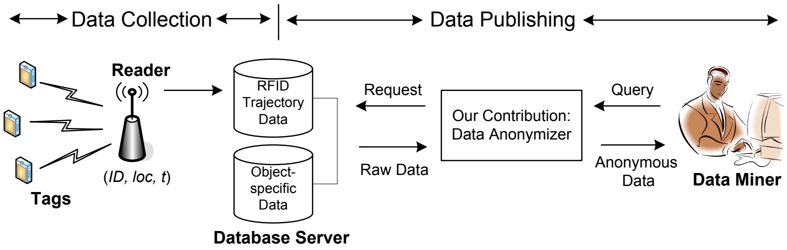


Fig. 1. Data flow in RFID system

TABLE 1
Raw passenger-specific trajectory data

ID	Trajectory	Status	...
1	$\langle b2 \rightarrow d3 \rightarrow c4 \rightarrow f6 \rightarrow c7\rangle$	On-welfare	...
2	$\langle f6 \rightarrow c7 \rightarrow e8\rangle$	Student	...
3	$\langle d3 \rightarrow c4 \rightarrow f6 \rightarrow e8\rangle$	Retired	...
4	$\langle b2 \rightarrow c5 \rightarrow c7 \rightarrow e8\rangle$	Student	...
5	$\langle d3 \rightarrow c7 \rightarrow e8\rangle$	Retired	...
6	$\langle c5 \rightarrow f6 \rightarrow e8\rangle$	Full-time	...
7	$\langle b2 \rightarrow f6 \rightarrow c7 \rightarrow e8\rangle$	Full-time	...
8	$\langle b2 \rightarrow c5 \rightarrow f6 \rightarrow c7\rangle$	On-welfare	...

1.1 Motivating Example

We illustrate the privacy threats of publishing raw RFID data by an example.

Example 1: A transit company wants to release the passenger-specific trajectory data (Table 1) to a data miner for research purposes. Each record contains a *trajectory* and some passenger-specific information, where the *trajectory* is a sequence of *pairs* $(loc_i t_i)$ indicating the passenger's visited location loc_i at time t_i. For example, $ID\#2$ has a trajectory $\langle f6 \rightarrow c7 \rightarrow e8\rangle$, meaning that the passenger has visited locations f, c, and e at time 6, 7, and 8, respectively. Without loss of generality, we assume that each record contains only one sensitive attribute, namely, *status*, in this example. We address two types of privacy threats:

Identity linkage: If a trajectory in the table is so specific that not many passengers match it, releasing the data may lead to linking the victim's record and, therefore, her status. Suppose the adversary knows that the data record of a target victim, Alice, is in Table 1, and Alice has visited $b2$ and $d3$. Alice's record, together with her sensitive value (On-welfare in this case), can be uniquely identified because $ID\#1$ is the *only* record that contains $b2$ and $d3$. Besides, the adversary can also determine the other visited locations of Alice, such as $c4$, $f6$, and $c7$.

Attribute linkage: If a sensitive value occurs frequently together with some sequence of pairs, then the sensitive information can be inferred from such sequence even though the exact record of the victim cannot be identified. Suppose the adversary knows that Bob has visited $b2$ and $f6$. Since two out of the three records ($ID\#$1,7,8) containing $b2$ and $f6$ have sensitive value *On-welfare*, the adversary can infer that Bob is on welfare with $2/3 = 67\%$ confidence. ■

Many privacy models, such as K-anonymity [29][31] and its extensions [21][25][36][37], have been proposed to thwart privacy threats caused by identity and attribute linkages in the context of relational databases. These models are based on the notion of *quasi-identifier* (*QID*), which is a set of attributes that may be used for linkages. The basic idea is to disorient potential linkages by generalizing the records into equivalent groups that share the same values on QID. These privacy models are effective for anonymizing relational data, but they are not applicable to RFID data due to the following challenges.

(1) High dimensionality: Consider a transit system having 50 stations that operate 24 hours per day. There are $50 \times 24 = 1200$ possible combinations (dimensions) of locations and timestamps. Each dimension could be a potential QID attribute used for identity and attribute linkages. Traditional K-anonymity would require every trajectory to be shared by at least K records. Due to *the curse of high dimensionality* [2], most of the data have to be suppressed in order to achieve K-anonymity. For example, to achieve 2-anonymity on the trajectory data in Table 1, all instances of $\{b2, d3, c4, c5\}$ have to be suppressed even though K is small.

(2) Data sparseness: Consider passengers in a public transit system or patients in a hospital. They usually visit only a few locations compared to all available locations, so each trajectory is relatively short. Anonymizing these short, little-overlapping trajectories in a high-dimensional space poses a significant challenge for traditional anonymization techniques because it is difficult to identify and group the trajectories together. Enforcing traditional K-anonymity on high-dimensional and sparse data would render the data useless.

(3) Sequential: Time is an essential factor of RFID data, which may incur unique privacy threats. Consider two trajectories $b3 \rightarrow e6$ and $e3 \rightarrow b6$. Both the trajectories have same locations but different timestamps; and thus, they are different from each other. Furthermore, the same location when associated with different timestamps should be considered different in the context of RFID data. For example, $b2 \rightarrow e8$ and $b3 \rightarrow e6$ are different due to different timestamps. These differences may provide an adversary more opportunities to succeed in a privacy attack, and therefore require more efforts in the anonymization algorithm.

TABLE 2
Anonymous data with $L = 2$, $K = 2$, $C = 50\%$

ID	Trajectory	Status	...
1	$\langle d3 \to f6 \to c7 \rangle$	On-welfare	...
2	$\langle f6 \to c7 \to e8 \rangle$	Student	...
3	$\langle d3 \to f6 \to e8 \rangle$	Retired	...
4	$\langle c5 \to c7 \to e8 \rangle$	Student	...
5	$\langle d3 \to c7 \to e8 \rangle$	Retired	...
6	$\langle c5 \to f6 \to e8 \rangle$	Full-time	...
7	$\langle f6 \to c7 \to e8 \rangle$	Full-time	...
8	$\langle c5 \to f6 \to c7 \rangle$	On-welfare	...

1.2 Privacy and Utility

Traditional K-anonymity and its extended privacy models assume that an adversary could potentially use any or even all of the QID attributes as background knowledge to perform identity or attribute linkages. However, in real-life privacy attacks, it is very difficult for an adversary to acquire *all* the visited locations and timestamps of a victim because it requires non-trivial effort to gather each piece of background knowledge from so many possible locations at different times. Thus, it is reasonable to assume that the adversary's background knowledge is bounded by at most L pairs of $(loc_i t_i)$ that the victim has visited.

Based on this reasonable assumption, we adopt a new privacy model called *LKC-privacy* [27] for anonymizing high-dimensional and sparse RFID trajectory data. The general idea of LKC-privacy has been previously applied on relational data [27], but this paper modifies the model to address the high-dimensional, sparse, and sequential RFID data. Although the intuition of LKC-privacy is applied here, the privacy model, the algorithm, and the data structures are completely different. The general intuition is to ensure that every sequence q with maximum length L of any trajectory in a data table T is shared by at least K records in T, and the confidence of inferring any sensitive value in S from q is not greater than C, where L and K are positive integer thresholds, C is a positive real number threshold, and S is a set of sensitive values specified by the data holder. LKC-privacy bounds the probability of a successful identity linkage to be $\leq 1/K$ and the probability of a successful attribute linkage to be $\leq C$. Table 2 shows an example of an anonymous table that satisfies $(2, 2, 50\%)$-privacy by suppressing $b2$ and $c4$ from Table 1. Every possible sequence q with maximum length 2 in Table 2 is shared by at least 2 records and the confidence of inferring the sensitive value *On-welfare* from q is not greater than 50%.

While protecting privacy is a critical element in data publishing, it is equally important to preserve the utility of the published data because this is the primary reason for publication. In this paper, we aim at preserving the *maximal frequent sequences* (*MFS*) because MFS often serves as the information basis for different primitive data mining tasks on trajectory data. MFS represents the set of longest sequences of visited locations by some minimum number of moving objects within a particular time interval. In the context of RFID data, frequent sequences can capture the major trajectories of moving objects [4]. MFS is also useful for trajectory pattern mining [15] and workflow mining [16].

One frequently raised question is: Given that the frequent sequence mining task is known in advance, why not publish the frequent sequences instead of the data records? The goal is to allow data sharing for frequent sequence mining in the presence of privacy concern. This problem is very different from secure multiparty computation [23], which allows "result sharing"(e.g., the frequent sequences in our case) but completely prohibits data sharing. In many applications, data sharing gives greater flexibility than result sharing because data recipients can perform their required analysis and data exploration, such as, mine patterns in a specific group of records, and try different modeling methods and parameters.

1.3 Contributions

Our contributions can be summarized as follows. First, based on the practical assumption that adversary has limited knowledge, we adopt LKC-privacy model to address the special challenges of anonymizing high-dimensional, sparse, and sequential RFID data. We further show that LKC-privacy is a generalized model of K-anonymity [29][31], confidence bounding [34], and (α, k)-anonymity [36] (Section 2). Second, we present an efficient anonymization algorithm to achieve LKC-privacy while preserving maximal frequent sequences in the anonymous RFID data (Section 3). Finally, extensive experimental results support that our proposed privacy model and anonymization method outperform the traditional approaches in terms of data quality, efficiency, and scalability (Section 4). To the best of our knowledge, this is the first work addressing the anonymization problem for RFID data and preserving maximal frequent sequences for data mining.

2 Problem Definition

We first describe the format of RFID data and then formally define the problem based on the privacy and utility requirements.

2.1 RFID Data Table

RFID data is generated in the form of (ID, loc, t), where ID is the unique identifier of a tag, loc is the location of the reader that reads the tag, and t is the time of reading. We assume that the RFID tags are carried by or attached to some moving persons or objects, such as patients in the hospitals or passengers in the public transit systems. A reader reads a tag either continuously or in a fix interval basis. Thus, the database may have duplicate entries showing the same location if the object has not moved. Gonzalez et al. [16] suggested some preprocessing methods to compress RFID data.

A *pair* $(loc_i t_i)$ represents the visited location loc_i of an object at time t_i. The *trajectory* of an object, denoted by $\langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n) \rangle$, is a sequence of pairs that can be obtained by first grouping the RFID entries by ID and then sorting the entries in each group by their timestamps. A timestamp is the entry time to a location, so the object is assumed to be staying in the same location until it has been detected again. An object may revisit the same locations at different time. At any time, an object can appear at only one location, so $\langle a1 \rightarrow b1 \rangle$ is not a valid sequence and timestamps in a trajectory increase monotonically.

A RFID data table T is a collection of records in the form $\langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n) \rangle : s_1, \ldots, s_p : d_1, \ldots, d_m$, where $\langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n) \rangle$ is the trajectory, $s_i \in S_i$ are the sensitive attributes, and $d_i \in D_i$ are the quasi-identifying attributes (QID) of an object. The sensitive and QID attributes are the object-specific data in the form of relational data. Identity and attribute linkages via the QID attributes can be avoided by applying existing anonymization methods for relational data [12][20][22][25][34]. In this paper, we focus on eliminating identity and attribute linkages via the RFID trajectory data as illustrated in Example 1.

2.2 Privacy Model

Suppose a data holder wants to publish a RFID data table T (e.g., Table 1) to some recipient(s) for data mining. Explicit identifiers, e.g., name, SSN, and ID, are removed. Note, we keep the ID in our examples for discussion purpose only. The trajectory, the object-specific QID, and sensitive attributes are assumed to be important for the data mining task; otherwise, they should be removed.

One recipient, who is an adversary, seeks to identify the record or sensitive values of some target victim V in T. As explained earlier, we assume that the adversary knows at most L pairs of location and timestamp that V has previously visited. We use q to denote such prior known sequence of pairs, where $|q| \leq L$. Based on the prior knowledge q, the adversary could identify a group of records, denoted by $T(q)$, that "contains" q. A record in T *contains* q if q is a subsequence of the trajectory in the record. For example in Table 1, records with $ID\#1, 2, 7, 8$ contain $q = \langle f6 \rightarrow c7 \rangle$, written as $T(q) = \{ID\#1, 2, 7, 8\}$. The prior knowledge q, may consist of any L pairs, not necessarily consecutive, such as $q = \langle b2 \rightarrow c7 \rangle$. Based on $T(q)$, the adversary could launch two types of privacy attacks:

- *Identity linkage*: Given prior knowledge q, $T(q)$ is a set of candidate records that contains the victim V's record. If the group size of $T(q)$, denoted by $|T(q)|$, is small, then the adversary may identify V's record from $T(q)$ and, therefore, V's sensitive value. For example, if $q = \langle b2 \rightarrow d3 \rangle$ in Table 1, $T(q) = \{ID\#1\}$. Thus, the adversary can easily infer that V's sensitive value is *On-welfare*.
- *Attribute linkage*: Given prior knowledge q, the adversary can identify $T(q)$ and infer that V has sensitive value s with confidence $P(s|q) = \frac{|T(q \wedge s)|}{|T(q)|}$, where $T(q \wedge s)$ denotes the set of records containing both q and s. $P(s|q)$ is the percentage of the records in $T(q)$ containing s. The privacy of V is at risk if $P(s|q)$ is high. For example, given $q = \langle b2 \rightarrow f6 \rangle$ in Table 1, $T(q \wedge \textit{On-welfare}) = \{ID\#1, 8\}$ and $T(q) = \{ID\#1, 7, 8\}$; therefore, $P(\textit{On-welfare}|q) = 2/3 = 67\%$.

To thwart the identity and attribute linkages, we require that every sequence with a maximum length L in the RFID trajectory data has to be shared by at least a certain number of records, and the ratio of sensitive value(s) in every group cannot be too high. *LKC-privacy* reflects this intuition.

Definition 1 (LKC-privacy): Let L be the maximum length of the prior knowledge. Let S be a set of sensitive values. A RFID data table T satisfies *LKC-privacy* if and only if for any sequence q with $|q| \leq L$ of any trajectory in T,

1) $|T(q)| \geq K$, where $K > 0$ is an integer anonymity threshold, and
2) $P(s|q) \leq C$ for any $s \in S$, where $0 < C \leq 1$ is a real number confidence threshold. ■

The data holder specifies the thresholds L, K, and C. The maximum length L reflects the assumption of the adversary's power. LKC-privacy guarantees that the probability of a successful identity linkage to be $\leq 1/K$ and the probability of a successful attribute linkage to be $\leq C$. LKC-privacy has several nice properties that make it suitable for anonymizing high-dimensional sparse RFID data. First, it only requires subsequences of a trajectory to be shared by at least K records. This is a major relaxation from traditional K-anonymity based on a very reasonable assumption that the adversary has limited power. Second, LKC-privacy generalizes several traditional privacy models. K-anonymity [29][31] is a special case of LKC-privacy with $C = 100\%$ and $L = |d|$, where $|d|$ is the number of dimensions, i.e., number of distinct pairs, in the RFID data table. Confidence bounding [34] is a special case of LKC-privacy with $K = 1$ and $L = |d|$. (α, k)-anonymity [36] is also a special case of LKC-privacy with $L = |d|$, $K = k$, and $C = \alpha$. Thus, the data holder can still achieve the traditional models, if needed. Third, it is flexible to adjust the trade-off between data privacy and data utility, and between an adversary's power and data utility. Increasing L and K, or decreasing C, would improve the privacy at the expense of data utility. Finally, LKC-privacy is a general privacy model that thwarts both identity linkage and attribute linkage, i.e., the privacy model is applicable to anonymize RFID data with or without sensitive attributes.

2.3 Utility Measure

The measure of data utility varies depending on the data mining task to be performed on the published data. In this paper, we aim at preserving the maximal frequent sequences. A sequence $q = \langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n) \rangle$ is an ordered set of locations. A sequence q is *frequent* in a RFID data table T if $|T(q)| \geq K'$, where $T(q)$ is the set of records containing q and K' is a minimum support threshold. Frequent sequences (FS) capture the major trajectories of the moving objects [4], and often form the information basis for different primitive data mining tasks on sequential data, e.g., association rules mining [3]. In the context of RFID data, association rules can be used to determine the subsequent locations of the moving object given the previously visited locations. This knowledge is important for workflow mining [16].

There is no doubt that FS are useful. Yet, mining all FS is a computationally expensive operation. When the data volume is large and FS are long, it is infeasible to identify all FS because all subsequences of an FS are also frequent. Since RFID data is high-dimensional and in large volume, a more feasible solution is to preserve only the *maximal frequent sequences* (*MFS*).

Definition 2 (Maximal frequent sequence): For a given minimum support threshold $K' > 0$, a sequence x is *maximal frequent* in a RFID data table T if x is frequent and no super sequence of x is frequent. ■

The set of MFS in T, denoted by $U(T)$, is much smaller than the set of FS in T given the same K'. MFS still contains the essential information for different kinds of data analysis [24]. For example, MFS captures the longest frequently visited trajectories. Any subsequence of an MFS is also a FS. Once all the MFS have been determined, the support counts of any particular FS can be computed by scanning the database once. Our data utility goal is to preserve as many MFS as possible, i.e., maximize $|U(T)|$, in the anonymous RFID data table.

2.4 Problem Statement

LKC-privacy can be achieved by performing a sequence of *generalization* and/or *suppression* operations on the RFID data table. Generalization replaces a specific value with a more general value for a given attribute according to a taxonomy tree. For example, location a can be generalized into a broader location ab according to the taxonomy tree in Figure 2. Similarly, ab can be further generalized into $abcd$. The same generalization can be performed on the time dimension. Suppression removes a pair from one or more trajectories in the RFID data table T. For example, Table 2 is the result of suppressing $b2$ and $c4$ from Table 1. In both the above schemes, if all the instances of a value are generalized or suppressed, then it is called *global recoding*. In contrast, if some instances of a value remain unchanged while other instances are generalized or suppressed, then it is called *local recoding*. Refer to [20] for detailed descriptions on different global and local recoding schemes.

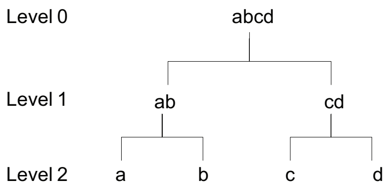


Fig. 2. Taxonomy tree on location

In this paper, we employ *global suppression*, meaning that if a pair p is chosen to be suppressed, *all* instances of p in T are suppressed. Global suppression offers several advantages over generalization and local suppression. First, suppression does not require a predefined taxonomy tree for generalization, which often is unavailable in real-life databases. Second, RFID data could be extremely sparse. Enforcing global generalization on RFID data will result in generalizing many sibling location or time values even if there is only a small number of outlier pairs, such as $c4$ in Table 1. Suppression offers the flexibility of removing those outliers without affecting the rest of the data. Note, we do not intend to claim that global suppression is always better than other schemes. For example, LeFevre et al. [20] present some local generalization schemes that may result in less data loss depending on the utility measure. Third, global suppression retains exactly the same support counts of the preserved MFS in the anonymous RFID data table as there were in the raw data. In contrast, a local suppression scheme may delete *some* instances of the chosen pair and, therefore, change the support counts of the preserved MFS. For example, if the support count of a sequence $\langle (loc_1 t_1) \rangle$ is 20 and the support count of its super sequence $\langle (loc_1 t_1) \rightarrow (loc_2 t_2) \rangle$ is 10, then the confidence of inferring the occurrence of $(loc_2 t_2)$ from $(loc_1 t_1)$ is 10/20 = 50%. Now, suppose we suppress only 10 instances of $(loc_1 t_1)$ from T. The support of $\langle (loc_1 t_1) \rightarrow (loc_2 t_2) \rangle$ will vary from 0 to 10 and the confidence of inferring the occurrence of $(loc_2 t_2)$ from $(loc_1 t_1)$ will vary from 0% to 100% depending on which instances have been suppressed. Hence, employing local suppression cannot preserve the truthful support counts of the preserved frequent sequences, implying that the derived knowledge, such as association rules, is not truthful, too.

Definition 3 (Anonymity for MFS in RFID): Given a RFID data table T, a LKC-privacy requirement, a minimum support threshold K', a set of sensitive values S, the problem of *anonymity for MFS in RFID* is to identify a transformed version of T that satisfies the LKC-privacy requirement while preserving the maximum number of MFS. ■

3 The Anonymization Algorithm

Given a RFID data table T, our first step is to identify all sequences that violate the given LKC-privacy requirement. Section 3.1 describes a method to identify violating sequences efficiently. Section 3.2 presents a greedy algorithm to eliminate the violating sequences with the goal of preserving as many maximal frequent sequences as possible.

3.1 Identifying Violating Sequences

An adversary may use any sequence with length not greater than L as background knowledge to launch a linkage attack. Thus, any non-empty sequence q with $|q| \leq L$ in T is a *violating sequence* if its group $T(q)$ does not satisfy condition 1, condition 2, or both in LKC-privacy in Definition 1.

Definition 4 (Violating sequence): Let q be a sequence of a trajectory in T with $|q| \leq L$. q is a *violating sequence* with respect to a LKC-privacy requirement if (1) q is non-empty, and (2) $|T(q)| < K$ or $P(s|q) > C$ for any sensitive value $s \in S$. ■

Example 2: Let $L = 2$, $K = 2$, $C = 50\%$, and $S = \{\textit{On-welfare}\}$. In Table 1, a sequence $q_1 = \langle b2 \rightarrow c4\rangle$ is a violating sequence because $|T(q_1)| = 1 < K$. A sequence $q_2 = \langle b2 \rightarrow f6\rangle$ is a violating sequence because $P(\textit{On-welfare}|q_2) = 67\% > C$. However, a sequence $q_3 = \langle b2 \rightarrow c5 \rightarrow f6 \rightarrow c7\rangle$ is not a violating sequence even if $|T(q_3)| = 1 < K$ and $P(\textit{On-welfare}|q_3) = 67\% > C$ because $|q_3| > L$. ■

A RFID data table satisfies a given LKC-privacy requirement, if all violating sequences with respect to the privacy requirement are removed, because all possible channels for identity and attribute linkages are eliminated. A naive approach is to first enumerate all possible violating sequences and then remove them. This approach is infeasible because of the huge number of violating sequences. Consider a violating sequence q with $|T(q)| < K$. Any super sequence of q with length less than or equal to L, denoted by q'', in the database T is also a violating sequence because $|T(q'')| \leq |T(q)| < K$.

One *incorrect* approach to achieve LKC-privacy is to ignore the sequences with size less than L and assume that if a table T satisfies LKC-privacy, then T satisfies $L'KC$-privacy where $L' < L$. Unfortunately, this monotonic property with respect to L does not hold in LKC-privacy.

Theorem 1: LKC-privacy is not monotonic with respect to adversary's knowledge L.

Proof. To prove that LKC-privacy is not monotonic with respect to L, it is sufficient to prove that one of the conditions of LKC-privacy in Definition 1 is not monotonic. Following we provide a counter example for both the conditions.

Condition 1: Anonymity threshold K is not monotonic with respect to L. If all the size-L sequences are non-violating, it does not guarantee that a sequence with size $L' \leq L$ is also non-violating. In Table 3, though the size-3 sequences satisfy privacy requirement for $K = 2$, the size-2 sequence, $q = \langle a1 \rightarrow d2\rangle$ does not satisfy the requirement.

Condition 2: Confidence threshold C is not monotonic with respect to L. If q is a non-violating sequence with $P(s|q) \leq C$ and $|T(q)| \geq K$, its subsequence q' may or may not be a non-violating sequence. We use a counter example to show that $P(s|q') \leq P(s|q) \leq C$ does not always hold. In Table 3, the sequence $q = \langle a1 \rightarrow b2 \rightarrow c3\rangle$ satisfies $P(\textit{On-welfare}|q) = 50\% \leq C$. However, its subsequence $q' = \langle a1 \rightarrow b2\rangle$ does not satisfy $P(\textit{On-welfare}|q') = 100\% > C$. ■

TABLE 3
Counter example for monotonic property

ID	Trajectory	Status	...
1	$\langle a1 \rightarrow d2\rangle$	Student	...
2	$\langle a1 \rightarrow b2\rangle$	On-welfare	...
3	$\langle a1 \rightarrow b2 \rightarrow c3\rangle$	On-welfare	...
4	$\langle a1 \rightarrow b2 \rightarrow c3\rangle$	Retired	...

To satisfy LKC-privacy, it is insufficient to ensure that every sequence q with only length L in T satisfies both the conditions of Definition 1. Instead, we need to ensure that every sequence q with length not greater than L in T satisfies both the conditions. To overcome this bottleneck of violating sequence enumeration, our insight is that there exists some "minimal" violating sequences among the violating sequences, and it is sufficient to achieve LKC-privacy by removing only the minimal violating sequences.

Definition 5 (Minimal violating sequence): A violating sequence q is a *minimal violating sequence* (*MVS*) if every proper subsequence of q is not a violating sequence. ■

Example 3: In Table 1, given $L = 3$, $K = 2$, $C = 50\%$, $S = \{\textit{On-welfare}\}$, the sequence $q = \langle b2 \rightarrow d3\rangle$ is a MVS because $\langle b2\rangle$ and $\langle d3\rangle$ are not violating sequences. The sequence $q = \langle b2 \rightarrow d3 \rightarrow c4\rangle$ is a violating sequence but not a MVS because its subsequence $\langle b2 \rightarrow d3\rangle$ is a violating sequence. ■

Every violating sequence is either a MVS or it contains a MVS. Thus, if T contains no MVS, then T contains no violating sequences.

Lemma 1: A RFID data table T satisfies LKC-privacy if and only if T contains no MVS.

Proof. Suppose a data table T does not satisfy LKC-privacy even if T contains no MVS. Then, by Definition 4, the table T contains violating sequence. But, a violating sequence must be a MVS or its subset is MVS, which is the contradiction of the initial assumption. Therefore, the data table T must satisfy LKC-privacy. ■

Next, we propose an algorithm to efficiently identify all MVS in T with respect to a LKC-privacy requirement. Based on Definition 5, we generate all MVS of size $i + 1$, denoted by V_{i+1}, by incrementally extending a non-violating sequence of size i, denoted by W_i, with an additional pair.

Algorithm 1 presents a method to efficiently generate all MVS. Line 1 puts all the size-1 sequences, i.e., all distinct pairs, as candidates X_1 of MVS. Line 4 scans T once to compute $|T(q)|$ and $P(s|q)$ for each sequence $q \in X_i$ and for each sensitive value $s \in S$. If the sequence q violates the LKC-privacy requirement in Line 6, then we add q to the MVS set V_i (Line 7); otherwise, add q to the non-violating sequence set W_i (Line 9) for generating the next candidate set X_{i+1}, which is a self-join of W_i (Line 12). Two sequences $q_x = \langle(loc_1^x t_1^x) \rightarrow \ldots \rightarrow (loc_i^x t_i^x)\rangle$ and $q_y = \langle(loc_1^y t_1^y) \rightarrow \ldots \rightarrow (loc_i^y t_i^y)\rangle$ in W_i can be joined only if the first $i - 1$ pairs of q_x and q_y are identical and $t_i^x < t_i^y$. The joined sequence is $\langle(loc_1^x t_1^x) \rightarrow \ldots \rightarrow$

Algorithm 1 MVS Generator

```
Input: Raw RFID data table $T$
Input: Thresholds $L$, $K$, and $C$
Input: Sensitive values $S$
Output: Minimal violating sequence $V(T)$
 1: $X_1 \leftarrow$ set of all distinct pairs in $T$;
 2: $i = 1$;
 3: while $i \leq L$ and $X_i \neq \emptyset$ do
 4:     Scan $T$ to compute $|T(q)|$ and $P(s|q)$, for $\forall q \in X_i$, $\forall s \in S$;
 5:     for $\forall q \in X_i$ where $|T(q)| > 0$ do
 6:         if $|T(q)| < K$ or $P(s|q) > C$ then
 7:             Add $q$ to $V_i$;
 8:         else
 9:             Add $q$ to $W_i$;
10:         end if
11:     end for
12:     $X_{i+1} \leftarrow W_i \bowtie W_i$;
13:     for $\forall q \in X_{i+1}$ do
14:         if $q$ is a super sequence of any $v \in V_i$ then
15:             Remove $q$ from $X_{i+1}$;
16:         end if
17:     end for
18:     $i$++;
19: end while
20: return $V(T) = V_1 \cup \ldots \cup V_{i-1}$;
```

$(loc_i^x t_i^x) \rightarrow (loc_i^y t_i^y)\rangle$. Lines 13-17 remove a candidate q from X_{i+1} if q is a super sequence of any sequence in V_i because any proper subsequence of a MVS cannot be a violating sequence. The set of MVS, denoted by $V(T)$, is the union of all V_i.

Example 4: Consider Table 1 with $L = 2$, $K = 2$, $C = 50\%$, and $S = \{$*On-welfare*$\}$. $X_1 = \{b2, d3, c4, c5, f6, c7, e8\}$. After scanning T, we divide X_1 into $V_1 = \emptyset$ and $W_1 = \{b2, d3, c4, c5, f6, c7, e8\}$. Next, from W_1 we generate the candidate set $X_2 = \{b2d3$, $b2c4$, $b2c5$, $b2f6$, $b2c7$, $b2e8$, $d3c4$, $d3c5$, $d3f6$, $d3c7$, $d3e8$, $c4c5$, $c4f6$, $c4c7$, $c4e8$, $c5f6$, $c5c7$, $c5e8$, $f6c7$, $f6e8$, $c7e8\}$. We scan T again to determine $V_2 = \{b2d3$, $b2c4$, $b2f6$, $c4c7$, $c4e8\}$. We do not further generate X_3 because $L = 2$. ■

Lemma 2: Algorithm 1 generates all the minimal violating sequences (MVS) of size $\leq L$.

Proof. We use a loop invariant to proof the correctness of Algorithm 1.

Loop Invariant: At the start of each iteration i of the while loop (Line 3), the MVS set $V(T)$ contains all the MVS of size $\leq (i - 1)$.

Initialization: Prior to the first iteration of the loop, $i = 1$, the MVS set $V(T)$ is empty. Invariant is true because by Definition 4 violating sequence can not be of size-0.

Maintenance: During the iteration, every candidate sequence $q \in X_i$ that does not satisfy $|T(q)| \geq K$ or $P(s|q) \leq C$ is added to the MVS set $V(T)$. Since, the candidate set contains all size-i sequences and the algorithm verifies all candidates, we conclude that loop invariant indeed remains true before the next iteration $i + 1$.

Termination: At termination, $i = L + 1$, by loop invariant, the MVS set $V(T)$ contains all the MVS of size $\leq L$. ■

Definition 6 (Violating pair): A pair p is a *violating pair* if it is part of a violating sequence. ■

Example 5: Given the set of minimal violating sequence, $V(T) = \{b2d3$, $b2c4$, $b2f6$, $c4c7$, $c4e8\}$, the violating pairs are $\{b2, d3, c4, f6, c7, e8\}$. ■

From Lemma 1, we have to remove all the MVS to satisfy LKC-privacy requirement. We can remove all the MVS by suppressing a subset of violating pairs. Given, $V(T) = \{b2d3$, $b2c4$, $b2f6$, $c4c7$, $c4e8\}$, we can either suppress $\{b2, c4\}$ or $\{b2, c7, e8\}$ and so on. Next, we prove that it is NP-hard to find an optimal subset of violating pairs.

Theorem 2: Given a RFID data table T and a LKC-privacy requirement, it is NP-hard to find the optimal anonymous solution.

Proof. The problem of finding the optimal anonymous solution can be converted into the *vertex cover problem* [7]. The vertex cover problem is a well-known problem in which, given an undirected graph $G = (V, E)$, it is NP-hard to find the smallest set of vertices S such that each edge has at least one endpoint in S. To reduce our problem into the vertex cover problem, we consider the set of violating pairs as the set of vertices V. The set of MVS, denoted by $V(T)$, is analogous to the set of edges E. Hence, the optimal vertex cover, S, means finding the smallest set of violating pairs that must be suppressed to obtain the optimal anonymous data set T'. Given that it is NP-hard to determine the smallest set of vertices S, it is also NP-hard to find the optimal set of violating pairs for suppression. ■

Finding an optimal solution for LKC-privacy is NP-hard. Thus, we propose a greedy algorithm to efficiently identify a reasonably "good" sub-optimal solution.

3.2 Eliminating Violating Sequences

We propose a greedy algorithm to transform the raw RFID data table T to an anonymous table T' with respect to a given LKC-privacy requirement by a sequence of suppressions. In each iteration, the algorithm selects a violating pair p for suppression based on a greedy selection function. In general, a suppression on a violating pair p in T increases privacy because it removes minimal violating sequences (MVS), and decreases data utility because it eliminates maximal frequent sequences (MFS) in T. Therefore, we define the greedy function, $Score(p)$, to select a suppression on a violating pair p that maximizes the number of MVS removed but minimizes the number of MFS removed in T. $Score(p)$ is defined as follows:

$$Score(p) = \frac{PrivGain(p)}{UtilityLoss(p) + 1} \qquad (1)$$

where $PrivGain(p)$ and $UtilityLoss(p)$ are the number of MVS and the number of MFS containing the violating pair p, respectively. A violating pair p may not belong to any MFS, resulting in $UtilityLoss(p) = 0$. To avoid dividing by zero, we add 1 to the denominator. The

Algorithm 2 Data Anonymizer

```
Input: Raw RFID data table $T$
Input: Thresholds $L$, $K$, $C$, and $K'$
Input: Sensitive values $S$
Output: Anonymous $T'$ that satisfies $LKC$-privacy
1:  Generate $V(T)$ by Algorithm 1 and build MVS-tree;
2:  Generate $U(T)$ by MFS algorithm and build MFS-tree;
3:  while $PG$ table is not empty do
4:      Select a pair $w$ that has the highest $Score$ to suppress;
5:      Delete all MVS and MFS containing $w$ from MVS-tree
        and MFS-tree;
6:      Update the $Score(p)$ if both $w$ and $p$ are contained in
        the same MVS or MFS;
7:      Remove $w$ from $PG$ Table;
8:      Add $w$ to $Sup$;
9:  end while
10: For $\forall w \in Sup$, suppress all instances of $w$ from $T$;
11: return the suppressed $T$ as $T'$;
```

violating pair p with the highest $Score(p)$ is called the *winner* pair, denoted by w.

Algorithm 2 summarizes the anonymization algorithm that removes all MVS. Line 1 calls Algorithm 1 to identify all MVS, denoted by $V(T)$, and then builds a MVS-tree with a PG table that keeps track of the $PrivGain(p)$ of all violating pairs for suppressions. Line 2 calls a maximal frequent sequence mining algorithm to identify all MFS, denoted by $U(T)$, and then builds a MFS-tree with a UL table that keeps track of the $UtilityLoss(p)$ of all candidate pairs. We modified *MAFIA* [6], which was originally designed for mining maximal frequent itemsets, to mine MFS. Any alternative MFS algorithm can be used as a plug-in to our method. At each iteration in Lines 3-9, the algorithm selects the winner pair w that has the highest $Score(w)$ from the PG table, removes all the MVS and MFS that contain w, incrementally updates the $Score$ of the affected violating pairs, and adds w to the set of suppressed values, denoted by Sup. Values in Sup are collectively suppressed in Line 10 in one scan of T. Finally, Algorithm 2 returns the anonymized T as T'. The most expensive operations are identifying the MVS and MFS containing w and updating the $Score$ of the affected candidates. Below, we propose two tree structures to efficiently perform these operations.

Definition 7 (MVS-tree): MVS-tree is a tree structure that represents each MVS as a tree path from root-to-leaf. Each node keeps track of a count of MVS sharing the same prefix. The count at the root is the total number of MVS. MVS-tree has a PG table that maintains every violating pair p for suppression, together with its $PrivGain(p)$. Each violating pair p in the PG table has a link, denoted by $Link_p$, that links up all the nodes in an MVS-tree containing p. $PrivGain(p)$ is the sum of the counts of MVS on $Link_p$. ■

Definition 8 (MFS-tree): MFS-tree is a tree structure that represents each MFS as a tree path from root-to-leaf. Each node keeps track of a count of MFS sharing the same prefix. The count at the root is the total number of MFS. MFS-tree has a UL table that keeps the $UtilityLoss(p)$ for every violating pair p. Each violating pair p in the UL table has a link, denoted by $Link_p$, that links up all the nodes in MFS-tree containing p. $UtilityLoss(p)$ is the sum of the counts of MFS on $Link_p$. ■

TABLE 4
Initial $Score$

	b2	d3	c4	f6	c7	e8
PrivGain	3	1	3	1	1	1
UtilityLoss (+1)	4	4	2	5	6	5
Score	0.75	0.25	1.5	0.2	0.16	0.2

TABLE 5
$Score$ after suppressing $c4$

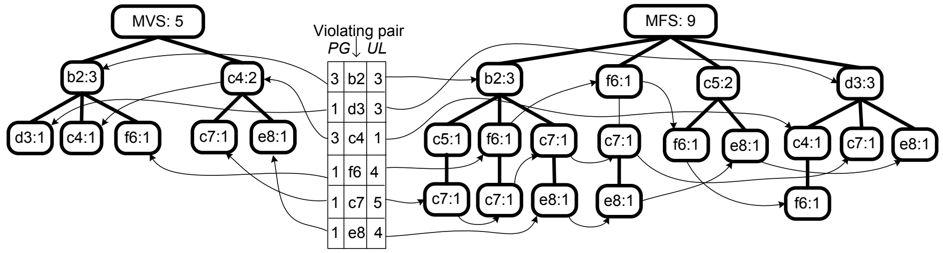
	b2	d3	f6
PrivGain	2	1	1
UtilityLoss (+1)	4	3	4
Score	0.5	0.33	0.25

Example 6: Figure 3 depicts both MVS-tree and MFS-tree generated from Table 1, where $V(T) = \{b2d3, b2c4, b2f6, c4c7, c4e8\}$ and $U(T) = \{b2c5c7, b2f6c7, b2c7e8, d3c4f6, f6c7e8, c5f6, c5e8, d3c7, d3e8\}$ with $L = 2$, $K = 2$, $C = 50\%$, and $K' = 2$. Each root-to-leaf path represents one sequence of MVS or MFS. To find all the MVS (or MFS) containing $c4$, follow $Link_{c4}$ starting from the PG (or UL) table. For illustration purposes, we show PG and UL as a single table. ■

Table 4 shows the initial $Score(p)$ of every violating pair. Identify the winner pair $c4$ from violating pairs. Then traverse $Link_{c4}$ to identify all MVS and MFS containing $c4$ and delete them from the MVS-tree and MFS-tree accordingly. These links are the key to efficient $Score$ updates and suppressions. When a winner pair w is suppressed from the trees, the entire branch of w is trimmed. The trees provide an efficient structure for updating the counts of MVS and MFS. For example, when $c4$ is suppressed, all its descendants are removed as well. The counts of $c4$'s ancestor nodes are decremented by the counts of the deleted $c4$ node. If a violating pair p and the winner pair w are contained in some common MVS or MFS, then $UtilityLoss(p)$, $PrivGain(p)$, and $Score(p)$, have to be updated by adding up the counts on $Link_p$. A violating pair p is removed from the PG table if $PrivGain(p) = 0$ because there is no more any MVS containing this pair. The resultant MVS-tree and MFS-tree are shown in Figures 4 after suppressing $c4$. Table 5 shows the updated $Score$ of the remaining violating pairs. In the next iteration, $b2$ is suppressed and thus all the remaining MVS are removed. Table 2 shows the resulting anonymous table T' for $(2, 2, 50\%)$-privacy.

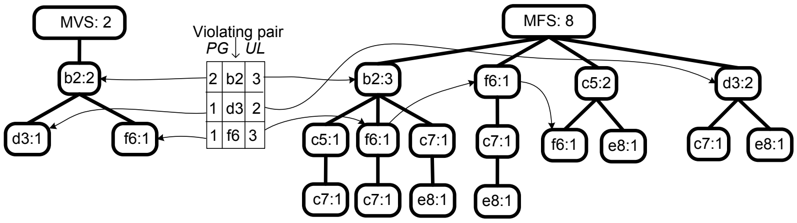
Lemma 3: Algorithm 2 eliminates all MVS without generating new MVS.

Proof. By Definition 7, MVS-tree represents all the MVS in a tree structure. Thus by suppressing the violating sequences iteratively, the algorithm eliminates all the MVS. However, global suppression does not generate



PG		UL
3	b2	3
1	d3	3
3	c4	1
1	f6	4
1	c7	5
1	e8	4

Fig. 3. MVS-tree and MFS-tree for efficient $Score$ updates



PG		UL
2	b2	3
1	d3	2
1	f6	3

Fig. 4. MVS-tree and MFS-tree after suppressing $c4$

any new MVS. Consider a new sequence q, which resulted from the suppression of its super sequence. The sequence q can not be a MVS since by Definition 5, all the subsequence of a MVS is a non-violating sequence. ■

We now prove that the anonymous data table T' is the LKC-private version of the raw data table T.

Theorem 3: Given a RFID data table T, the anonymous data table T' produced by the anonymization algorithm satisfies LKC-privacy.

Proof. The proof follows directly from Lemmas 1, 2 and 3. Since, the anonymization algorithm can enumerate all the MVS (Lema 2) and subsequently remove them without generating new MVS (Lema 3), the anonymous table contains no MVS. Finally, according to Lemma 1, the anonymous data table T' satisfies LKC-privacy because it has no MVS. ■

3.3 Complexity Analysis

Our anonymization algorithm has two steps. In the first step, we determine the set of MVS and the set of MFS. In the second step, we build the MVS-tree and MFS-tree, and suppress the violating pairs iteratively according to their $Score$. The most expensive operation of our algorithm is scanning the raw RFID data table T once to compute $|T(q)|$ and $P(s|q)$ for all sequence q in the candidate set X_i. This operation takes place during MVS generation. The cost of this operation is approximated as $Cost = \sum_{i=1}^{L} m_i i$, where $m_i = |X_i|$. Note that the searching cost depends on the value of L and size of the candidate set. When $i = 1$, the candidate set X_i is the set of all distinct pairs in T. Hence, the upper limit of $m_i = |d|$, where $|d|$ is the number of dimensions. It is unlikely to have any single pair violating the LKC-privacy; therefore, $m_2 = |d|(|d| - 1)/2$. In practice, most of the candidate sets are of size-2; therefore, the lower bound of the $Cost \leq m_1 + 2m_2 = |d|^2$. Finally, including the dependence on the data size, the time complexity of our algorithm is $O(|d|^2 n)$.

TABLE 6
Data sets statistics

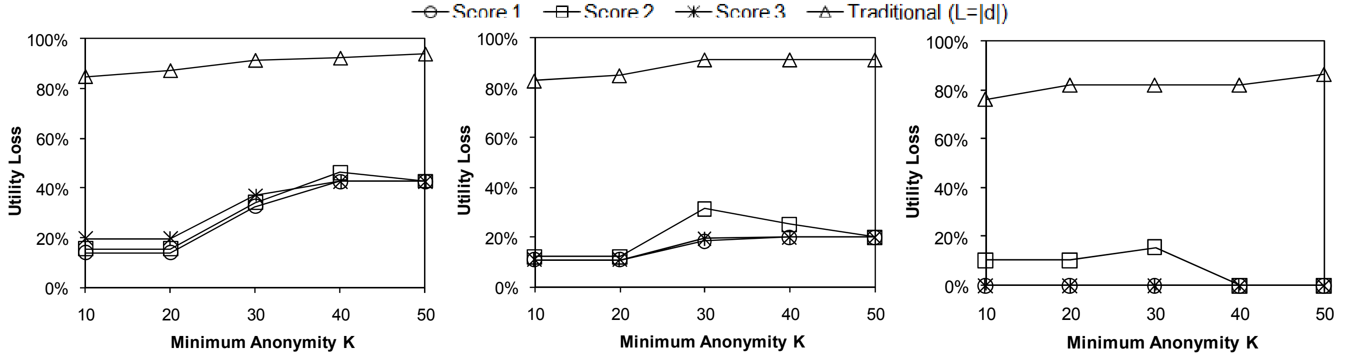
Dataset	Records $\|T\|$	Avg. trajectory length	Dimensions $\|d\|$	Data size (K bytes)
City80K	80,000	8	624	2,297
Metro100K	100,000	8	3,900	6,184

In the second step, we insert the MVS and MFS into the respective trees and delete them iteratively afterward. This operation is proportional to the number of MVS and thus in the order of $O(|V(T)|)$. Due to MVS-tree and MFS-tree data structures, our approach can efficiently calculate and update the the score of the violating pairs.

4 Empirical Study

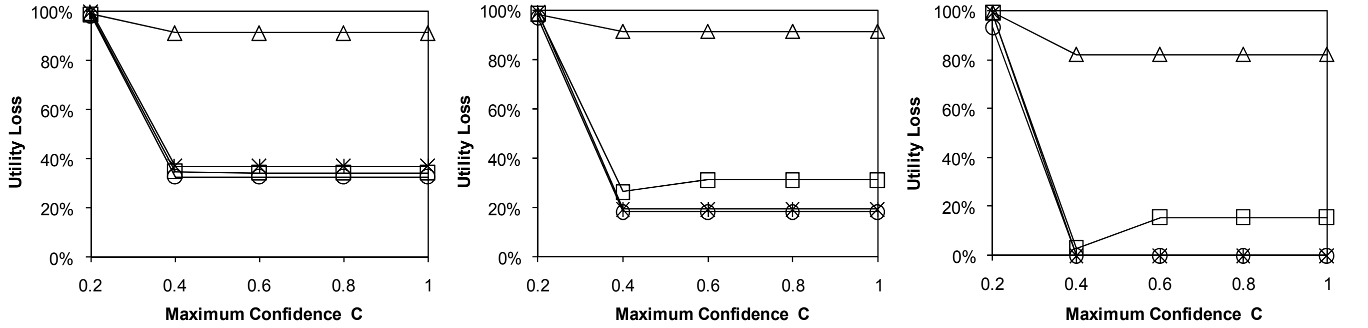
The main objective of our empirical study is to evaluate the performance of our proposed algorithm in terms of *utility loss* caused by anonymization, and *scalability* for handling large data sets. The utility loss is defined as $\frac{|U(T)|-|U(T)'|}{|U(T)|}$, where $|U(T)|$ and $|U(T)'|$ are the numbers of maximal frequent sequences before and after the anonymization of the data set T. It measures the percentage of MFS loss due to suppressions, so lower utility loss implies better data quality. We could not directly compare our methods with others because no method exists that can anonymize high-dimensional RFID data while preserving maximal frequent sequences. We convert the RFID data into relational data and attempt to apply the state-of-the-art anonymization algorithms, such as [12][20][34]. Unfortunately, all these methods are not scalable to high dimensionality and fail to finish the anonymization. We evaluate our algorithm with three different $Score$ functions:

- $Score1(p) = \frac{PrivGain(p)}{UtilityLoss(p)+1}$ (from Equation 1)
- $Score2(p) = PrivGain(p)$
- $Score3(p) = \frac{1}{UtilityLoss(p)+1}$



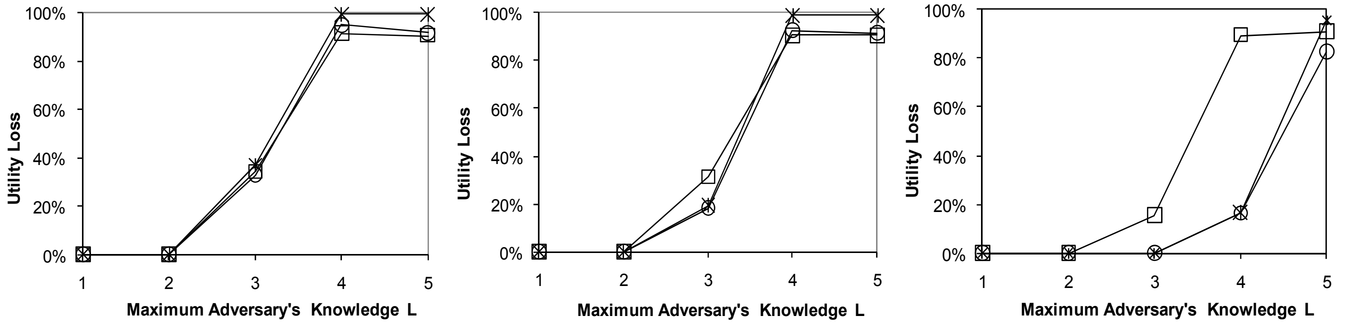
(a) $K' = 0.5\%$ (b) $K' = 1\%$ (c) $K' = 1.5\%$

Fig. 5. Utility loss vs. K on City80K ($L = 3, C = 60\%$)



(a) $K' = 0.5\%$ (b) $K' = 1\%$ (c) $K' = 1.5\%$

Fig. 6. Utility loss vs. C on City80K ($L = 3, K = 30$)



(a) $K' = 0.5\%$ (b) $K' = 1\%$ (c) $K' = 1.5\%$

Fig. 7. Utility loss vs. L on City80K ($K = 30, C = 60\%$)

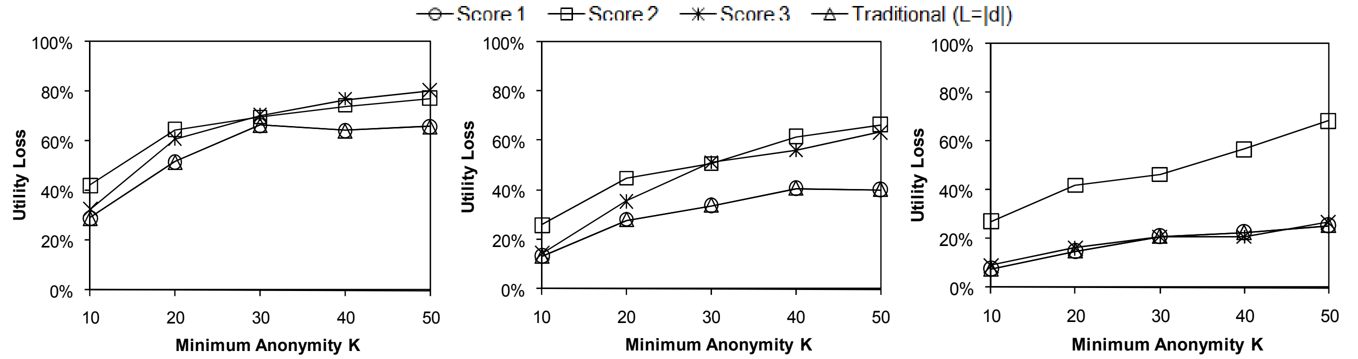
We used two data sets for the experiments: $City80K$ and $Metro100K$. $City80K$ is a data set simulating the routes of 80,000 citizens in a metropolitan area with 26 city blocks in 24 hours, thus forming 624 dimensions (different possible pairs). $Metro100K$ is a data set simulating the travel routes of 100,000 passengers in the Montreal subway transit system with 65 stations in 60 minutes, forming 3,900 dimensions. Each record in the data set corresponds to the route of one passenger. The passengers' traffic patterns are simulated based on information obtained from the Montreal metro information website[1]. Based on the published annual report, all the passengers have an average trajectory length of 8 stations. The data generator also simulates the trajectories according to the current metro map and passengers' flow in each station. In both data sets, each record contains an attribute with five possible values, where one of them is considered to be sensitive.

Following the convention for extracting MFS, we specify the minimum support threshold K' as the percentage of the total number of records in the database. For both data sets, we set $K' = 0.5\%$, and 1.5% and vary the thresholds of minimum anonymity K, maximum confidence C, and maximum adversary's knowledge L to evaluate the performance of the algorithm. All experiments are conducted on a PC with Intel Core2 Duo 1.6GHz CPU with 2GB of RAM.

4.1 Utility Loss

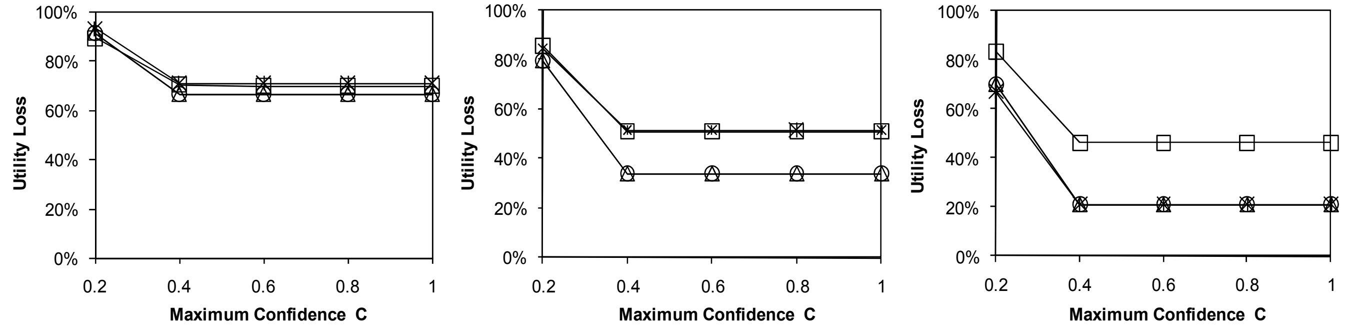
Figure 5. We vary the threshold K from 10 to 50 while fixing $L = 3$ and $C = 100\%$ on $City80K$. This setting allows us to measure the performance of the algorithm against identity linkages without considering attribute linkages. The utility loss of $Score1$ and $Score3$ generally increases as K increases, so it exhibits some trade-off

[1] http://www.metrodemontreal.com



(a) $K' = 0.5\%$ (b) $K' = 1\%$ (c) $K' = 1.5\%$

Fig. 8. Utility loss vs. K on Metro100K ($L = 3, C = 60\%$)



(a) $K' = 0.5\%$ (b) $K' = 1\%$ (c) $K' = 1.5\%$

Fig. 9. Utility loss vs. C on Metro100K ($L = 3, K = 30$)

between data privacy and data utility. The utility loss, sometimes, has a slight drop when K increases. This is due to the fact that the greedy algorithm finds only the sub-optimal solution. $Score2$ has higher utility loss than $Score1$ and $Score3$ because $Score2$ does not take into account the number of MFS lost during the elimination of MVS.

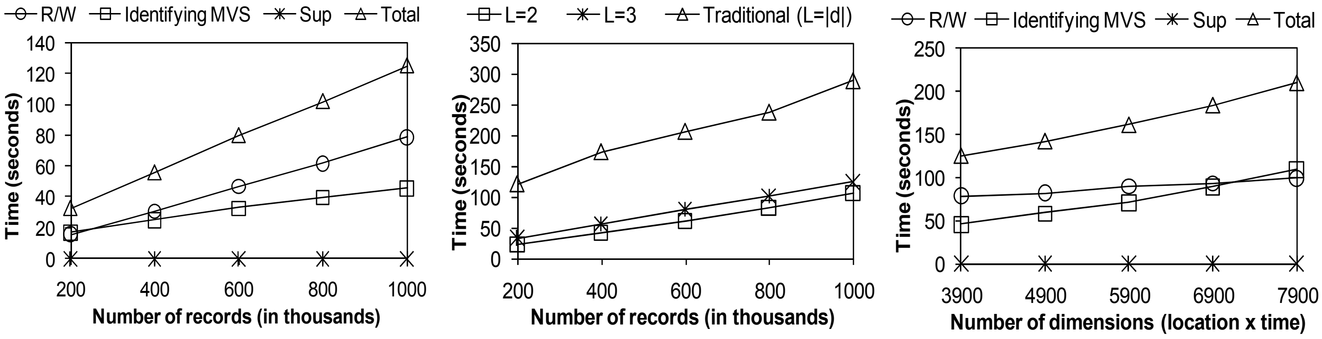
As K' increases, the utility loss decreases because the number of MFS decreases and there is less overlapping between $V(T)$ and $U(T)$, so suppressions have less effect on MFS. Though none of the traditional K-anonymization algorithms can handle the high-dimensional RFID data in our experiments, our method can achieve K-anonymity by setting $L = |d|$, where $|d|$ is the number of dimensions. The result strongly suggests that applying LKC-privacy would result in significantly lowering the utility loss than would applying traditional K-anonymity.

Figure 6. We vary the threshold C from 20% to 100% while fixing $L = 3$ and $K = 30$ on $City80K$. This allows us to examine the effect of attribute linkages. Approximately 1/5 of the records contain a sensitive value, so the utility loss is high at $C = 0.2$. As C increases, the effect of attribute linkages becomes insignificant. As K' increases, the utility loss drops quickly due to less overlapping between $V(T)$ and $U(T)$. The traditional confidence bounding anonymization method [34] cannot handle high-dimensional RFID data, so we achieve confidence bounding by setting $L = |d|$. Again, the traditional confidence bounding model results in significantly higher utility loss.

Figure 7. We vary the threshold L from 1 to 5 while fixing $K = 30$ and $C = 60\%$ on $City80K$. This allows us to quantify the utility loss with the increment of an adversary's background knowledge. The result suggests that up to $L = 2$, there is no utility loss. As L increases, the loss increases quickly due to the increase in the number of violating sequences.

Figure 8. $Metro100K$ is a relatively higher dimensional data set (3,900 dimensions) compared to $City80K$ (624 dimensions). Unlike in $City80K$, passengers follow predefined tracks based on the metro map. In Figure 8, following the same setting of $City80K$, we vary the value of K from 10 to 50, while fixing $L = 3$ and $C = 100\%$ on $Metro100K$. $Metro100K$ has a large number of violating sequences and thus many pairs are suppressed during anonymization. The general trend in $Metro100K$ is more obvious than in $City80K$. For example, in Figure 8(a), as K increases from 10 to 50, the utility loss of $Score1$ increases from 29% to 66%. As K' increases from 0.5% to 1.5%, the utility loss of $Score1$ at $K = 30$ drops from 66% to 21%. In all test cases, $Score1$ and $Score3$ consistently outperform $Score2$, suggesting that it is vital to consider the loss of MVS in the greedy function. Interestingly, the utility loss is the same for $L = 3$ and $L = |d|$ because most of the MVS are of size-3 or less. In other words, there is no difference between $L = 3$ and $L = 4$ or above in terms of the generated MVS. Hence, the utility loss for $L \geq 3$ remains unchanged; therefore, we omit the figure on utility loss vs. L.

Figure 9. We vary the value of C from 20% to 100% while fixing $L = 3$ and $K = 30$ on $Metro100K$. The



(a) Runtime vs. # of records

(b) Runtime vs. # of L

(c) Runtime vs. dimensionality

Fig. 10. Scalability $(K = 30, C = 60\%, K' = 1\%)$

results have characteristics similar to those in Figure 6. The utility loss increases when $C < 40\%$. Moreover, as K' increases, the utility loss decreases significantly.

4.2 Scalability

One major contribution of our work is the development of an efficient and scalable algorithm for achieving LKC-privacy, traditional K-anonymity, and confidence bounding on high-dimensional RFID data. Every previous test case can finish the entire anonymization process within 15 seconds. We further evaluate scalability with respect to data volume and dimensionality. We conduct all the experiments on the data set $Metro100K$ since it is larger in size and dimensionality. Unless otherwise specified, we fix $L = 3$, $K = 30$, $C = 60\%$, and $K' = 1\%$.

Figure 10(a) depicts the runtime in seconds from 200,000 to 1 million records. The total runtime for anonymizing 1 million records is 125 seconds, of which 46 seconds are spent identifying MVS and 79 seconds are spent reading the raw data set and writing the anonymous data set. It takes less than 1 second to suppress all the MVS due to our efficient MVS-tree and MFS-tree. As the number of records increases from 200,000 towards 1 million, the runtime for read/write and identifying MVS also increases linearly, suggesting that our algorithm is scalable to anonymize large datasets. Figure 10(b) compares the total runtime for $L = 2$, $L = 3$, and $L = |d|$. $L = |d|$ represents the runtime for achieving traditional K-anonymity and confidence bounding. The runtime for achieving those models is much longer than ours because $L = |d|$ requires verifying many sequences up to $L = |d|$. In Figure 10(c), we increase the dimension on the data set with 1 million records. As the number of dimensions increases, the number of MVS also increases due to sparseness; therefore, the runtime for identifying MVS also increases.

4.3 Summary

(1) As anonymity threshold K or an adversary's knowledge L increases, the data utility decreases. The trend is less obvious on C. (2) As minimum support threshold K' increases, the set of MVS and the set of MFS have less overlapping, so suppressing pairs in MVS has less effect on MFS. (3) $Score1$ and $Score3$ outperforms $Score2$, suggesting it is important to consider the loss of MFS in the greedy function. (4) High-dimensional data generally has more violating sequences and, therefore, higher utility loss. (5) Our proposed method is scalable with respect to the data size.

5 Discussion

In this section, we provide answers to the following frequently raised questions: Why does the data holder want to publish the sensitive attributes when the goal is to preserve maximal frequent sequences? Can the proposed algorithm be applied to anonymize any data set involving moving objects? What if the adversary only uses time or location to identify an individual?

Sensitive Attribute. The data hold may publish the sensitive attributes because some data mining tasks on RFID data require both trajectory and object-specific data. Analyzing the workflow (traffic flow) without understanding what the objects are often meaningless. For example, transit companies like to understand the characteristics of the passengers' traffic. However, if there is no such data mining purpose, the sensitive attributes should be removed. Our proposed anonymization algorithm (Section 3) is flexible enough to handle RFID data with or without sensitive attributes. Note that, none of the previous works consider the privacy threats caused by attribute linkages between the trajectory and the sensitive attributes.

Trajectory from Moving Objects. Our algorithm requires the trajectories to have the following form $\langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_n t_n) \rangle$. If the trajectory of a moving object does not have this form, then preprocessing is needed before applying the proposed algorithm. For example, the trajectory from a mobile device is as a sequence of spatio-temporal points in the form $\langle (x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n) \rangle$, where $t_1 < t_2 < \ldots t_n$ and the coordinate (x_i, y_i) represents the location of the device at time t_i, obtained with the help of GPS devices and/or by localization techniques. In the preprocessing step, the space can be divided into $\epsilon \times \epsilon$ grids, where each coordinate is represented by a grid. Thus, the continuous spatio-temporal points can be transformed into discrete $(loc_i t_i)$ pairs, where each grid is represented by loc_i. Once the trajectories are transformed into the $(loc_i t_i)$ pairs, the proposed algorithm can be used.

Time and Location. It is possible that the adversary's background knowledge q' contains only the location loc_i or only the timestamp t_i. This type of attack is obviously weaker than the attack based on background knowledge q containing $(loc_i t_i)$ because the identified group $|T(q')| \geq |T(q)|$. Thus, an LKC-privacy preserved table that can thwart linkages on q can also thwart linkages on q'.

6 Related Work

Privacy-preserving techniques on RFID can be broadly grouped into two categories: data collection and data publishing. While the work on data collection focuses on the privacy and security issues of the RFID tags and readers at the communication level [19], the work on data publishing phase focuses on the privacy and utility at the data level [16]. Below, we briefly summarize the techniques applicable to RFID data privacy.

Location Privacy. Different solutions have been proposed to protect the privacy of location-based service (LBS) users. The anonymity of a user in LBS is achieved by mixing the user's identity and request with other users. Example of such techniques are Mix Zones [5], cloaking [17], and location-based k-anonymity [13]. The objective of these techniques is very different from ours. First, their goal is to anonymize an individual user's identity resulting from a set of LBS requests, but our goal is to anonymize a high-dimensional RFID data set. Second, they deal with small dynamic groups of users at a time, but we anonymize a large static data set. Hence, their problem is very different from RFID data publishing.

Privacy Models. Traditional K-anonymity [29], [31], ℓ-diversity [25], confidence bounding [34], and (α, k)-anonymity [36] are based on a predefined set of QID attributes. As discussed earlier, a traditional QID-based approach suffers from the curse of high dimensionality [2] and renders the high-dimensional data useless for data mining. In this paper, we solve the problem of dimensionality by assuming that the adversary knows at most L pairs of a victim's locations and the corresponding times. In [27], Mohammed et al. propose LKC-privacy model that addresses the privacy issues on high-dimensional relational data. They adopt top-down approach to generalize relational data while preserving data utility for classification analysis. Unlike [27], the proposed algorithm suppresses the violating pairs based on a heuristic that preserves maximal frequent sequences. This is the first paper that propose an anonymization algorithm to achieve LKC-privacy model on RFID data. Furthermore, none of the tested traditional QID-based anonymization methods, namely [12][20][25], are scalable to handle the high-dimensional data in our experiments. Since K-anonymity [29][31], confidence bounding [34], and (α, k)-anonymity [36] are special cases of the LKC-privacy model, our anonymization algorithm can also be viewed as a scalable solution for achieving a traditional privacy model for RFID data.

Dwork [9] proposes a privacy model called *differential privacy*, which ensures that the removal or addition of a single data record does not significantly affect the overall privacy of the database. Most of the works in differential privacy are based on interactive privacy model, where the result of a query is in the form of aggregation [8], [10].

Anonymizing High-Dimensional Data. There are some recent works on anonymizing high-dimensional transaction data [14][33][38][39]. The methods presented in [33][38][39] model the adversary's power by a maximum number of known items as background knowledge. This assumption is similar to ours, but our problem has two major differences. First, a transaction is a *set* of items, but a moving object's trajectory is a *sequence* of visited location-time pairs. Sequential data drastically increases the computational complexity for counting the support counts as compared to transaction data because $\langle a \rightarrow b \rangle$ is different from $\langle b \rightarrow a \rangle$. Hence, their proposed models are not applicable to spatio-temporal data. Second, we have different privacy and utility measures. The privacy model of [33] is based on only K-anonymity and does not consider attribute linkages. Xu et al. [38][39] measure their data utility in terms of preserved item instances and frequent itemsets, respectively, while we measure the utility based on the number of preserved maximal frequent sequences.

Anonymizing Moving Objects. Some recent works [1], [18], [32], [28], [40], [11] address the anonymity of moving objects. Abul et al. [1] propose a new privacy model called (k, δ)*-anonymity* that exploits the inherent uncertainty of moving objects' locations. Their method relies on a basic assumption that every trajectory is continuous. Though this assumption is valid for GPS-like devices where the object can be traced all the time, it does not hold for RFID-based moving objects. Another major difference is that [1] achieves the anonymity by space translation that changes the actual location of an object. In contrast, our approach employs suppression for anonymity and thus preserves the data truthfulness and maximal frequent sequences with true support counts. Hoh et al. [18] present an uncertainly-aware privacy algorithm for GPS traces. They selectively remove trajectory pairs to increase uncertainly between trajectories to hinder identification. Both the works target GPS traces and can not be employed for anonymizing RFID data.

The privacy model proposed in [32] assumes that different adversaries have different background knowledge about the trajectories, and thus their objective is to prevent adversaries from gaining any further information from the published data. They consider the locations in a trajectory as sensitive information and assume that the data holder has the background knowledge of all the adversaries. In reality, such information is difficult to obtain. Pensa et al. [28] propose a k-anonymity notion for sequence datasets. The proposed algorithm also aims to preserve frequent sequential patterns. However to achieve anonymity, they transform a sequence into the other by insertion, deletion or substitution of a single item. Thus, their approach also spoils data truthfulness. Yarovoy et al. [40] consider time as a QID attribute. However, there is no fixed set of time for all moving objects. Each trajectory has its own set of times as its QID. It is unclear how the data holder can determine the QID attributes for each trajectory. Finally, Fung et al. [11] propose a method for anonymizing RFID data without preserving maximal frequent sequences. As shown in Section 4, it is important to consider the loss of MFS in order to preserve MFS.

This paper is the extension of our previous work [26], where we address the problem of achieving anonymity and preserving maximal frequent sequences. In this paper, we propose an efficient data structure for eliminating violating sequences (Section 3.2). We also evaluate our proposed algorithm though experiments to demonstrate that our anonymization algorithm can effectively retain the essential information in anonymous data and is scalable for anonymizing high-dimensional data sets.

7 Conclusion

We have studied the problem of anonymizing high-dimensional RFID data and have illustrated that traditional QID-based anonymization methods, such as K-anonymity and its variants, are not suitable for anonymizing RFID data, due to the curse of high dimensionality. Applying K-anonymity on high-dimensional data would result in a high utility loss. To overcome the problem, we adopt LKC-privacy model based on a practical assumption that an adversary has limited background knowledge about the victim. We also presented an efficient algorithm for achieving LKC-privacy with the goal of preserving maximal frequent sequences, which serves as the basis of many data mining tasks on sequential data. One future work is to address privacy threats caused by the combination of QID attributes and RFID trajectory data of the moving objects.

References

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *IEEE ICDE*, 2008.

[2] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, 2005.

[3] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, 1994.

[4] Z. Berenyi and H. Charaf. Retrieving frequent walks from tracking data in RFID-equipped warehouses. In *Human System Interactions*, 2008.

[5] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2003.

[6] D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: a maximal frequent itemset algorithm for transactional databases. In *IEEE ICDE*, 2001.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

[8] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *ACM PODS*, 2003.

[9] C. Dwork. Differential privacy. In *ICALP*, 2006.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[11] B. C. M. Fung, M. Cao, B. C. Desai, and H. Xu. Privacy protection for RFID data. In *ACM SIGAPP Symposium on Applied Computing (SAC)*, 2009.

[12] B. C. M. Fung, K. Wang, and P. S. Yu. Anonymizing classification data for privacy preservation. *IEEE TKDE*, 2007.

[13] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE TMC*, 2007.

[14] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In *IEEE ICDE*, 2008.

[15] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Trajectory pattern mining. In *ACM SIGKDD*, 2007.

[16] H. Gonzalez, J. Han, and X. Li. Mining compressed commodity workflows from massive RFID data sets. In *ACM CIKM*, 2006.

[17] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[18] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *ACM CCS*, 2007.

[19] A. Juels. RFID security and privacy: a research survey. *IEEE J-SAC*, 2006.

[20] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *ACM SIGMOD*, 2005.

[21] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *IEEE ICDE*, 2006.

[22] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization techniques for large-scale data sets. *ACM TODS*, 2008.

[23] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3):177–206, 2002.

[24] C. Luo and S. Chung. A scalable algorithm for mining maximal frequent sequences using sampling. In *ICTAI*, 2004.

[25] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 2007.

[26] N. Mohammed, B. C. M. Fung, and M. Debbabi. Walking in the crowd: Anonymizing trajectory data for pattern analysis. In *ACM CIKM*, 2009.

[27] N. Mohammed, B. C. M. Fung, P. C. K. Hung, and C. K. Lee. Anonymizing healthcare data: a case study on the blood transfusion service. In *ACM SIGKDD*, 2009.

[28] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern-preserving k-anonymization of sequences and its application to mobility data mining. In *International Workshop on Privacy in Location-Based Applications*, 2008.

[29] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knwledge and Data Engineering (TKDE)*, 2001.

[30] S. Spiekermann and S. Evdokimov. Critical RFID privacy-enhancing technologies. *IEEE Security and Privacy*, 2009.

[31] L. Sweeney. k-anonymity: A model for protecting privacy. In *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[32] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, 2008.

[33] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. In *VLDB*, 2008.

[34] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *KAIS*, 2007.

[35] S.-W. Wang, W.-H. Chen, C.-S. Ong, L. Liu, and Y. Chuang. RFID applications in hospitals: a case study on a demonstration RFID project in a taiwan hospital. In *HICSS*, 2006.

[36] R. C. W. Wong, J. Li., A. W. C. Fu, and K. Wang. (α,k)-anonymous data publishing. *Journal of Intelligent Information Systems*, in press.

[37] X. Xiao and Y. Tao. Personalized privacy preservation. In *SIGMOD*, Chicago, IL, 2006.

[38] Y. Xu, B. C. M. Fung, K. Wang, A. W. C. Fu, and J. Pei. Publishing sensitive transactions for itemset utility. In *IEEE ICDM*, 2008.

[39] Y. Xu, K. Wang, A. W. C. Fu, and P. S. Yu. Anonymizing transaction databases for publication. In *ACM SIGKDD*, 2008.

[40] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang. Anonymizing moving objects: How to hide a MOB in a crowd? In *EDBT*, 2009.