Privacy Protection for RFID Data

Benjamin C. M. Fung
Concordia Institute for
Information Systems
Engineering
Concordia University
Montreal, QC, Canada
fung@ciise.concordia.ca

Ming Cao
Concordia Institute for
Information Systems
Engineering
Concordia University
Montreal, QC, Canada
min_ca@encs.concordia.ca

Bipin C. Desai
Department of Computer
Science & Software
Engineering
Concordia University
Montreal, QC, Canada
bcdesai@cs.concordia.ca

Heng Xu
College of Information
Sciences and Technology
Penn State University
University Park, PA 16802
hxu@ist.psu.edu

ABSTRACT

Radio Frequency IDentification (RFID) is a technology of automatic object identification. Retailers and manufacturers have created compelling business cases for deploying RFID in their supply chains. Yet, the uniquely identifiable objects pose a privacy threat to individuals. In this paper, we study the privacy threats caused by publishing RFID data. Even if the explicit identifying information, such as name and social security number, has been removed from the published RFID data, an adversary may identify a target victim's record or infer her sensitive value by matching a priori known visited locations and timestamps. RFID data by default is high-dimensional and sparse, so applying traditional K-anonymity to RFID data suffers from the curse of high dimensionality, and would result in poor data usefulness. We define a new privacy model, develop an anonymization algorithm to accommodate special challenges on RFID data, and evaluate its performance in terms of data quality, efficiency, and scalability. To the best of our knowledge, this is the first work on anonymizing high-dimensional RFID data.

Categories and Subject Descriptors

H.2.7 [**Database Administration**]: Security, integrity, and protection; H.2.8 [**Database Applications**]: Data mining

Keywords

Information sharing, privacy protection, anonymity, sensitive information, data mining

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.
Copyright 2009 ACM 978-1-60558-166-8/09/03 ...$5.00.

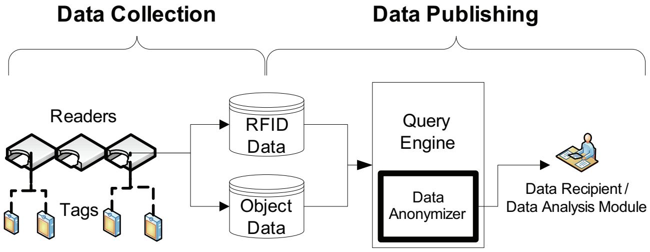


Figure 1: Data Flow in RFID System

1. INTRODUCTION

Radio Frequency IDentification (RFID) is a technology for automatic identification of single or bulk objects from a distance, using radio signals. RFID has wide applications in many areas including manufacturing, healthcare, and transportation. Figure 1 depicts an overview of a RFID information system, typically consisting of a large number of tags and readers and an infrastructure for handling high volume of RFID data. A tag is a small device that can be attached to an object, such as a person or a manufactured item, for the purpose of unique identification. A reader is an electronic device positioned in a strategic location, such as warehouse loading bay or metro station entrance, that communicates with the RFID tag. A reader broadcasts a radio signal to the tag, which then transmits its information back to the reader [17]. Streams of RFID data records, in the format of $\langle EPC, loc, t\rangle$, are then stored in a RFID database, where EPC (Electronic Product Code) is a unique identifier of the tagged object, loc is the location of the reader, and t is the time of detection. A data recipient (or a data analysis module) could obtain the information on either specific tagged objects or general workflow patterns [10] by submitting data requests to the query engine. The query engine then responds to the requests by joining the RFID data with some object-specific data.

Retailers and manufacturers have created compelling business cases for deploying RFID in their supply chains, from reducing out-of-stocks at Wal-Mart to up-selling consumers

Table 1: Raw patient-specific path table T

EPC	Path	Diagnosis	...
1	$\langle a1 \to d2 \to b3 \to e4 \to f6 \to c7 \rangle$	HIV	
2	$\langle b3 \to e4 \to f6 \to e8 \rangle$	Flu	
3	$\langle b3 \to c7 \to e8 \rangle$	Flu	
4	$\langle d2 \to f6 \to c7 \to e8 \rangle$	Allergy	
5	$\langle d2 \to c5 \to f6 \to c7 \rangle$	HIV	
6	$\langle c5 \to f6 \to e9 \rangle$	Allergy	
7	$\langle d2 \to c5 \to c7 \to e9 \rangle$	Fever	
8	$\langle f6 \to c7 \to e9 \rangle$	Fever	

in Prada. Yet, the uniquely identifiable objects pose a privacy threat to individuals, such as tracing a person's movements, and profiling individuals become possible. Most previous work on privacy-preserving RFID technology [17] focused on the threats caused by the physical RFID tags. They proposed techniques like EPC re-encryption and killing tags [11] to address the privacy issues in the *data collection* phase, but these techniques cannot address the privacy threats in the *data publishing* phase, when a large volume of RFID data is released to a third party.

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 object-specific RFID data into a version that is immunized against privacy attacks. The term "publishing" has a broad sense here. It includes sharing the RFID data with specific recipients and releasing data for public download. The general assumption is that the recipient could be an adversary, who attempts to associate a target victim (or multiple victims) to some sensitive information from the published data.

There are many real-life examples on RFID data publishing in healthcare [23]. Recently, some hospitals have adopted RFID sensory system to track the positions of their patients, doctors, medical equipments, and devices inside a hospital, with the goals of minimizing medical errors and improving the management of patients and resources. Analyzing RFID data, however, is a non-trivial task. The hospital management often does not have the expertise to perform the analysis hence outsource this process, thus, requires granting a third party access to the RFIDs and patient data. The following example illustrates the privacy threats caused by publishing RFID data.

Example 1.1. A hospital wants to release the patient-specific path table, Table 1, to a third party for data analysis. Explicit identifiers, such as patient names and *EPC*, are removed. Each record contains a *path* and some patient-specific information, where a *path* contains a sequence of *pairs* ($loc_i t_i$) indicating the patient's visited location loc_i at timestamp t_i. For example, $EPC\#3$ has a path $\langle b3 \to c7 \to e8 \rangle$, meaning that the patient has visited locations b, c, and e at timestamps 3, 7, and 8, respectively. Without loss of generality, we assume that each data record contains only one sensitive attribute, namely diagnosis, in this example.

A data recipient, who could be an adversary, seeks to identify the record and/or sensitive value of a target victim from the published data. We focus on two types of privacy attacks: (1) *Record linkage*: if a path in the table is so specific that not many people match it, releasing the RFID data may lead to linking the victim's record, and therefore, her contracted diagnosis. Suppose that the adversary knows that the target victim, Alice, has visited e and c at timestamps 4 and 7, respectively. Alice's record, together with her sensitive value (HIV in this case), can be uniquely identified because $EPC\#1$ is the *only* record that contains $e4$ and $c7$. (2) *Attribute linkage*: if a sensitive value occurs frequently together with some combination of pairs, then the sensitive information can be inferred from such combination even though the exact record of the victim cannot be identified. Suppose the adversary knows that another target victim, Bob, has visited $d2$ and $f6$. Since two out of the three records ($EPC\#$1,4,5) containing $d2$ and $f6$ have sensitive value HIV, the adversary can infer that Bob has HIV with $2/3 = 67\%$ confidence. ■

Table 2: Anonymous table T' for L=2, K=2, C=50%

EPC	Path	Diagnosis	...
1	$\langle b3 \to f6 \to c7 \rangle$	HIV	
2	$\langle b3 \to f6 \to e8 \rangle$	Flu	
3	$\langle b3 \to c7 \to e8 \rangle$	Flu	
4	$\langle f6 \to c7 \to e8 \rangle$	Allergy	
5	$\langle c5 \to f6 \to c7 \rangle$	HIV	
6	$\langle c5 \to f6 \to e9 \rangle$	Allergy	
7	$\langle c5 \to c7 \to e9 \rangle$	Fever	
8	$\langle f6 \to c7 \to e9 \rangle$	Fever	

Many privacy models, such as K-anonymity [3][4][5][6][7][12][16][20][26], ℓ-diversity [14], confidence bounding [21][22], and t-closeness [13] have been proposed to thwart privacy threats caused by record linkages and attribute linkages in the context of relational databases. All these works assume a given set of attributes called *quasi-identifier* (QID) that can identify an individual. Although these privacy models are effective for anonymization on relational databases, they are not applicable to RFID data due to two special challenges posed by RFID data:

High dimensionality: RFID data by default is high-dimensional due to the large combinations of locations and timestamps. Consider a hospital having 50 rooms that operate 12 hours per day. The RFID data table would have $50 \times 12 = 600$ dimensions. Each dimension could be a potential quasi-identifying (QID) attribute used for record or attribute linkages. Traditional privacy model, say K-anonymity, would include all dimensions into a single QID and require every path to be shared by at least K records. Due to the curse of high dimensionality [2], it is very likely that lots of data have to be suppressed in order to satisfy K-anonymity. For example, to achieve 2-anonymity in Table 1, $a1, d2, b3, e4, c7, e9$ have to be suppressed even if K is small. Such anonymous data becomes useless for data analysis.

Data sparseness: RFID data is usually sparse. Consider patients in a hospital or passengers in a public transit system. They usually visit only few locations compared to all available locations, so each RFID path is relatively short. Anonymizing these short paths in a high-dimensional space poses great challenge for traditional anonymization techniques because the paths have little overlap. Enforcing K-anonymity on sparse data would render the data useless.

Traditional K-anonymity and its extended privacy models assume that a QID contains all attributes (dimensions) because the adversary could potentially use any or even all QID attributes as prior knowledge to perform record or attribute linkages. However, in real-life privacy attacks, it is unlikely that an adversary could know *all* locations and timestamps that the target victim has visited because it requires non-trivial effort to gather each piece of prior knowledge from so many possible locations at different time. Thus, it is reasonable to assume that the adversary's prior knowledge is bounded by at most L pairs of locations and timestamps that the target victim has visited.

Based on this assumption, we define a new privacy model called *LKC-privacy* for anonymizing high-dimensional, sparse RFID data. The general intuition is to ensure that every possible subsequence q with maximum length L in any path of a RFID data table T is shared by at least K records in T and the confidence of inferring any sensitive values S from q is not greater than C, where L and K are positive integer thresholds, $0 \leq C \leq 1$ is a real number threshold, and S is a set of sensitive values specified by the data holder. LKC-privacy bounds the probability of a successful record linkage attack to be $\leq 1/K$ and bounds the probability of a successful attribute linkage attack to be $\leq C$, provided that the adversary's prior knowledge on the target victim is not more than L pairs of locations and timestamps. Table 2 shows an example of anonymous table T' that satisfies $(2, 2, 50\%)$-privacy by suppressing $a1, d2, e4$ from Table 1. Every possible subsequence q with maximum length 2 is shared by at least 2 records and the confidence of inferring the sensitive value HIV from q is not greater than 50%. In contrast, to achieve traditional 2-anonymity, we need to further suppress $b3, c7, e9$, resulting in much higher information loss.

Our contributions in this paper are summarized as follows. First, we formally define a new privacy model, called LKC-privacy, for anonymizing high-dimensional, sparse RFID data (Section 3). Second, we propose an efficient anonymization algorithm to transform a table to satisfy a given LKC-privacy requirement (Section 4). Finally, we evaluate the performance of our proposed model and method in terms of data quality, efficiency, and scalability (Section 5). To the best of our knowledge, this is a pioneering work on anonymizing high-dimensional, sparse RFID data.

2. RELATED WORK

Most previous research on RFID focuses on utilizing RFID technology and analyzing RFID data [9][10]. Solutions for addressing its privacy issues are limited. Below, we summarize the literature related to RFID privacy.

A comprehensive privacy-preserving information system must protect its data throughout its lifecycle, from data collection to data analysis. Most previous work on privacy-preserving RFID technology [17] focused on the threats caused by the physical RFID tags and proposed techniques like killing tags, sleeping tags, and EPC re-encryption [11]. They addressed the privacy and security issues at the communication layer among tags and readers, but ignored the protection of the database layer, where a large amount of RFID data actually resides. This paper provides a complement to the existing privacy-preserving RFID hardware technology.

The database community has spent lots of effort on privacy-preserving data publishing, where the goal is to transform relational data into an anonymous version for preventing record and attribute linkages. K-anonymity [3][12][16] and its extensions [4][5][6][7][13][14][20][21][22][24][25][26] are not applicable to anonymizing RFID data due to the problem of high dimensionality [2] and data sparseness discussed in Section 1. We tackle this challenge by exploiting the assumption that the adversary knows at most L pairs of previously visited locations and timestamps by a target victim.

There are some recent study of anonymizing high-dimensional transaction data [8][19][27]. Ghinita et al [8] proposed a permutation method, which the general idea is to first group transactions with close proximity and then associate each group to a set of mixed sensitive values.Terrovotis et al [19] and Yu et al [27] extended the traditional K-anonymity model by assuming that the adversary knows at most m transaction items of the target victim. All these works [8][19] [27] consider a transaction as a *set* of items. In contrast, our RFID path is a *sequence* of locations and timestamps. In our model, an adversary having prior knowledge sequence $\langle a, b\rangle$ is considered to be different from prior knowledge of sequence $\langle b, a\rangle$; therefore, their proposed privacy models and methods are not applicable to our problem.

Recently, there are few works on anonymizing moving objects [1][18]. [1] extends the traditional K-anonymity model to anonymize a set of moving objects. The intuition is to have at least K moving objects within the radius of the path of every moving object, where the radius is a user-specified threshold. Our approach is different from [1] in two major aspects. First, their model does not consider the privacy threats caused by attribute linkage between the path and the sensitive attribute. Second, they assume that all moving objects have continuous timestamps. This assumption may hold in mobile phone or LBS applications, where the user's location is continuously detected while the phone is turned on. However, this assumption does not hold for RFID because a RFID-tagged object (e.g., smart cards used in transportation) is unlikely to be continuously detected by a RFID reader. These differences imply different privacy threats and models. Terrovitis et al [18] assumes a very different attack model on moving objects. They consider that the locations themselves are sensitive information and the adversary attempts to infer some sensitive locations visited by the target victim that are unknown to the adversary. They do not specifically address the high dimensionality problem in RFID data, which is the theme of this paper.

3. PROBLEM DEFINITION

3.1 Object-Specific Path Table

A typical RFID system generates a sequence of RFID data records of the form $\langle EPC, loc, t\rangle$, where each record indicates a RFID reader in location loc has detected an object having electronic product code (EPC) at time t. We assume that the RFID-tagged item is attached to or carried by some moving object, for example, patients in a hospital or passengers in a public transit system.

A *pair* $(loc_i t_i)$ represents that the object has visited location loc_i at time t_i. The *path* of an object, denoted by $\langle (loc_1 t_1) \ldots (loc_n t_n)\rangle$, is a sequence of pairs that can be obtained by first grouping the RFID records by EPC and then sorting the records in each group by timestamps. A timestamp is the entry time to a location, so the object is assumed to be staying in the same location until its new location is detected by another reader. An object may revisit the same locations at different timestamps, but consecutive pairs having the same location are duplicates and, therefore, are removed. For example, in $\langle a1 \rightarrow b3 \rightarrow b4 \rightarrow b6 \rightarrow c7 \rightarrow b8\rangle$, $b4$ and $b6$ are removed but $b8$ is kept. At any time, an object can be at only one location, so $a1 \rightarrow b1$ is not a valid sequence. Timestamps in a path must increase monotonically.

An *object-specific path 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 a path, $s_i \in S_i$ are sensitive attributes, and $d_i \in D_i$ are quasi-identifying (QID) attributes associated with the object. In the rest of this paper, the term "record" refers to the above form. The QID attributes are relational data and can be anonymized by existing methods [7][13][14][16][22] for relational data. This paper focuses on the paths and sensitive attributes.

3.2 Privacy Threats

Suppose a data holder wants to publish an object-specific path table T to some recipient(s) for data analysis. Explicit identifiers, e.g., name, SSN, and EPC, have been removed. The path, together with the object-specific attributes, are assumed to be important for the task of data analysis; otherwise, they should be removed. One recipient, the adversary, seeks to identify the record or sensitive values of some target victim V in T. As explained in Section 1, we assume that the adversary knows at most L pairs of location and timestamp that the victim V has previously visited. We use $\kappa = \langle (loc_1 t_1) \rightarrow \ldots \rightarrow (loc_z t_z) \rangle$ to denote such prior knowledge, where $z \leq L$. Using the prior knowledge κ, the adversary could identify a group of records in T, denoted by $G(\kappa)$, that "matches" κ. A record *matches* κ if κ is a subsequence of the path in the record. For example in Table 1, if $\kappa = \langle e4 \rightarrow c7 \rangle$, then $EPC\#1$ $[\langle a1 \rightarrow d2 \rightarrow b3 \rightarrow e4 \rightarrow f6 \rightarrow c7 \rangle : HIV]$ matches κ, but $EPC\#4$ $[\langle d2 \rightarrow f6 \rightarrow c7 \rightarrow e8 \rangle : Allergy]$ does not. An adversary could utilize $G(\kappa)$ to perform two types of privacy attacks:

1. *Record linkage*: $G(\kappa)$ is a set of candidate records that contains the victim V's record. If the group size of $G(\kappa)$, denoted by $|G(\kappa)|$, is small, then the adversary may identify V's record from $G(\kappa)$, and therefore, V's sensitive value.

2. *Attribute linkage*: Given $G(\kappa)$, the adversary may infer that V has sensitive value s with confidence

$$Conf(s|G(\kappa)) = \frac{|G(\kappa \bigcup s)|}{|G(\kappa)|},$$

where $G(\kappa \bigcup s)$ denotes the set of records containing both κ and s. $Conf(s|G(\kappa))$ is the percentage of the records in $G(\kappa)$ containing s. The privacy of V is at risk if $Conf(s|G(\kappa))$ is high.

Example 1.1 illustrates these two types of attacks.

3.3 Privacy Models

The problem studied in this paper is to transform the raw object-specific path table T to a version T' that is immunized against record and attribute linkages. We define two separate privacy models *LK-anonymity* and *LC-dilution* to thwart record linkages and attribute linkages, respectively, followed by a unified model. The adversary's prior knowledge κ could be any subsequence q with a maximum length L of any path in T.

Definition 3.1 (LK-anonymity). An object-specific path table T *satisfies LK-anonymity* if and only if $|G(q)| \geq K$ for any subsequence q with $|q| \leq L$ of any path in T, where K is a positive anonymity threshold. ■

Definition 3.2 (LC-dilution). Let S be a set of data holder-specified sensitive values from sensitive attributes $S_1, \ldots, S_m$. An object-specific path table T *satisfies LC-dilution* if and only if $Conf(s|G(q)) \leq C$ for any $s \in S$ and for any subsequence q with $|q| \leq L$ of any path in T, where $0 \leq C \leq 1$ is a confidence threshold. ■

Definition 3.3 (LKC-privacy). An object-specific path table T *satisfies LKC-privacy* if T satisfies both LK-anonymity and LC-dilution. ■

LK-anonymity bounds the probability of a successful record linkage to $\leq 1/K$. LK-dilution bounds the probability of a successful attribute linkage to $\leq C$. LKC-privacy bounds both. Note, not all values in sensitive attributes $S_1, \ldots, S_m$ are sensitive. For example, HIV could be sensitive, but flu may not be. Our proposed privacy model is flexible to accommodate different privacy need by allowing the data holder to specify a set of sensitive values S in Definition 3.2.

3.4 Problem Statement

We can transform an object-specific path table T to satisfy LKC-privacy by performing a sequence of suppressions on selected pairs from T. 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. We use Sup to denote the set of suppressed pairs. Table 2 is the result of suppressing $a1$, $d2$, and $e4$ from Table 1. This suppression scheme offers several advantages over generalization for anonymizing RFID data. First, it 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 generalization on RFID data may result in generalizing many "neighbor" objects even if there is only a small number of outlier pairs, such as $a1$ in Table 1. Suppression offers the flexibility of removing those outliers without affecting the rest of the data.

Definition 3.4 (Anonymization for RFID). Given an object-specific path table T a LKC-privacy requirement, and a set of sensitive values S, the problem of *anonymization for RFID* is to identify a transformed version T' that satisfies the LKC-privacy requirement by suppressing a minimal number of instances of pairs in T. ■

K-anonymity [16] is a special case of LKC-privacy with $L = \infty$ and $C = 100\%$. Confidence bounding [22] is a special case LKC-privacy with $L = \infty$ and $K = 1$. Given that achieving optimal K-anonymity and optimal confidence bounding have been proven to be NP-hard [15][22], achieving optimal LKC-privacy is also NP-hard. Thus, we propose a greedy algorithm to efficiently identify a sub-optimal solution.

4. ANONYMIZATION METHOD

Given an object-specific path table T and a LKC-privacy requirement, our goal is to remove all "violations" from T, where a *violation* is a subsequence of a path in T that violates the LKC-privacy requirement. We first define the notion of violation in Section 4.1 followed by a greedy algorithm in Section 4.2 to remove all violations.

4.1 Identifying Violations

A subsequence q in T is a *violation* if its length is less than maximum length threshold L and its group $G(q)$ violates LK-anonymity, LC-dilution, or both. The adversary's prior knowledge κ could be any of such subsequence q. Thus, removing all violations means eliminating all possible channels of record and attribute linkage attacks.

Definition 4.1 (Violation). Let q be a subsequence of a path in T with $|q| \leq L$ and $|G(q)| > 0$. q is a *violation* with respect to a LKC-privacy requirement if $|G(q)| < K$ or $Conf(s|G(q)) > C$. ■

Example 4.1. In Table 1, a sequence $q_1 = \langle e4 \rightarrow c7 \rangle$ is a violation if $K = 2$ because $|G(q_1)| = 1 < 2$. A sequence $q_2 = \langle d2 \rightarrow f6 \rangle$ is a violation if $C = 50\%$ and $S = \{HIV\}$ because $Conf(HIV|G(q_2)) = 67\% > 50\%$. ■

We note two properties in the notion of violation. (1) If q is a violation with $|G(q)| < K$, then any super sequence of q, denoted by q', is also a violation because $|G(q')| \leq |G(q)| < K$. This property has two implications. First, it implies that the number of violations could be huge, so it is not feasible to first generate all violations and then remove them. Second, if $L \leq L'$, a table T satisfying $L'K$-anonymity must satisfy LK-anonymity because $|G(q)| \geq |G(q')| \geq K$. (2) If q is a violation with $Conf(s|G(q)) > C$ and $|G(q)| \geq K$, its super sequence q' may or may not be a violation because $Conf(s|G(q')) \geq Conf(s|G(q))$ does not always hold. Thus, to achieve LC-dilution, it is insufficient to ensure any subsequence q with length L in T to satisfy $Conf(s|G(q)) \geq C$. Instead, we need to ensure any subsequence q with length less than or equal to L in T to satisfy $Conf(s|G(q)) \geq C$.

Enumerating all possible violations is infeasible. Our insight is that among all the violations, there exists some minimal sequences called "critical violations." We show that a violation exists in table T if and only if a critical violation exists in T.

Definition 4.2 (Critical violation). A violation q is a *critical violation* if every proper subsequence of q is a non-violation. ■

Example 4.2. In Table 1, if $K = 2$, $C = 50\%$, $S = \{HIV\}$, a sequence $q_1 = \langle e4 \rightarrow c7 \rangle$ is a critical violation because $|G(q_1)| = 1 < 2$, and both $\langle e4 \rangle$ and $\langle c7 \rangle$ are non-violations. A sequence $q_2 = \langle d2 \rightarrow e4 \rightarrow c7 \rangle$ is a violation but it is a not a critical violation because its subsequence $\langle e4 \rightarrow c7 \rangle$ is a violation. ■

Observation 4.1. A table T' satisfies LKC-privacy if and only if T' contains no critical violation because each violation is a super sequence of a critical violation. Thus, if T' contains no critical violations, then T' contains no violations. ■

Next, we propose an algorithm to efficiently identify all critical violations in T with respect to a LKC-privacy requirement. Based on Definition 4.2, we generate all critical violations of size $i + 1$, denoted by V_{i+1}, by incrementally extending non-violations of size i, denoted by U_i, with an additional pair.

Algorithm 1 summarizes the steps for generating critical violations. Line 1 initializes the candidate set C_1 to be the set of all distinct pairs in any paths in the raw table T. Line 4 scans the raw data once to obtain the support counts to compute $|G(q)|$ and $Conf(s|G(q))$ for every sequence $q \in C_i$ and for every sensitive value $s \in S$. Lines 5-13 loops through every candidate $q \in C_i$ of $|G(q)| > 0$, and puts q to the critical violation set V_i if it violates LK-anonymity or LC-dilution; otherwise, puts q to the non-violation set U_i. Once a violation is found, we remove it from subsequent iterations because its super sequence must not be a critical violation. Line 15 generates a candidate set C_i by self-joining U_{i-1}. Two sequences $q_x = \langle (loc_1^x t_1^x) \rightarrow \ldots \rightarrow (loc_{i-1}^x t_{i-1}^x) \rangle$ and $q_y = \langle (loc_1^y t_1^y) \rightarrow \ldots \rightarrow (loc_{i-1}^y t_{i-1}^y) \rangle$ in U_{i-1} can be joined only if the first $i - 2$ pairs of q_x and q_y are identical and $t_{i-1}^x < t_{i-1}^y$. The joined sequence is $\langle (loc_1^x t_1^x) \rightarrow \ldots \rightarrow (loc_{i-1}^x t_{i-1}^x) \rightarrow (loc_{i-1}^y t_{i-1}^y) \rangle$. Lines 16-20 removes a candidate q from C_i if q is a super sequence of any sequence in V_{i-1} because all proper subsequences of a critical violation must be a non-violation.

Algorithm 1 Generate Critical Violations (GenViolations)

```
Input: Raw RFID path table $T$
Input: Thresholds $L$, $K$, and $C$.
Input: Sensitive values $S$.
Output: Critical violations $V$.
 1: let candidate set $C_1$ be the set of all distinct pairs in $T$;
 2: $i = 1$;
 3: repeat
 4:     scan $T$ once to obtain $|G(q)|$ and $Conf(s|G(q))$ for every
        sequence $q \in C_i$ and for every sensitive value $s \in S$;
 5:     for all sequence $q \in C_i$ do
 6:         if $|G(q)| > 0$ then
 7:             if $|G(q)| < K$ or $Conf(s|G(q)) > C$ for any $s \in S$
                then
 8:                 add $q$ to $V_i$;
 9:             else
10:                 add $q$ to $U_i$;
11:             end if
12:         end if
13:     end for
14:     $++i$;
15:     generate candidate set $C_i$ by $U_{i-1} \bowtie U_{i-1}$;
16:     for all sequence $q \in C_i$ do
17:         if $q$ is a super sequence of $v$ for any $v \in V_{i-1}$ then
18:             remove $q$ from $C_i$;
19:         end if
20:     end for
21: until $i > L$ or $C_i = \emptyset$
22: return $V = V_1 \bigcup \ldots \bigcup V_{i-1}$;
```

Example 4.3. Consider Table 1 with $L = 2$, $K = 2$, $C = 50\%$, and $S = \{HIV\}$. First, we generate candidate set $C_1 = \{a1, d2, b3, e4c5, f6, c7, e8, e9\}$, which is a set of distinct pairs in T. Then, we scan Table 1 to identify the critical violations from C_1 and put them in $V_1 = \{a1\}$. The remaining sequences are non-violations $U_1 = \{d2, b3, e4, c5, f6, c7, e8, e9\}$. Next, we generate $C_2 = \{d2b3, d2e4, d2c5, d2f6, d2c7, d2e8, d2e9, b3e4, b3c5, b3f6, b3c7, b3e8, b3e9, e4c5, e4f6, e4c7, e4e8, e4e9, c5f6, c5c7, c5e8, c5e9, f6c7, f6e8, f6e9, c7e8, c7e9, e8e9\}$ and scan once Table 1 to determine critical violations $V_2 = \{d2b3, d2e4, d2f6, d2e8, d2e9, e4c7, e4e8\}$.

4.2 Anonymization Algorithm

We propose a greedy algorithm to transform raw table T to an anonymous table T' with respect to a given LKC-privacy requirement by a sequence of suppressions. In each

Algorithm 2 RFID Data Anonymizer

```
Input: Raw RFID path table $T$
Input: Thresholds $L$, $K$, and $C$.
Input: Sensitive values $S$.
Output: Anonymous $T'$ that satisfies $LKC$-privacy.
 1: $V$ = Call GenViolations($T$, $L$, $K$, $C$, $S$) in Algorithm 1;
 2: build the Critical Violation Tree (CVT) with Score Table;
 3: while Score Table is not empty do
 4:     select winner pair $w$ that has the highest $Score$;
 5:     delete all critical violations containing $w$ in CVT;
 6:     update $Score$ of a candidate $x$ if both $w$ and $x$ were contained in the same critical violation;
 7:     remove $w$ in Score Table;
 8:     add $w$ to $Sup$;
 9: end while
10: for every $w \in Sup$, suppress all instances of $w$ from $T$;
11: return the suppressed $T$ as $T'$;
```

iteration, the algorithm selects a suppression on value v based on a greedy selection function. In general, a suppression on a value v in T increases privacy because it removes critical violations, and decreases information utility because it suppresses pairs in T. Therefore, we define the greedy function, $Score(p)$, to select a suppression on a pair p that maximizes the number of critical violations removed and minimizes the number of pair instances suppressed in T. $Score(p)$ is formally defined as follows:

$$Score(p) = \frac{PrivGain(p)}{InfoLoss(p)}, \tag{1}$$

where $PrivGain(p)$ is the number of critical violations containing pair p and $InfoLoss(p)$ is the number of instances of pair p in T. Alternative greedy functions could be

$$Score(p) = PrivGain(p), \tag{2}$$

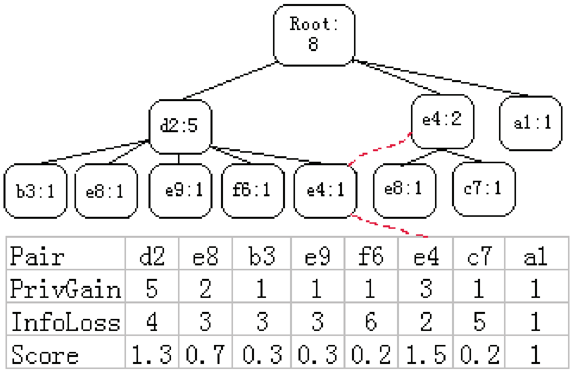
which aims at eliminating all critical violations but ignores the information loss caused by the suppression, or

$$Score(p) = \frac{1}{InfoLoss(p)}, \tag{3}$$

which aims at minimizing the number of suppressed instances in T but ignores how many critical violations can be removed by the suppression. In Section 5, we will evaluate the performance of these variations.

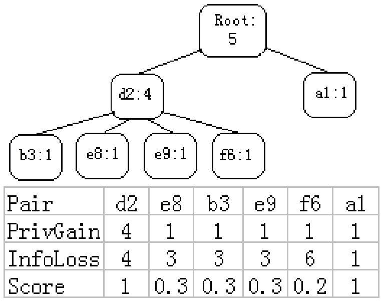
Algorithm 2 summarizes the RFID data anonymization algorithm. Lines 1-2 call Algorithm 1 to generate all critical violations and build a tree to represent them. At each iteration in Lines 3-9, the algorithm selects the winner pair w that has the highest $Score(p)$ among all candidates for suppression, removes the critical violations containing w, and incrementally updates the $Score$ of the affected candidates due to the suppression on w. Sup denotes the set of all suppressed winner pairs. They 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 to identify the critical violations containing w and to update the $Score$ of the affected candidates. Below, we propose a data structure called *critical violation tree (CVT)* to efficiently support these operations.

Definition 4.3 (Critical Violation Tree (CVT)). CVT is a tree structure that represents each critical violation as a tree path from root-to-leaf. Each node keeps track of a count of critical violations sharing the same prefix. The count at the root is the total number of critical violations. CVT has a Score Table that maintains every candidate pair p for suppression, together with its $PrivGain(p)$, $InfoLoss(p)$, and $Score(p)$. Each candidate pair p in the Score Table has a link, denoted by $Link_p$, that links up all the nodes in CVT containing p. $PrivGain(p)$ is the sum of the counts of critical violations on $Link_p$. ■



Pair	d2	e8	b3	e9	f6	e4	c7	a1
PrivGain	5	2	1	1	1	3	1	1
InfoLoss	4	3	3	3	6	2	5	1
Score	1.3	0.7	0.3	0.3	0.2	1.5	0.2	1

Figure 2: Initial Critical Violation Tree (CVT)



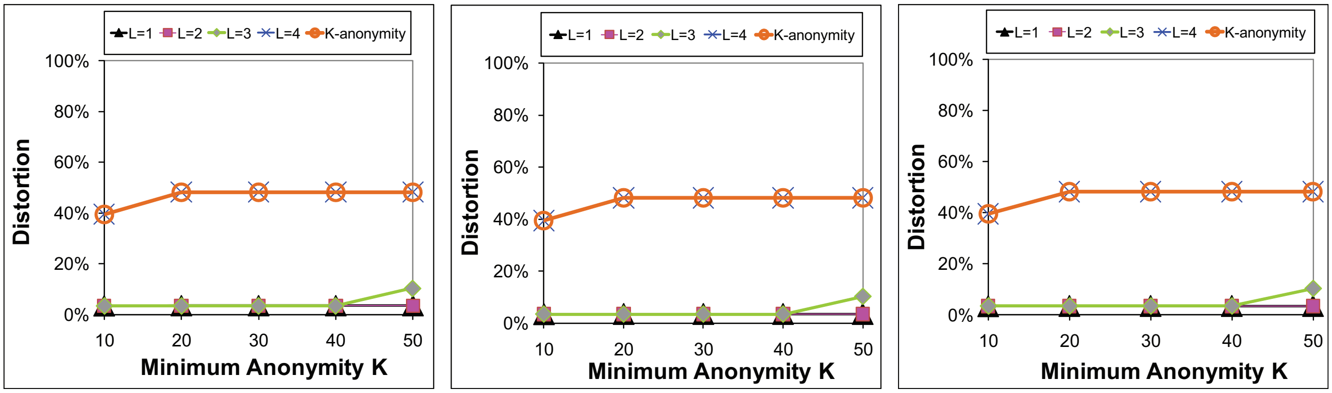
Pair	d2	e8	b3	e9	f6	a1
PrivGain	4	1	1	1	1	1
InfoLoss	4	3	3	3	6	1
Score	1	0.3	0.3	0.3	0.2	1

Figure 3: CVT after suppressing $e4$

Figure 2 depicts the initial CVT generated from V_1 and V_2 in Example 4.3. The winner pair $e4$, which has the highest $Score$, is identified from the Score Table. Then, the algorithm traverses $Link_{e4}$ to identify all critical violations containing $e4$ and deletes them from CVT accordingly. When a winner pair w is suppressed from CVT, the entire branch of w is trimmed. This provides an efficient method for removing critical violations. In Figures 2 and 3, when $e4$ is suppressed, all its descendants are removed as well. The count of critical violations of $e4$'s ancestor nodes is decremented by the count of critical violations of the deleted $e4$ node. If a candidate pair p and the winner pair w are contained in some critical violation, then $PrivGain(p)$, and therefore $Score(p)$, has to be updated for adding up the counts on $Link_p$. For example, after $e4$ is suppressed, $PrivGain(d2)$, $PrivGain(c7)$, and $PrivGain(e8)$ have to be updated. A pair p with $PrivGain(p) = 0$ in Score Table is removed.

5. EMPIRICAL STUDY

We evaluated the performance of our proposed method in terms of data quality after anonymization, efficiency of anonymization, and scalability for handling large data set. All experiments were conducted on a PC with Intel Core2 Quad 2.4GHz with 2GB of RAM. Unless otherwise specified, all experiments on our proposed method use Equation 1 as the $Score$ function.



(a) $C = 20\%$ (b) $C = 60\%$ (c) $C = 100\%$

Figure 4: Distortion ratio vs. K

The employed data set is a simulation of the travel route of 20,000 passengers in a subway transit system with 26 stations for 24 hours. Each record in the object-specific path table corresponds to the route of one passenger. There are 26 × 24 possible pairs, forming 624 dimensions. To simulate different traveling patterns, 16,000 passengers have a maximum path length of 4 pairs, 3,500 passengers have a maximum path length of 6 pairs, and 500 passengers have a maximum path length of 24 pairs. Each record contains a sensitive attribute with 5 possible values. We considered one of them, namely HIV, to be sensitive in our experiments.

5.1 Distortion

Our first experiment is to measure the data quality of the LKC-privacy protected table T'. We use distortion ratio to measure the information loss caused by suppression. Let $N(T)$ and $N(T')$ be the total number of pair instances in tables T and T', respectively. The *distortion ratio*, computed by $\frac{N(T)-N(T')}{N(T)}$, measures the percentage of pair instances suppressed for achieving a given LKC-privacy requirement. Higher distortion ratio means lower data quality. We also compare our method with the traditional K-anonymization.

Figure 4 depicts the distortion ratio of our method for maximum length $1 \leq L \leq 3$ for anonymity thresholds $10 \leq K \leq 50$ at confidence thresholds $C = 20\%, 60\%, 100\%$, and compares the result with the traditional K-anonymity. In general, the distortion ratio is insensitive to the increase of K and stays between 3% to 10% for $1 \leq L \leq 3$ because this requirement only requires every sequence with a maximum length of 3 to be shared by at least 50 records among 20,000 records. Compared to traditional K-anonymity which consistently stays above 40%, our anonymization method can effectively reduce information loss on high-dimensional data. As L increases to 4, the distortion ratio increases significantly because the majority of records have a path length of 4 pairs. Therefore, setting $L = 4$ yields similar result to traditional K-anonymity. It is also interesting to note that the distortion ratio is insensitive to the change of confidence threshold C, implying that the primary driving force for suppressions is LK-anonymity, not LC-dilution. This fact is also reflected in Figure 4(c) at $C = 100\%$, which is equivalent to ignoring LC-dilution.

The result can be summarized as follows. (1) The distortion ratio is not sensitive to the change of anonymity threshold K unless K is set to an unreasonably high range such as $K > 1000$. (2) The distortion ratio is not sensitive to the change of confidence threshold C. (3) As the maximum length L increases, distortion ratio increases. (4) $Score(p) = \frac{PrivGain(p)}{InfoLoss(p)}$ consistently yields the lowest distortion ratio among the three $Score$ functions given in Section 4.2.

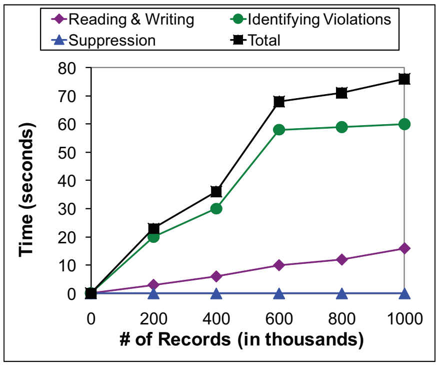


Figure 5: Scalability ($L = 3$, $K = 30$, $C = 60\%$)

5.2 Efficiency and Scalability

Next, we examine the efficiency and scalability of our proposed anonymization method. For all the test cases conducted in Section 5.1, our method takes less than 1 second to complete. In an effort to further evaluate the scalability of our method, we conducted an experiment on some extremely large synthetic RFID data sets.

Figure 5 depicts the runtime in seconds from 200,000 to 1 million records for $L = 3$, $K = 30$, $C = 60\%$. The total runtime for anonymizing 1 million records is 76 seconds, where 60 seconds are spent on identifying critical violations and 16 seconds are spent on reading raw data file and writing anonymous file. Thanks to the effective critical violation tree (CVT) data structure, the program takes less than 1 second on suppressing all violations.

6. CONCLUSION

RFID is a promising technology applicable in many areas, but many of its privacy issues have not yet been addressed. In this paper, we illustrate the privacy threats caused by publishing RFID data, formally define a privacy model, called LKC-privacy, for the high-dimensional, sparse

RFID data, and propose an efficient anonymization algorithm to transform a RFID data set to satisfy a given LKC-privacy requirement. We demonstrate that applying traditional K-anonymity on high-dimensional RFID data would render the data useless due to the curse of high-dimensionality. Experimental result suggests that our method can efficiently anonymize large RFID data sets with significantly better data quality than the traditional K-anonymity method.

7. ACKNOWLEDGEMENTS

The research is supported in part by the Discovery Grants (356065-2008) from the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Faculty Start-up Funds from Concordia University.

8. REFERENCES

[1] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proc. of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 376–385, April 2008.

[2] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proc. of the 31st Very Large Data Bases (VLDB)*, pages 901–909, 2005.

[3] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE)*, pages 217–228, Tokyo, Japan, 2005.

[4] B. C. M. Fung, K. Wang, A. W. C. Fu, and J. Pei. Anonymity for continuous data publishing. In *Proc. of the 11th International Conference on Extending Database Technology (EDBT)*, March 2008.

[5] B. C. M. Fung, K. Wang, L. Wang, and M. Debbabi. A framework for privacy-preserving cluster analysis. In *Proc. of the 2008 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Taipei, Taiwan, June 2008.

[6] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE)*, pages 205–216, Tokyo, Japan, April 2005.

[7] B. C. M. Fung, K. Wang, and P. S. Yu. Anonymizing classification data for privacy preservation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(5):711–725, May 2007.

[8] G. Ghinita, Y. Tao, and P. Kalnis. On the anonymization of sparse high-dimensional data. In *Proc. of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 715–724, April 2008.

[9] H. Gonzalez, J. Han, and X. Li. Flowcube: Constructing RFID flowcubes for multi-dimensional analysis of commodity flows. In *Proc. of the International Conference on Very Large Data Bases (VLDB)*, pages 1–19, Seoul, Korea, September 2006.

[10] H. Gonzalez, J. Han, and X. Li. Mining compressed commodity workflows from massive rfid data sets. In *Proc. of the International Conference on Information and Knowledge Management (CIKM)*, November 2006.

[11] A. Juels. Rfid security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381– 394, February 2006.

[12] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proc. of ACM SIGMOD*, pages 49–60, Baltimore, ML, 2005.

[13] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proc. of the 21st IEEE International Conference on Data Engineering (ICDE)*, Istanbul, Turkey, April 2007.

[14] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *Proc. of the 22nd IEEE International Conference on Data Engineering (ICDE)*, 2006.

[15] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proc. of the 23rd ACM PODS*, pages 223–228, Paris, France, 2004.

[16] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proc. of the 17th ACM PODS*, page 188, June 1998.

[17] S. E. Sarma, S. A. Weis, and D. W. Engels. Rfid systems and security and privacy implications. In *Proc. of the 4th International Workshop of Cryptographic Hardware and Embedded Systems (CHES)*, pages 1–19, San Diego, 2003.

[18] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proc. of the 9th International Conference on Mobile Data Management (MDM)*, pages 65–72, April 2008.

[19] M. Terrovitis, N. Mamoulis, and P. Kalnis. Anonymity in unstructured data. Technical Report TR-2004-04, Department of Computer Science, University of Hong Kong, April 2008.

[20] K. Wang and B. C. M. Fung. Anonymizing sequential releases. In *Proc. of the 12th ACM SIGKDD*, Philadelphia, PA, August 2006.

[21] K. Wang, B. C. M. Fung, and P. S. Yu. Template-based privacy preservation in classification problems. In *Proc. of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 466–473, Houston, TX, November 2005.

[22] K. Wang, B. C. M. Fung, and P. S. Yu. Handicapping attacker's confidence: An alternative to k-anonymization. *Knowledge and Information Systems (KAIS)*, 11(3):345–368, April 2007.

[23] 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 *Proc. of the 39th Hawaii International Conference on System Sciences*, 2006.

[24] R. C. W. Wong, J. Li., A. W. C. Fu, and K. Wang. (α,k)-anonymity: An enhanced k-anonymity model for privacy preserving data publishing. In *Proc. of the 12th ACM SIGKDD*, 2006.

[25] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proc. of ACM SIGMOD*, Chicago, IL, 2006.

[26] Y. Xu, B. C. M. Fung, K. Wang, A. W. C. Fu, and J. Pei. Publishing sensitive transactions for itemset utility. In *Proc. of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, Pisa, Italy, December 2008. IEEE Computer Society.

[27] Y. Xu, K. Wang, A. W. C. Fu, and P. S. Yu. Anonymizing transaction databases for publication. In *Proc. of the 14th ACM SIGKDD*, August 2008.