JS-Reduce: Defending Your Data from Sequential Background Knowledge Attacks

Daniele Riboni, Linda Pareschi, and Claudio Bettini, *Member*, *IEEE Computer Society*

Abstract—Web queries, credit card transactions, and medical records are examples of transaction data flowing in corporate data stores, and often revealing associations between individuals and sensitive information. The serial release of these data to partner institutions or data analysis centers in a nonaggregated form is a common situation. In this paper, we show that correlations among sensitive values associated to the same individuals in different releases can be easily used to violate users' privacy by adversaries observing multiple data releases, even if state-of-the-art privacy protection techniques are applied. We show how the above sequential background knowledge can be actually obtained by an adversary, and used to identify with high confidence the sensitive values of an individual. Our proposed defense algorithm is based on Jensen-Shannon divergence; experiments show its superiority with respect to other applicable solutions. To the best of our knowledge, this is the first work that systematically investigates the role of sequential background knowledge in serial release of transaction data.

Index Terms—Privacy-preserving release of transaction data, anonymity, sequential background knowledge.

1 Introduction

Large amounts of data related to individuals are continuously acquired, and stored by corporate and government institutions. Examples include mobile service requests, web queries, credit card transactions, and transit database records. These institutions often need to repeatedly release new or updated portions of their data to other partner institutions for different purposes, including distributed processing, participation in interorganizational workflows, and data analysis. The medical domain is an interesting example: many countries have recently established centralized data stores that exchange patients' data with medical institutions; new records are periodically released to data analysis centers in nonaggregated form.

A challenging issue in this scenario is the protection of users' privacy, considering that potential adversaries have access to multiple serial releases and can easily acquire background knowledge related to the specific domain. This knowledge includes the fact that certain sequences of values in subsequent releases are more likely to be observed than other sequences. For example, it is pretty straightforward to extract from the medical literature or from a public data set that a sequence of medical exam results within a certain time frame has higher probability to be observed than another sequence.

Privacy protection approaches can be divided in microdata anonymity and differential privacy methods.

Microdata anonymity works have focused on techniques dealing either with multiple data releases, or with adversary background knowledge, but limited to a single data release. We are not aware of any work taking into account the combination of these conditions. This case cannot be addressed by simply combining the two types of techniques mentioned above, since background knowledge can enable new kinds of privacy threats on sequential data releases. Extensions of data anonymization techniques to deal with multiple data releases have been proposed under different assumptions (e.g., [2], [4], [14], [25], [26]). However, our running example in Section 2 shows that those techniques are not effective when an adversary can obtain background knowledge on the transition probabilities of sensitive values. The work that is closest to ours is probably the one in [2], in which sensitive values are divided in *transient* values that may freely change with time, and *persistent* values that never change. However, that technique is effective only when the transition probability among transient values is uniform, and this is often not the case, with the medical domain being a clear counterexample. On the contrary, our privacy preserving technique captures nonuniform transition probabilities. Techniques considering background knowledge have also been proposed (e.g., in [6], [16], [17]); however, they are devised for a single release of the data, and, as shown in Section 7, they are ineffective when an adversary having background knowledge on sequences of sensitive values may observe multiple releases.

Differential privacy methods [7] have been proposed for privacy-preserving query answering over statistical databases. Those methods are supported by a rigorous and quantifiable notion of privacy; however, recent studies showed that their effectiveness is also subject to assumptions about the background knowledge available to an adversary [12]. In particular, even if current methods for differential privacy are formally applied, the presence of correlations about sensitive values of different tuples in the data set may allow an adversary to reconstruct the private values of some respondents. In Section 3.3, we discuss this issue, showing that existing differential privacy methods cannot be applied to our scenario.

• *The authors are with the Dipartimento di Informatica e Comunicazione (DICo), Università degli Studi di Milano, Via Comelico 39, Milano I-20135, Italy. E-mail: {riboni, pareschi}@dico.unimi.it, claudio.bettini@unimi.it.*

Manuscript received 14 Oct. 2011; revised 13 Jan. 2012; accepted 18 Jan. 2012; published online 26 Jan. 2012.
Recommended for acceptance by E. Ferrari and B. Thuraisingham.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-2011-10-0233.
Digital Object Identifier no. 10.1109/TDSC.2012.19.

In this paper, we formally model privacy attacks based on background knowledge extended to serial microdata releases. We present a new probabilistic defense technique taking into account adversary's background knowledge and how he can revise it each time new data are released. Similarly to other anonymization techniques, our method is based on the generalization of quasi-identifier (QI) values, but generalization is performed with a new goal: minimizing the difference among sensitive values probability distributions within each QI-group, while considering the knowledge revision process. Jensen-Shannon divergence is used as a measure of similarity. We consider different methods and accuracy levels for the extraction of background knowledge, and we show that our defense is effective under different combinations of the knowledge of the adversary and the defender.

Contributions. The contributions of this paper can be summarized as follows:

1. We model privacy attacks on sequential data release based on background knowledge about the probability distributions of sensitive values and sequences of sensitive values. We show that current anonymization techniques are not resistant to these privacy attacks.
2. We propose *JS-reduce* as a new probabilistic defense technique based on Jensen-Shannon divergence.
3. Through an experimental evaluation on a large data set, we show the effectiveness of our defense under different methods used to extract background knowledge; our results also show that JS-reduce provides a very good tradeoff between privacy and data utility.

Open challenges. As any other anonymization technique, JS-reduce is based on a specific adversary model. Hence, an open research issue is how much the abilities of the adversary should be constrained, and how realistic these constraints are. Our work significantly extends the state of the art by observing that a form of temporal knowledge that can be easily extracted by an adversary defeats current defenses. However, we emphasize that, while JS-reduce provides a defense against that particular adversary model, it does not provide any formal guarantee of defense against arbitrary background knowledge and attacks; as explained above, this is an open challenge even for differential privacy approaches.

Paper outline. In Section 2, we illustrate the privacy problem through an example in the medical domain, and we show the inadequacy of state-of-the-art techniques. In Section 3, we present relevant related work. In Section 4, we formally model the privacy attack, as well as the considered forms of background knowledge. In Section 5, we show how an adversary can actually extract background knowledge, and revise his knowledge in order to perform the attack. In Section 6, we propose our JS-reduce defense algorithm that is experimentally evaluated in Section 7. Section 8 concludes the paper.

2 Motivating Scenario

In this section, we focus on a specific scenario in the medical domain to illustrate the privacy attacks enabled by background knowledge on sequences of sensitive values. The running example also shows the inadequacy of state-of-the-art techniques. Table 2 reports a quick reference to notation.

We consider the case of transaction data representing the results of medical exams taken by patients, and the need to periodically release these transactions for data analysis. Each released view contains one tuple for each patient who performed an exam during the week preceding the publication. We assume that data are published weekly. For the sake of simplicity, we also assume that each user cannot perform more than one exam per week; hence, no more than one tuple per user can appear in the same view. Each generalized tuple includes the age, gender, and zip code of the patient, as well as the performed exam together with its result. We refer to this latter data, represented by the multivalue attribute *Ex-res*, as *exam result*.[1] We denote as positive (*pos*) a result that reveals something anomalous; negative (*neg*) otherwise. The attribute Ex-res is considered the *sensitive attribute*, while the other attributes play the role of *quasi identifiers* (*QI*), since they may be used, joined with external information, to restrict the set of candidate respondents. We consider the realistic case in which the adversary's background knowledge includes both *sensitive values background knowledge* (BK^{sv}) and *sequential background knowledge* (BK^{seq}). Intuitively, BK^{sv} regards the probability of performing an exam with a given result based on data such as patient's gender, age, and ZIP code, e.g., "middle-aged females have a sensible probability to undergo a mammography with a positive result (MAM-pos), while teenagers do not." BK^{seq} regards the probability of a patient's exam result given the previous exam results. For instance, "when the mammography signals a possible malignancy (MAM-pos) for patient r, there is high probability that a blood sample of r examined within a month would detect a breast cancer marker (BCM-pos)." A simple form of BK^{seq} is reported in Table 3b; in particular, the first row in the table represents the above statement, where the probability of the event is set to 0.6. As we show in Section 5.1, both sequential and sensitive values background knowledge can be easily acquired, either through the scientific literature or from the data. We name *posterior knowledge* (PK^{sv}) *at* τ_i the adversary's confidence about the exam results of tuples respondents after observing the data released at time τ_i (e.g., "The probability that Alice is the respondent of a tuple with $\text{Ex-res} = \text{MAM-pos}$ released at τ_1 is 0.5").

Consider the original transaction data at time τ_1 (first week) and τ_2 (second week) shown in Tables 1a and 1c, respectively, and the corresponding generalized transaction data in Tables 1b and 1d. Note that these generalized views satisfy state-of-the-art techniques for privacy preservation. In particular, they satisfy l-diversity [19] with $l = 2$, m-invariance[2] [26] with $m = 2$, as well as the privacy properties proposed in [2]. However, we show that the release of these views can lead to a serious privacy threat. Consider tuples released at τ_1 belonging to QI-group 1, having private values MAM-pos and CX-neg, whose possible respondents are Alice and Betty. Since Alice and Betty are almost the same age, and live in the same area, the

1. MAM = mammography, CX = chest X-ray, BCM = breast cancer marker, PNE = pneumonia.

2. Note that even though the signatures of QI-groups in first and second release are different, m-invariance is not violated, since the sensitive values of group respondents in the first release are different from the ones in the second release.

TABLE 1
Original and Generalized Transaction Data at the First and Second Release (First and Second Week, Respectively)

(a) Original transaction data at time τ_1

Name	Age	Gender	Zip	Ex-res
Alice	51	F	12030	MAM-pos
Betty	52	F	12030	CX-neg
Carol	51	F	12031	CX-pos
Doris	52	F	12031	BS-neg

(b) Generalized transaction data: 1st release

QI-group	Age	Gender	Zip	Ex-res
1	[51,52]	F	12030	MAM-pos
1	[51,52]	F	12030	CX-neg
2	[51,52]	F	12031	CX-pos
2	[51,52]	F	12031	BS-neg

(c) Original transaction data at time τ_2

Name	Age	Gender	Zip	Ex-res
Alice	51	F	12030	BCM-pos
Carol	51	F	12031	PNE-pos
Elisa	51	F	12044	MAM-neg
Fran	51	F	12045	CX-neg
Grace	51	F	12040	CX-pos

(d) Generalized transaction data: 2nd release

QI-group	Age	Gender	Zip	Ex-res
3	51	F	1203*	BCM-pos
3	51	F	1203*	PNE-pos
4	51	F	1204*	MAM-neg
4	51	F	1204*	CX-neg
4	51	F	1204*	CX-pos

adversary cannot exploit BK^{sv} (reported in Table 3b) to infer whether Alice or Betty is the respondent of the tuple with value MAM-pos. Hence, his posterior knowledge after observing tuples released at τ_1 states that, both for Alice and Betty, the probability of being the respondent of one tuple with private value MAM-pos is the same of being the respondent of one tuple with private value CX-neg, i.e., 0.5. Analogously, Carol and Doris have equal probability of being the respondent of one tuple with private value CX-pos and of one with private value BS-neg.

Now, consider tuples released at τ_2 (in Table 1d) belonging to QI-group 3, having private values BCM-pos and PNE-pos, whose possible respondents are Alice and Carol. Since Alice and Carol are the same age, and live in very close areas, once again the adversary cannot exploit BK^{sv} to infer whether Alice's private value is BCM-pos and Carol's one is PNE-pos, or vice-versa. However, the adversary may exploit PK^{sv} at τ_1 and BK^{seq} to derive a new kind of knowledge, which we name *revised sensitive values background knowledge* (RBK^{sv}) at τ_2. This knowledge represents the revision of sensitive values background knowledge computed on the basis of the history of released views, and of sequential background knowledge. The actual method for computing RBK^{sv} is shown in Section 5; here we give an intuition of the adversary reasoning. Since the exam result of Alice at τ_1 is either MAM-pos or CX-neg, and the one at τ_2 is either BCM-pos or PNE-pos, four possible sequences of sensitive values about Alice exist. Among these sequences, according to BK^{seq}, the one having MAM-pos at τ_1 and BCM-pos at τ_2 is more probable than the others, since a positive mammography result is frequently followed by a positive breast cancer marker test. Analogously, among the possible sequences regarding Carol, the most probable is the one having CX-pos at τ_1 and PNE-pos at τ_2. Through this reasoning, the adversary revises his sensitive values background knowledge, associating high confidence to the fact that at τ_2 Alice is positive to breast cancer markers, while Carol has pneumonia. Hence, based on RBK^{sv}, the adversary can assign with high confidence the correct sensitive values to Alice and Carol.

TABLE 2
Summary of Notation Used in the Paper

V_i (resp. V_i^*)	original (resp. generalized) view at time τ_i
$\mathcal{H}_j^* = \langle V_1^*, \ldots, V_j^* \rangle$	history of released generalized views
BK^{sv}	sensitive values background knowledge
BK^{seq}	sequential background knowledge
PK_i^{sv}	posterior knowledge at τ_i
RBK_i^{sv}	revised sensitive values backgr. knowl. at τ_i
IE-BK^{seq}	BK^{seq} extracted from the original data
SPM-BK^{seq}	BK^{seq} mined from an available dataset
DK-BK^{seq}	BK^{seq} extracted from domain knowledge

3 Related Work

Many research efforts have been made to cope with privacy issues when either 1) transaction data are incrementally released, or 2) an adversary has prior knowledge about the probability of association between user identities and released tuples. However, up to the time of writing, these problems have been considered separately. A different approach, based on the notion of differential privacy, has also been proposed for privacy-preserving data mining. In this section, we review the most prominent proposals, showing that they are insufficient to protect against the attacks identified in this paper.

3.1 Incremental Release of Transaction Data

Most of the proposed techniques for protecting users' privacy in incremental release of microdata are inspired by privacy models devised for single publications, such as *k-anonymity* [24] and *l-diversity* [19]. In particular, many proposed techniques are aimed at enforcing anonymity and diversity when the same tuples may appear in different views of the same table released at different times. One of the first attempts to address privacy issues in data republication can be found in [3]. That work proposes a technique to preserve a form of diversity when the table is updated by insertions only. The first work to enforce diversity in data republication when both insertions and deletions are allowed is presented by Xiao and Tao in [26], in which the m-invariance property is introduced.

Our considered scenario, in which tuples published at each release correspond to a new event, has some similarity with the publication of data streams. Privacy issues in this area have been firstly formalized in [14], where a privacy

TABLE 3
Adversary's Background Knowledge

(a) Sensitive values background knowledge at τ_1

Name	Age	Gender	Zip	Ex-res	BK^{sv}
Alice	51	F	12030	MAM-pos	0.002
Betty	52	F	12030	MAM-pos	0.002
Alice	51	F	12030	CX-neg	0.05
Betty	52	F	12030	CX-neg	0.05
Carol	51	F	12031	CX-pos	0.0003
Doris	52	F	12031	CX-pos	0.0003
Carol	51	F	12031	BS-neg	0.2
Doris	52	F	12031	BS-neg	0.2
Alice	51	F	12030	BCM-pos	0.001

(b) Sequential background knowledge

Ex-res at τ_1	Ex-res at τ_2	$\widetilde{p}(s_{\tau_2}\|s_{\tau_1})$
MAM-pos	BCM-pos	0.6
CX-neg	BCM-pos	0.02
CX-pos	BCM-pos	0.02
BS-neg	BCM-pos	0.02
MAM-pos	PNE-pos	0.02
CX-neg	PNE-pos	0.08
CX-pos	PNE-pos	0.6
BS-neg	PNE-pos	0.02

preserving solution based on k-anonymity is proposed at the cost of a limited distortion in the publishing order. Work in [4] enforces both k-anonymity and l-diversity, providing strong guarantees on the maximum delay of output data. However, as shown in Section 2, k-anonymity and l-diversity are insufficient when an adversary has sequential background knowledge.

The work that is closest to ours is probably the one presented in [2], in which sensitive values are divided in *transient* values that may freely change with time (for instance, one patient can be hospitalized in January for dyspepsia and in February for a flu), and *persistent* values that never change (e.g., AIDS). However, the proposed technique is effective only when the transition probability among transient values is uniform; this limitation is shared by the work in [25] as well. Of course, the above assumption holds only in specific domains. For instance, in the medical domain, diseases are often subject to characteristic evolutions; hence, the transition probability among transient values is not uniform. Indeed, our running example shows that released views (Table 1) are not safe from privacy attacks based on sequential background knowledge although they satisfy the privacy requirements imposed by that technique. Note that views in Table 1 also satisfy the requirements dictated by other techniques for incremental release of data; in particular, the technique proposed in [26].

3.2 Background Knowledge

A different research direction deals with the extraction, representation, and integration of background knowledge in privacy models. To the best of our knowledge, solutions proposed in this area are limited to the protection of single publications. However, as it is shown in Section 7, techniques devised for single publications are ineffective when an adversary having sequential background knowledge may observe multiple releases.

Proposed models for expressing adversary's background knowledge can be classified in two main categories: 1) models based on logic assertions and rules; and 2) models based on probabilistic tools. As regards the first category, in [20] the authors propose a language for expressing implications among sensitive values of different individuals, and a defense for limiting the disclosure risk with respect to the adversary's amount of information. More recently, a technique for deriving background knowledge (in particular, negative associations) from data has been proposed in [16].

As regards the second category of models, in [6] the adversary's background knowledge is modeled as the probability of associating a private value to an individual given the values of her quasi identifiers. The defense technique proposed in that work aims at producing anonymized views that maximize the entropy of the probability distributions of sensitive values. A similar model is presented in [17], the adversary has sensitive values background knowledge and he updates his belief after the observation of an anonymized view. The proposed defense aims at limiting the adversarial information gain to a given threshold.

3.3 Differential Privacy

Differential privacy [7] has been proposed to provide noisy query answers over statistical databases while enforcing privacy. Generally speaking, differential privacy guarantees that the probability distribution of query answers is the same, irrespective of whether or not a respondent's tuple is present in the database.

Differential privacy has recently received considerable attention, since it provides a rigorous and quantifiable notion of privacy, and it can be applied to several data mining and machine learning tasks. However, privacy protection against arbitrary background knowledge is still an open issue, not only for microdata anonymity approaches, but also for differential privacy methods. Indeed, recent studies demonstrate that differential privacy provides guarantees only under specific assumptions about the adversary's background knowledge [12]. In the following, we show that existing differential privacy methods cannot be applied to our scenario.

Consider the simplest case in which the data set is static, i.e., no tuple is either added to or removed from the data set, and tuples values do not change with time. This is the case considered in almost all of the existing differential privacy literature. It is a particular case of our scenario, in which the data curator releases a single view of the database. Even in this case, as shown in [12], differential privacy may be ineffective if an adversary has background knowledge about the data generation model. In particular, differential privacy is prone to attacks when the sensitive values of different tuples are correlated. In the medical domain, tuples values (e.g., results of medical exams) in the same release may be strongly correlated: for instance, family members may have contracted the same disease, or people living in a restricted area may be affected by an

epidemic. Hence, even if differential privacy is applied, an adversary may reconstruct with high confidence the private values of some respondents by exploiting the correlations.

A very few studies have considered the more complex case in which the data set is dynamic, and updated statistics are periodically released. How to enforce differential privacy under this setting is still an open issue. Up to the time of writing, only techniques to release differentially private statistics about a single counter have been proposed [5], [8]. It is still an open research question whether differential privacy can be applied to more complex data sets, in the presence of sequential background knowledge, without disrupting data utility.

4 Modeling Attacks Based on Background and Revised Knowledge

In this section, we formally model privacy attacks based on background and revised knowledge.

4.1 Problem Definition

We denote by V_i a view on the original transaction data at time τ_i, and by V_i^* the generalization of V_i released by the data publisher. We denote a *history* of released generalized views by $\mathcal{H}_j^* = \langle V_1^*, V_2^*, \ldots, V_j^* \rangle$. We assume that the schema remains unchanged throughout the release history, and we partition the view columns into a set $A^{qi} = \{A_1, A_2, \ldots, A_m\}$ of quasi-identifier attributes, and into a single private attribute S. For simplicity, we assume that the domain of each quasi-identifier attribute is numeric, but our notions and techniques can be easily extended to categorical attributes. Given a tuple t in a view and an attribute A in its schema, $t[A]$ is the projection of tuple t onto attribute A.

Views are generalized by a *generalization function* $G()$ that removes possible explicit identifiers from the original tuples, and generalizes the quasi identifiers. Tuples in V_j^* are partitioned into *QI-groups*, i.e., sets of tuples having the same values for their quasi-identifier attributes. Even if we consider generalization-based anonymity, both our attack model and defense method can be seamlessly applied to bucketization-based techniques.

At each release of a view V_j^*, the goal of an adversary is to reconstruct, with a certain degree of confidence, the *sensitive association* between the identity of a respondent of a tuple t in V_j^* and her sensitive value $t[S]$. The adversary model considered in this paper is based on the following assumptions:

- The generalization function $G()$ is publicly known.
- The adversary may have external information about respondents' personal data. For example, for each QI-group Q, the adversary may know its set of respondents.
- The adversary may observe a history $\mathcal{H}_j^*$ of generalized views.
- The adversary may have background knowledge on sensitive values BK^{sv} and BK^{seq} as formally defined in Sections 4.2 and 4.3, respectively.

Note that the first two assumptions are shared by most work on anonymity. As illustrated in Section 1, the third and the fourth (limited to BK^{sv}) have also been considered by related work but not in combination. Finally, BK^{seq} is original to this work.

4.2 Sensitive Values Background Knowledge (BK^{sv})

Sensitive values background knowledge represents the a priori probability of associating an individual to a sensitive value. We model the sensitive value referring to a respondent r by means of the discrete random variable $\mathcal{S}$ having values in $D[S]$. BK^{sv} is modeled according to the following definition.

Definition 1. *The* sensitive values background knowledge *is a function* $BK^{sv} : R \rightarrow \Upsilon$, *where* R *is the set of possible respondents' identities, and*

$$\Upsilon = \left\{ (p_1, \ldots, p_n) \;\middle|\; \sum_{1 \leq i \leq n} p_i = 1 \; (0 \leq p_i \leq 1) \right\},$$

is the set of possible probability distributions of S, *where* $D[S] = \{s_1, s_2, \ldots, s_n\}$.

For example, if $r \in R$ is a possible respondent of a tuple in a released view, $BK^{sv}(r)$ returns, for each sensitive value $s_j \in D[S]$, the probability p_j of r being actually associated with s_j.

4.3 Sequential Background Knowledge (BK^{seq})

Sequential background knowledge is a function that returns the probability distribution of $\mathcal{S}$ at τ_j given a sequence $\Lambda = \langle s_1, s_2, \ldots, s_{j-1} \rangle$ of past observations at $T = \langle \tau_1, \tau_2, \ldots, \tau_{j-1} \rangle$.

Definition 2. The sequential background knowledge *is a function* $BK^{seq} : \overline{\Lambda} \times \overline{T} \times R \times \mathcal{T} \rightarrow \Upsilon$, *where* $\overline{\Lambda}$ *is the set of possible sequences of past observations of a respondent's sensitive values,* $\overline{T}$ *is the set of possible sequences of time instants at which the observations were taken,* R *is the set of respondents' identities,* $\mathcal{T}$ *is the set of possible time instants, and* Υ *is the set of possible probability distributions of* $\mathcal{S}$.

For example, if $r \in R$ is a possible respondent of a tuple in a released view, and the adversary knows that r has been associated with values s_1, and s_2 at past instants τ_1, τ_2, respectively, then BK^{seq} returns the probability p_j of r being associated with s_j at τ_3, for each possible sensitive value s_j.

4.4 Posterior (PK^{sv}) and Revised Sensitive Values Background Knowledge (RBK^{sv})

As intuitively described in the running example of Section 2, posterior knowledge at τ_i represents the adversary's confidence about the association between a respondent and sensitive values *after* the observation of view V_i^*. For the sake of readability, we denote PK^{sv} *at* τ_i by PK_i^{sv}.

Definition 3. *The* posterior knowledge *is a function* $PK^{sv} : R \times \mathcal{T} \rightarrow \Upsilon$, *where* R *is the set of respondents' identities,* $\mathcal{T}$ *is the set of possible time instants, and* Υ *is the set of possible probability distributions of* $\mathcal{S}$.

A method to compute PK^{sv} is described in Section 5.2.

After observing view V_{j-1}^*, an adversary may exploit posterior knowledge at τ_1, $\tau_2, \ldots, \tau_{j-1}$, together with sequential background knowledge BK^{seq}, to derive new information about the probability distribution of $\mathcal{S}$ at τ_j. We call this information *revised sensitive values background knowledge* at τ_j (denoted as RBK_j^{sv}); it is essentially the revision of sensitive values background knowledge due to

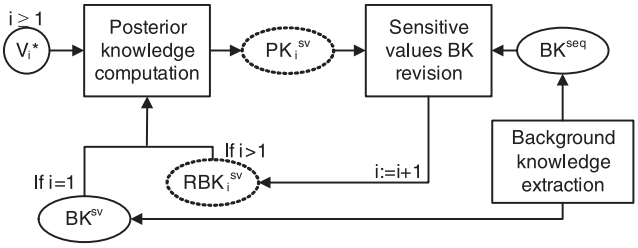


Fig. 1. Adversary's inference mechanisms.

the observation of a history of released tuples. RBK_j^{sv} can be used by an adversary to calculate posterior knowledge after the observation of generalized view V_j^*.

The *revised sensitive values background knowledge* is a function RBK^{sv} having the same domain and codomain as function PK^{sv} defined in Definition 3. The method to compute RBK^{sv} is described in Section 5.3.

4.5 The Privacy Attack

The inference method adopted by an adversary to reconstruct the sensitive association is depicted in Fig. 1. The adversary obtains sensitive values background knowledge BK^{sv}, as well as sequential background knowledge BK^{seq}, using one of the techniques explained in Section 5.1. When the first view V_1^* is released at time τ_1, the adversary computes posterior knowledge PK_1^{sv} based on V_1^* and on BK^{sv}; a method for posterior knowledge computation is illustrated in Section 5.2. Then, the adversary computes revised sensitive values background knowledge RBK_2^{sv}, based on PK_1^{sv} and on sequential background knowledge BK^{seq}. A technique for knowledge revision is illustrated in Section 5.3. Hence, when view V_2^* is released, the adversary computes PK_2^{sv} based on V_2^* and on RBK_2^{sv}. Then, the knowledge revision cycle continues with the computation of RBK_3^{sv} based on PK_2^{sv} and BK^{seq}, and so on. When V_i^* includes a tuple of respondent r, and no tuples of r appeared in $\mathcal{H}_{i-1}^*$, $RBK^{sv}(r, \tau_i)$ cannot be computed, since no historical information about r's tuples is available; in this case BK^{sv} is used instead of $RBK^{sv}(r, \tau_i)$.

5 Knowledge Extraction and Revision

In this section, we illustrate how an adversary may obtain background knowledge, and use it to reconstruct the association between respondents of released tuples and their sensitive values.

5.1 Extracting Background Knowledge

Intuitively, the more accurate is the adversary's background knowledge (i.e., close to the underlying process that generated the data), the more effective will be his attack. Background knowledge can be obtained using different methods, depending on the available data, and on the data domain. The problem of extracting sensitive values background knowledge based on a corpus of available data has been thoroughly studied, and effective techniques are available (e.g., the ones proposed in [6], [17]). Hence, in the rest of this paper we assume that the adversary extracts BK^{sv} using one of the existing methods. However, existing privacy-preserving techniques do not consider the extraction of BK^{seq}. For this reason, we illustrate how this knowledge can actually be obtained.

- *Incrementally extracting BK^{seq} from the data to be released*. One of the methods proposed to compute the background knowledge that an adversary may obtain is to extract it from the same data that are going to be generalized and released, as done in [17]. At the time of writing, these techniques are limited to the calculation of BK^{sv}. However, based on a sequence $\mathcal{H}_i$ of original views, sequential pattern mining (SPM) methods [1] can be used to calculate a function $IE\text{-}BK^{seq}$ that approximates the exact BK^{seq}. In Section 7.3, we illustrate the algorithm we adopt to calculate $IE\text{-}BK^{seq}$ for the sake of our experiments. Of course, this technique can be used by the defender only, since we assume that the adversary cannot observe original views.
- *Mining BK^{seq} from an available corpus of data*. Even if an adversary cannot observe the original data, he may apply SPM methods to a corpus of external data from the same domain to calculate a function $SPM\text{-}BK^{seq}$ that approximates the exact BK^{seq}.
- *Exploiting domain knowledge*. In many cases, it is possible to exploit domain knowledge extracted from the scientific literature. For instance, in the medical domain, a number of surveys have been published, which report accurate statistics about the probability of disease evolution with time. Given this knowledge, it is easy to design a function $DK\text{-}BK^{seq}$, which approximates the exact BK^{seq}.

5.2 Computing Posterior Knowledge

In order to compute PK_i^{sv}, it is possible to reason considering a QI-group at a time. In particular, in our case, given a QI-group Q having R as the set of respondents, a *possible configuration* is a function $c : \mathcal{Q} \rightarrow \mathcal{R}$, i.e., a one-to-one correspondence between elements in $Q \in \mathcal{Q}$ and elements in $R \in \mathcal{R}$. Given a possible configuration c, for each tuple $t \in Q$ we say that "r is the respondent of t in the possible configuration c" if $c(t) = r$.

Example 1. Consider Table 1d released at τ_2 in our running example, and QI-group 3 composed of Alice's and Carol's tuples. In this case, two possible configurations c_1 and c_2 exist. According to c_1, Alice is the respondent of the tuple with sensitive value BCM-pos, and Carol is the respondent of the one with PNE-pos. According to c_2, Alice is the respondent of the tuple with PNE-pos, and Carol is the respondent of the one with BCM-pos.

Each possible configuration c_j is associated to a confidence degree d_j, that depends on the background knowledge of the adversary. d_j is computed as the sum of the probabilities, given by RBK^{sv} (or BK^{sv}), of the single associations between respondents and sensitive values in c_j. Given $r \in R$, and the set C of possible configurations, in order to calculate $PK^{sv}(r, \tau_i) = (p_1, p_2, \ldots, p_n)$ we need to compute, for each $p_m \in \{p_1, p_2, \ldots, p_n\}$, the sum of the degree of confidence of every possible configuration in which r is the respondent of a tuple having sensitive value s_m, divided by the sum of the degree of confidence of every possible configuration:

$$p_m = \frac{\sum_{\forall c_j \in C:\, c_j(t)=r \wedge t[S]=s_m} d_j}{\sum_{\forall c_j \in C} d_j}.$$

TABLE 4
Adversary's Posterior and Revised Knowledge

(a) PK^{sv} at τ_1

Name	Ex-res	p
Alice	MAM-pos	0.5
Alice	CX-neg	0.5
Betty	MAM-pos	0.5
Betty	CX-neg	0.5
Carol	CX-pos	0.5
Carol	BS-neg	0.5
Doris	CX-pos	0.5
Doris	BS-neg	0.5

(b) RBK^{sv} at τ_2

Name	BCM-pos	PNE-pos
Alice	0.31	0.05
Carol	0.02	0.31

(c) PK^{sv} at τ_2

Name	Ex-res	p
Alice	BCM-pos	0.9
Alice	PNE-pos	0.1
Carol	BCM-pos	0.1
Carol	PNE-pos	0.9

Example 2. Continuing Example 1, according to RBK_2^{sv} (Table 4b), the degree of confidence for c_1 is much higher than the one for c_2. Indeed, the probability of Alice being the respondent of a tuple with sensitive value BCM-pos is 0.31, which is also the probability of Carol being the respondent of the other tuple; hence, $d_1 = 0.31 + 0.31 = 0.62$. The probabilities regarding configuration c_2 are much lower, i.e., 0.05 and 0.02, respectively, i.e., $d_2 = 0.07$. Hence, if p_m is the probability of Alice being the respondent of a tuple with sensitive value BCM-pos, by applying the above formula we obtain $p_m = \frac{0.62}{0.62+0.07} \simeq 0.9$. The values of PK^{sv} at τ_2 are shown in Table 4c.

However, in general the exact computation of PK^{sv} is intractable; indeed, if the cardinality of the QI-group is k, the number of possible configurations is $k!$ Hence, an approximate algorithm is the natural candidate for the computation of posterior knowledge. In our experimental evaluation, we calculate posterior knowledge by the Ω-estimate method proposed by Li et al. in [17].

5.3 Computing Revised Knowledge

In order to compute revised sensitive values background knowledge at τ_i ($i > 1$) the adversary needs to calculate, for each respondent r of a tuple in V_i^*, and for each sensitive value $s \in D[S]$, the marginal probability of r to be the respondent of a tuple with private value s in V_i^*, given PK^{sv} and BK^{seq}. Let $\mathcal{V}^* = \langle V_1^*, V_2^*, \ldots, V_{i-1}^* \rangle$ be the history of released views containing a tuple of r, and $\mathcal{S}_i$ the random variable representing the sensitive value of r's tuple released at τ_i. Then, by applying the conditioning rule, we have

$$P(\mathcal{S}_i) = \sum_{\lambda \in \Lambda} (BK^{seq}(\lambda, T, r, \tau_i) \cdot P(\lambda)),$$

where $T = \langle \tau_1, \tau_2, \ldots, \tau_{i-1} \rangle$, Λ is the set of possible sequences of sensitive values of r's tuples released at T, and $P(\lambda)$ is the probability of sequence $\lambda \in \Lambda$. In particular, given the sequence $\lambda = \langle s_1, s_2, \ldots, s_{i-1} \rangle$, $P(\lambda)$ is the joint probability of the occurrence of each $s_j \in \lambda$ at τ_j based on PK^{sv}. If we denote as $p(r, s_j, \tau_j)$ that probability according to $PK^{sv}(r, \tau_j)$, we have

$$P(\lambda) = \prod_{s_j \in \lambda} (p(r, s_j, \tau_j)).$$

Note that, in our work, we do not make assumptions of independence among sensitive values over different releases. In fact, the independent posterior knowledge PK^{sv} is calculated using RBK^{sv}, which is obtained by applying the conditional probability given by BK^{seq} on the sequence of sensitive values in the release history. Hence, there is a dependency among sensitive values over different releases.

Example 3. Considering our running example, the adversary revises his sensitive values background knowledge after observing view V_1^* to obtain RBK_2^{sv} as follows: the probability $p(Alice, s, \tau_1)$ that Alice is the respondent of a tuple released at τ_1 having sensitive value s is given by PK_1^{sv} (Table 4a). Moreover, we represent by $\widetilde{p}(BCM\text{-}pos \mid s)$ the probability that an individual is the respondent of a tuple released at τ_2 with sensitive value BCM-pos provided that the same individual was the respondent of a tuple released at τ_1 with sensitive value s; this conditional probability is given by BK^{seq} (Table 3b). Then, the marginal probability of Alice to be the respondent of one tuple with BCM-pos at τ_2 can be calculated as

$$\begin{aligned} &p(Alice, BCM\text{-}pos, \tau_2) \\ &\quad = \sum_{\forall s \in D[S]} (p(Alice, s, \tau_1) \cdot \widetilde{p}(BCM\text{-}pos \mid s)) \\ &\quad = p(Alice, MAM\text{-}pos, \tau_1) \\ &\qquad \cdot \widetilde{p}(BCM\text{-}pos \mid MAM\text{-}pos) \\ &\qquad + p(Alice, CX\text{-}neg, \tau_1) \cdot \widetilde{p}(BCM\text{-}pos \mid CX\text{-}neg) \\ &\quad = 0.5 \cdot 0.6 + 0.5 \cdot 0.02 = 0.31. \end{aligned}$$

Conditioning over any possible private value s' other than MAM-pos and CX-neg is omitted from the above formula, since the probability $p(Alice, s', \tau_1)$ according to PK_1^{sv} is 0. Analogously, the adversary calculates that, according to RBK_2^{sv}, Alice has 0.05 probability to be the respondent of a tuple with private value PNE-pos, while the probability of Carol is 0.31 for PNE-pos, and 0.02 for BCM-pos (Table 4b).

6 JS-Reduce Defense

In this section, we illustrate the *JS-reduce* defense against the identified background knowledge attacks.

6.1 Defense Strategy

In order to enforce anonymity, it is necessary to limit the adversary's capability of identifying the actual respondent of a tuple in a given QI-group. Referring to the terminology introduced in Section 5.2, this means reducing the confidence of the adversary in discriminating a configuration $\widetilde{c}$ among the possible ones, based on his revised knowledge RBK^{sv}.

The goal of JS-reduce is to create QI-groups whose tuple respondents have similar RBK^{sv} (resp. BK^{sv}) distributions. Indeed, if the respondents of tuples in a QI-group are indistinguishable with respect to RBK^{sv} (resp. BK^{sv}), the adversary cannot exploit background knowledge to perform the attack. Of course, defending against background knowledge attacks is not sufficient to guarantee privacy protection against other kinds of attacks. For this reason, JS-reduce also enforces k-anonymity and t-closeness [15], in order to protect against well-known identity and attribute-disclosure attacks,

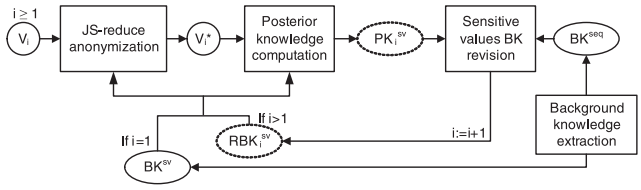


Fig. 2. Defense mechanisms.

respectively. Note that JS-reduce can be easily extended to enforce additional privacy models, like l-diversity [19].

6.2 Defending against Sequential Background Knowledge Attacks

In order to measure the similarity of probability distributions RBK^{sv} (BK^{sv}), we adopt *Jensen-Shannon divergence (JS)* [18]; this measure is used in [17] to quantify information disclosure. With respect to other distance measures among probability distributions, this function has three important properties: 1) it can be computed on a set of more than two distributions; 2) it is always a definite number; 3) it is symmetric with respect to the order of the arguments. Suppose that $\mathbf{P} = \{\overline{p}^1, \ldots, \overline{p}^u\}$ is a set of probability distributions such that each element has form: $\overline{p}^i = (p_1^i, \ldots, p_s^i)$; $\pi^1, \ldots, \pi^u$ denote the *weights* of the probability distributions, and $\sum_{i=1}^{u} \pi^i = 1$. Then, the JS divergence among distributions in $\mathbf{P}$ is

$$JS(\mathbf{P}) = H\left(\sum_{i=1}^{u} \pi^i \cdot \overline{p}^i\right) - \sum_{i=1}^{u} \pi^i \cdot H(\overline{p}^i),$$

where $H(\overline{p})$ is the Shannon entropy of $\overline{p} = (p_1, \ldots, p_s)$. In our case, each $\overline{p}^i$ corresponds to the background knowledge about a tuple respondent; since this probability $\overline{p}^i$ already includes the adversary's confidence, when we compute the above formula we assign the same weight to each probability distribution.

Given a required threshold j, the JS-reduce defense guarantees that, for each QI-group Q in an anonymized view, the JS divergence of the set of probability distributions RBK^{sv} (BK^{sv}) of respondents of tuples in Q is below j. Note that, given the privacy preferences expressed by the data owner, the actual value of threshold j must be chosen according to many domain-specific factors, including the diversity of sensitive values in released views, and background knowledge. Similar considerations apply for the choice of the parameter k of k-anonymity and t of t-closeness.

Clearly, in order to be effective against sequential background knowledge attacks, JS-reduce needs to calculate the RBK^{sv} distribution of respondents before anonymizing data. Hence, similarly to the knowledge revision cycle presented in Section 5, the defense technique (illustrated in Fig. 2), performs posterior knowledge computation, and sensitive values background knowledge revision. BK^{sv} and BK^{seq} are obtained using one of the techniques illustrated in Section 5.1.

6.3 The JS-Reduce Algorithm

The pseudocode of the JS-reduce algorithm is shown in Algorithm 1. The algorithm takes as input:

1. A sequence $\mathcal{H}_n = \langle V_1, \ldots, V_n \rangle$ of original views.
2. The set R of respondents of tuples in $\mathcal{H}_n$, as well as their QI values.
3. Sensitive values background knowledge BK^{sv} and sequential background knowledge BK^{seq}.
4. The minimum level k of k-anonymity, threshold t of t-closeness, and threshold j of JS divergence.

It returns V_n^*, the generalization of V_n.

Algorithm 1. JS-reduce algorithm

Input: Sequence $\mathcal{H}_n = \langle V_1, \ldots, V_n \rangle$, the set R of possible respondents as well as their QI values, BK^{sv}, BK^{seq}, the minimum level k of k-anonymity, threshold t of t-closeness, threshold j of JS divergence.
Output: V_n^*

JS-reduce$(\mathcal{H}_n, R, BK^{sv}, BK^{seq}, k, t, j)$
begin
  forall $r \in R$ **do**
    $RBK_1^{sv}(r) \leftarrow BK^{sv}(r)$
  end
  for $h = 1$ to n **do**
    $V_h^* \leftarrow$ Generalize$(V_h, RBK_h^{sv}, t, j, k)$
    forall $r \in R_h$ **do**
      $PK_h^{sv}(r) \leftarrow$ PKComputation(V_h^*, RBK_h^{sv}, r)
      $RBK_{h+1}^{sv}(r) \leftarrow$ BKRevision$(PK^{sv}(r), BK^{seq}, r)$
    end
  end
  return V_n^*
end

Algorithm 2. PK^{sv} computation procedure

Input: The anonymized release V_h^*, the set RBK_h^{sv} of revised background knowledge for each respondent of a tuple in V_h^*, respondent r
Output: $PK_h^{sv}(r)$

PKComputation(V_h^*, RBK_h^{sv}, r)
begin
  QI-group $Q \leftarrow Q' \in V_h^*$ s.t. r is the respondent of one tuple in Q'
  C $\leftarrow \{c_j \mid c_j$ is a valid configuration for $Q\}$
  forall $c_j \in C$ **do**
    confidence degree $d_j \leftarrow 0$
    forall r' *s.t.* $\exists t \in Q \mid c_j(t) = r'$ **do**
      $t' \leftarrow t \mid c_j(t) = r'$
      $d_j \leftarrow d_j + RBK_h^{sv}(r', t'[S])$
    end
  end
  forall $s \in D[S]$ **do**
    $p(r, s) \leftarrow \frac{\sum_{\forall c_j \in C \mid c_j(t) = r \wedge t[S] = s} d_j}{\sum_{c_j \in C} d_j}$
  end
  $PK_h^{sv}(r) \leftarrow \{p(r, \tilde{s}), \forall \tilde{s} \in D[S]\}$
  return $PK_h^{sv}(r)$
end

Algorithm 3. RBK^{sv} revision procedure

Input: The set of posterior knowledge of respondent r $PK^{sv}(r) = \{PK_1^{sv}(r), \ldots, PK_h^{sv}(r)\}$, the available sequential background knowledge BK^{seq}, respondent r
Output: $RBK_{h+1}^{sv}(r)$

1 **BKRevision**$(PK^{sv}(r), BK^{seq}, r)$
2 **begin**
3 $\Lambda \leftarrow \{\lambda = \langle s_1, \ldots, s_i \rangle \mid s_j$ is a possible sensitive value for r released at $\tau_j\}$
4 **forall** $\lambda \in \Lambda$ **do**
5 $P(\lambda) \leftarrow 1$
6 **forall** $s_j \in \lambda$ **do**
7 $P(\lambda) \leftarrow P(\lambda) \cdot PK_j^{sv}(r, s_j)$
8 **end**
9 **end**
10 **forall** $s \in D[S]$ **do**
11 $\widetilde{p}(s \mid \lambda)$ is the conditional probability given by BK^{seq}
12 $p(s) \leftarrow \sum_{\lambda \in \Lambda} \widetilde{p}(s \mid \lambda) \cdot P(\lambda)$
13 **end**
14 $RBK_{h+1}^{sv}(r) \leftarrow \{p(s), \forall s \in D[S]\}$
15 **return** $RBK_{h+1}^{sv}(r)$
16 **end**

At first (lines 2 to 4), for each respondent of tuples in $\mathcal{H}_n$, RBK^{sv} at τ_1 is initialized according to BK^{sv}. Then (lines 5 to 11), each view V_i in $\mathcal{H}_n$ is processed in turn, from V_1 to V_n. In particular, each V_i is generalized by the *Generalize* procedure (line 6) in order to enforce thresholds j of JS divergence, t of t-closeness, and minimum cardinality k. The algorithm for generalization, specifically designed to preserve the data quality, is described in detail in Section 6.4. We call V_i^* the generalization of V_i, and R_i the set of respondents of tuples in V_i^*. After the generalization, for each respondent in R_i, JS-reduce calculates the posterior knowledge and the revised sensitive values background knowledge at τ_{i+1}. Finally (line 12), the generalized view V_n^* is returned. The algorithm always terminates (it sequentially assigns tuples to a QI-group); however, as for other privacy-preserving techniques (e.g., [26]), it is possible that some tuples cannot be arranged in any QI-group without violating some of the privacy requirements. In this case, different solutions can be taken, including: 1) suppressing those tuples outside QI-groups; 2) executing the algorithm using less stringent values for privacy parameters (k, t, and j); 3) counterfeiting the sensitive values of those tuples outside QI-groups, so that they can be inserted in existing QI-groups without violating privacy requirements. For the sake of this paper, we adopt solution 1. Experimental results, reported in Section 7, show that the percentage of suppressed tuples is negligible. However, if an adversary knows the whole set of tuple respondents of the *original* view (e.g., according to our running example, "people that performed an exam during week i"), solution 1 may be prone to attacks in which the adversary recognizes that some tuples have been suppressed, and infers the cause of suppression. In order to defend against this additional knowledge assumption, JS-reduce may be easily modified to adopt solutions 2 or 3, which are not subject to that attack.

6.4 Data Quality-Oriented Generalization

Any anonymization technique based on QI generalization needs to carefully consider the resulting data quality: the more the QI values are generalized, the lower is the quality of released data. Hence, instead of adopting a general-purpose anonymization framework such as Mondrian [13], we devised an ad hoc technique. Note that finding the optimal generalization of data that satisfies the privacy requirements of JS-reduce (i.e., the one that minimizes QI generalization) is an NP-hard problem; indeed, it is well known that even optimal k-anonymous generalization is NP-hard [21]. For this reason, we devised an approximate algorithm (Algorithm 4). The *Generalize* procedure receives as input: 1) the original view V_h; 2) revised sensitive values background knowledge at τ_h; 3) a minimum level k of k-anonymity, threshold t of t-closeness and threshold j of *JS* divergence. It returns V_h^*, the generalization of V_h.

Algorithm 4. Generalization procedure

1 **Generalize**$(V_h, RBK_h^{sv}, t, j, k)$
2 **begin**
3 $V_h^* = \emptyset$
4 **forall** $v \in V_h$ **do**
5 $i_v \leftarrow$ ComputeHilbertIndex(v)
6 **end**
7 $\widetilde{V_h} \leftarrow$ OrderOnHilbertIndex(V_h)
8 $Q \leftarrow \emptyset$
9 **for** $\tilde{v} = v_1$ to $v_{\widetilde{|V_h|}}$ **do**
10 $Q \leftarrow Q \cup \tilde{v}$
11 **If** $|Q| \geq k \wedge t\text{-}clos(Q) \leq t \wedge js(Q) \leq j$ **then**
12 CreateQIG(Q)
13 $Q \leftarrow \emptyset$
14 **end**
15 **end**
16 **If** $Q \neq \emptyset$ **then**
17 Remove tuples $v \in Q$
18 **end**
19 **return** V_h^*
20 **end**

1 **CreateQIG**(Q)
2 **begin**
3 GeneralizeQIvalues(Q)
4 $V_h^* \leftarrow V_h^* \cup Q$
5 $\widetilde{V_h} \leftarrow \widetilde{V_h} \setminus \{v \in Q\}$
6 **end**

As proposed in [11], in order to partition tuples in QI-groups, the procedure exploits the Hilbert space-filling curves.[3] For each tuple in V_h, function *ComputeHilbertIndex* (line 4) computes its Hilbert index considering the multidimensional space having the QI attributes as dimensions. Then, tuples in V_h are reordered with respect to their Hilbert index, obtaining an auxiliary list $\widetilde{V_h}$ (line 6). The procedure takes the first k tuples in $\widetilde{V_h}$ and checks if the

3. A Hilbert space-filling curve is a function that maps a point in a multidimensional space into an integer. With this technique, two points that are close in the multidimensional space are also close, with high probability, in the 1D space obtained by the Hilbert transformation.

t-closeness and *JS* divergence values of that group are below thresholds t and j, respectively. Note that, according to the Hilbert transformation, tuples with similar QI values are close in the list $\widetilde{V}_h$, and respondents having similar QI values are also likely to have similar probability distributions according to BK^{sv}. Hence, we achieve both of our goals: 1) it is likely to find groups of tuples satisfying privacy constraints, and 2) we limit the generalization of QI values. Then, if the required privacy constraints are satisfied, a new QI-group is created (line 10) by procedure *CreateQIG*: the QI values are substituted with intervals including the QI values of each tuple, and the original tuples are removed from $\widetilde{V}_h$; the same procedure is repeated with the remaining tuples. Otherwise (if constraints are violated), the next tuple in $\widetilde{V}_h$ is added to the group until the constraints are satisfied (line 10).

As explained in Section 6, it may happen that a few tuples cannot be grouped into a QI-group (line 15) during the first phase. In the current version of the algorithm, those tuples are suppressed. However, the algorithm can be easily modified to apply other solutions.

7 Experimental Evaluation

In this section, we present an experimental evaluation of the privacy threats due to sequential background knowledge attacks, we compare our defense with other applicable solutions, and we evaluate data quality.

7.1 Experimental Setup

To the best of our knowledge, all the data sets used for experimental evaluation of proposed privacy defenses for serial data publication were created from sets of tuples nontemporally characterized, in which each tuple was randomly assigned to a release. Clearly, these data sets are not realistic for investigating the use that an adversary can make of temporal correlations. The data set used in our experiments has been synthetically created based on domain knowledge extracted from the medical literature; in particular, studies reported in [9], [10], [22], and [23]. Each of those papers provides the probabilities that a specific disease evolves from one stage to another based on the characteristics of the patient (age, gender, and weight) and on the past evolution of the disease. Based on that information, we computed BK^{seq} as the probability of a patient performing an exam at τ_i to obtain a given result $ex\text{-}res_i$ given a sequence of results of exams performed by that person in the previous weeks. BK^{sv} was calculated dividing age and weight into three subintervals (each one containing 10 values), and assigning different probability distributions to each of the 18 classes of users obtained combining age, weight, and gender values. The data set has been made available from our group and can be used to replicate our experiments, or as a testbed for other research.[4]

Experiments were performed on a history of 24 views, each one containing 5,000 tuples. A total of 16,160 individuals appear in at least one view of the history. Tuples in the data set represent the results of medical exams performed in a given institute. One view per week is released, and each view contains the records of exams performed during that week. A tuple is composed of three QI attributes *age*, *gender*, and *weight*, and a sensitive attribute *Ex-res*. *Age* has values in the interval $[45, 74]$, *gender* in $[1, 2]$, and *weight* in $[60, 89]$. The domain of *Ex-res* includes 17 different values associated to stages of different diseases (five stages of liver disease, four of the HIV syndrome, three of Alzheimer, and five of sepsis), as well as two sensitive values to describe the *deceased* and *discharged* events.

TABLE 5
Values of Privacy Parameters Used in the Experiments

	l	t	B	j
l-div.	$[2, 8]$ **2**	-	-	-
t-clos.	-	$[0.5, 1]$ **0.8**	-	-
(B,t)-priv.	-	$[0.5, 0.8]$ **0.8**	$[0.3, 0.7]$ **0.5**	-
JS-red.	-	$[0.5, 0.8]$ **0.5**	-	$[0.2, 0.8]$ **0.6**

Since our study is the first to consider the role of sequential background knowledge in privacy-preserving data publishing, a direct comparison with techniques specifically devoted to protect against the identified threats was not possible. However, we performed experiments to compare JS-reduce with state-of-the-art privacy protection methods that are applicable to our case: 1) distinct l-diversity (each QI-group must contain at least l tuples having different sensitive values), 2) t-closeness [15], and 3) (B, t)-privacy [17]. We used the Mondrian framework [13] and the modules provided by the corresponding authors to apply the latter methods; we used Algorithm 1 to apply the JS-reduce defense. Experiments were performed on a 2.4 GHz workstation with 4 GB RAM. The time required for anonymizing a view with the JS-reduce algorithm varied from a few minutes to a maximum of 43 minutes; this is an acceptable time, since, in most cases, microdata anonymization is performed offline. For each considered technique, we made experiments with different values of the corresponding privacy parameters. Of course, stricter values determine stronger privacy protection but lower data utility; hence, for each technique, values were chosen considering the privacy/utility tradeoff. Values of privacy parameters are shown in Table 5; bold numbers indicate the values used in the following experiments.

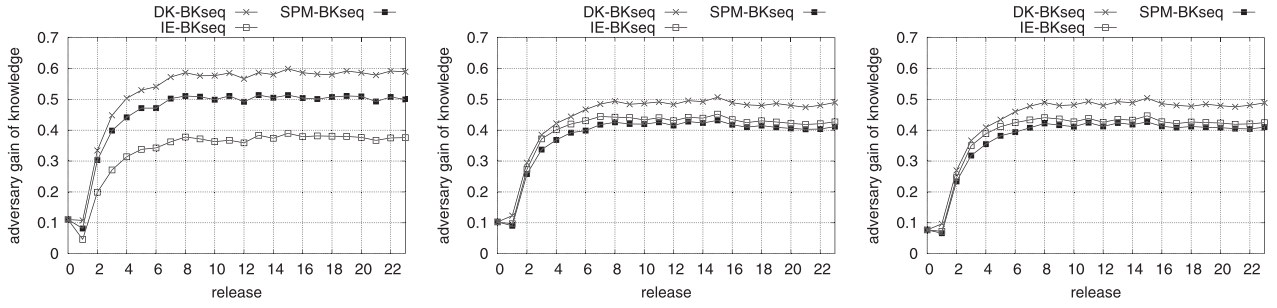
7.2 Measuring the Adversary Gain of Knowledge

In order to evaluate the privacy threat, we measured the *gain of knowledge* when an adversary is able to exploit sequential background knowledge. For a given generalized view V_i^* released at τ_i containing N tuples, we measured the *average adversary gain* ρ as follows:

$$\rho = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{p(r_j, s_{i_j}, \tau_i) - \frac{m(s_{i_j})}{|Q_{i_j}|}}{1 - \frac{m(s_{i_j})}{|Q_{i_j}|}}\right),$$

where $p(r_j, s_{i_j}, \tau_i)$ is the value of posterior knowledge computed based on background knowledge for respondent r_j and her actual private value s_{i_j} at τ_i; Q_{i_j} is the QI-group of V_i^* containing the tuple whose respondent is r_j; and $m(s_{i_j})$ is the number of tuples t in Q_{i_j} such that $t[S] = s_{i_j}$. The adversary gain measures the information obtained with the

4. http://webmind.dico.unimi.it/BKseq-data set.zip.



(a) l-div. (b) t-clos. (c) (B,t)-priv.

Fig. 3. Adversary gain versus different kinds of adversary's BK^{seq}.

use of background knowledge with respect to an attack based only on the observation of the frequency of sensitive values in the QI-group.

7.3 The Role of Adversary's Background Knowledge

We performed experiments to evaluate the role of background knowledge on the privacy threats investigated in this paper:

- *Incrementally extracted knowledge* $IE\text{-}BK^{seq}$. Since it was the subject of related studies (e.g., [16], [17]), the first kind of background knowledge we consider is the one directly extracted from the data to be released. $IE\text{-}BK^{seq}$ can be calculated by applying sequential pattern mining (SPM) techniques on the history of original (i.e., nonanonymized) data at each time τ_i, $IE\text{-}BK^{seq}$ is calculated based on V_i. Since the size of the corpus is relatively small, we applied a simple SPM algorithm, which is essentially based on a frequency count of sequences appearing in the history. The algorithm is illustrated in Algorithm 5.
- *Mined knowledge* $SPM\text{-}BK^{seq}$. In practice, an adversary may approximate BK^{seq} by applying SPM techniques on an external corpus of nonanonymized data. We created a data corpus using the same model that we used to generate our data set; the corpus consists in a history of 24 views containing 5,000 tuples each. $SPM\text{-}BK^{seq}$ was calculated by applying Algorithm 5 to that corpus.
- *Domain knowledge* $DK\text{-}BK^{seq}$. Since our data set was generated based on domain knowledge, in our experiments $DK\text{-}BK^{seq}$ corresponds to the exact BK^{seq}; i.e., it is the "best" knowledge that an adversary may have. However, in general an adversary's knowledge may only approximate the exact BK^{seq}. Hence, we also considered another kind of knowledge, whose temporal extent is limited to a number n of past observations. We denote this knowledge as *n-steps* $DK\text{-}BK^{seq}$, and we consider $n = 1$, $n = 2$, and $n = 3$.

Fig. 3 shows the adversary gain when views are anonymized using existing techniques. Results show that existing techniques are effective when an adversary cannot exploit sequential background knowledge; indeed, at the first release, the adversary gain is very low (less than 0.1) with each considered technique. However, existing techniques are not effective when an adversary may observe sequential releases and perform the attacks identified in this paper based on sequential background knowledge. Indeed, with each kind of background knowledge, the adversary gain grows rapidly during the first 6/8 releases, exceeding the value of 0.4.

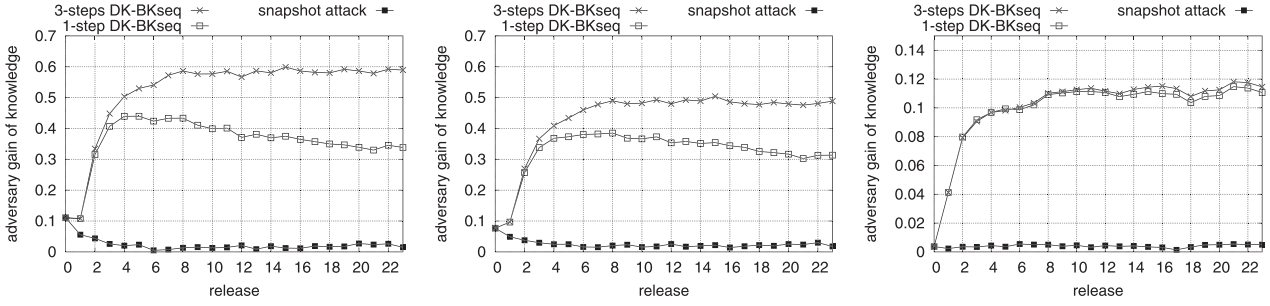
Algorithm 5. $SPM\text{-}BK^{seq}$ extraction

Input: History of original views $\mathcal{H}_r = \langle V_1, \ldots, V_r \rangle$, a sequence of sensitive values seq, and a sensitive value s.

Output: The conditional probability $p(s|seq)$, which corresponds to the frequency of sequence $\langle seq, s \rangle$ in $\mathcal{H}_r$.

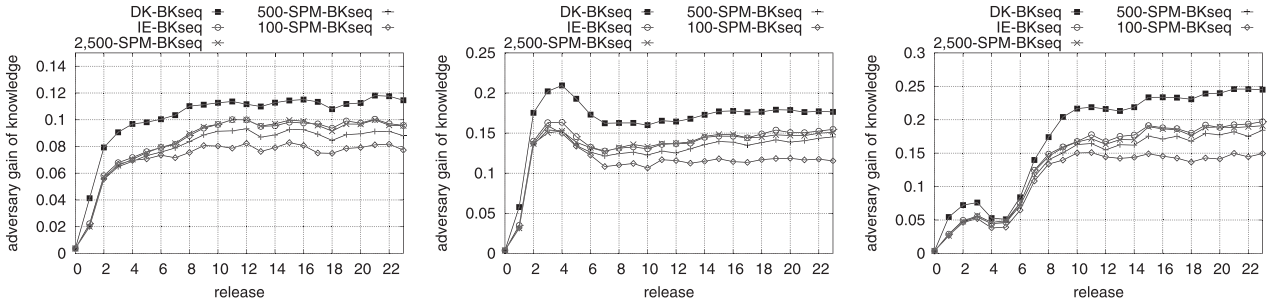
```
1  SPM(H_r, seq, s) begin
2      for h = 1 to r do
3          forall respondent u of a tuple in V_h do
4              for j = h to 1 do
5                  seq_j = seq. of past j sensitive values of
                   u in H_h
6                  seq_j.numOcc = seq_j.numOcc + 1
7              end
8          end
9      end
10     if (seq.numOcc == 0) then return 0
11     else
12         sequence = ⟨seq, s⟩
13         return sequence.numOcc / seq.numOcc
14     end
15 end
```

For each considered anonymization technique, the form of background knowledge that determines the highest adversary gain is full $DK\text{-}BK^{seq}$, since in our experiments it corresponds to the exact BK^{seq}. Hence, we considered approximate $DK\text{-}BK^{seq}$ in order to better evaluate the role of domain knowledge. Results illustrated in Figs. 4a and 4b show that even attacks based on approximate $DK\text{-}BK^{seq}$ are effective against existing anonymization techniques; attacks exploiting 3-steps $DK\text{-}BK^{seq}$ are more successful than the ones exploiting 2-steps and 1-step knowledge (we omit the plot for t-closeness since it is analogous to the one for (B,t)-privacy). Results also show that when the adversary exploits only BK^{sv} (i.e., when he performs a *snapshot* attack), the gain of information with respect to an attack considering only the frequency of sensitive values is negligible. The descending shape of curves for the 1-step



(a) l-div. (b) (B,t)-priv. (c) JS-red.

Fig. 4. Adversary gain versus accuracy of adversary's domain knowledge $DK\text{-}BK^{seq}$.



(a) Defense based on $DK\text{-}BK^{seq}$ (b) Defense based on $IE\text{-}BK^{seq}$ (c) Defense based on $SPM\text{-}BK^{seq}$

Fig. 5. JS-reduce versus different kinds of adversary's BK^{seq}.

and snapshot attacks is due to the fact that the background knowledge used by the adversary tends to diverge from the one that generated the data.

7.4 Effectiveness of the JS-Reduce Defense

Results reported in Fig. 4c show that, when views are anonymized with JS-reduce, the adversary gain remains below 0.12, independently from the length of the released history, and on the kind of domain knowledge available to the adversary. This result shows that JS-reduce significantly limits the inference capabilities of the adversary with respect to the other techniques that lead to an adversary gain higher than 0.5.

We performed other experiments to evaluate the effectiveness of JS-reduce with different combinations of background knowledge available to the defender and to the adversary, respectively. In Fig. 5a, we considered the case in which the defender has background knowledge $DK\text{-}BK^{seq}$. In this case, the defense is very effective, even when the adversary has the same background knowledge as the defender. When the adversary's background knowledge is extracted from the data, we observe that the adversary gain is lower. With the label $n\text{-}SPM\text{-}BK^{seq}$ in Fig. 5, we denote that the adversary's $SPM\text{-}BK^{seq}$ is extracted based on a history of 24 views containing n tuples each. The adversary gain is lower with smaller values of n, since the resulting $SPM\text{-}BK^{seq}$ is a coarser approximation of the exact BK^{seq}. The adversary gain with incrementally extracted knowledge is comparable to the one obtained with $SPM\text{-}BK^{seq}$.

We also considered the unfortunate case in which the adversary has more accurate background knowledge than the defender. Results illustrated in Figs. 5b and 5c show the adversary gain when the defender's background knowledge is $IE\text{-}BK^{seq}$ and $SPM\text{-}BK^{seq}$, respectively. As expected, the more accurate the attacker's background knowledge with respect to the defender's one, the more effective the attack. However, results show that JS-reduce provides privacy protection even in the worst case; indeed, the adversary gain remains below 0.25. It is important to note that JS-reduce is effective even when the defender has neither domain knowledge, nor external data to derive background knowledge. Indeed, even extracting background knowledge from the data to be released, the adversary gain is low.

In order to study in more detail the effectiveness of JS-reduce, we considered a further metric, named *average adversary confidence*. We call *adversary confidence regarding respondent r at release τ_j* the value of the posterior probability $PK^{sv}(r, \tau_j)$ computed by the adversary for the actual private value of r at τ_j. The average adversary confidence about a generalized view V_j^* is the average of the adversary confidence regarding respondents of tuples in V_j^*. Fig. 6 shows a comparison among the considered privacy techniques in terms of the adversary confidence with respect to the number of observed anonymized views (attack and defense are based on $DK\text{-}BK^{seq}$). These results show that with our technique the adversary confidence does not significantly grow with respect to the length of the release history. On the contrary, with the other techniques,

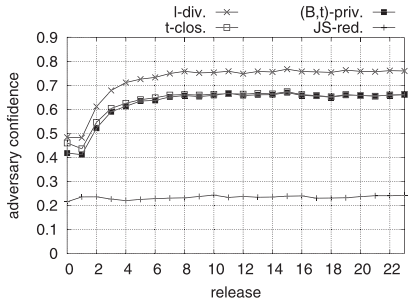


Fig. 6. Adversary confidence.

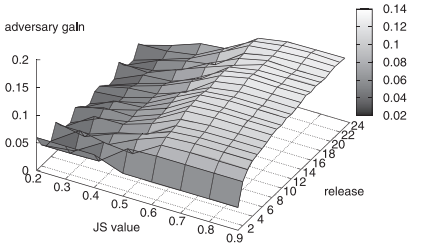
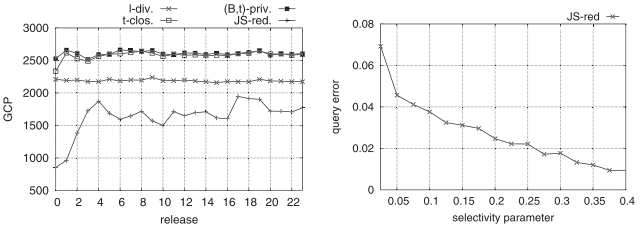


Fig. 7. Adversary gain versus *JS* divergence ($t = 0.5$).



(a) GCP (b) Query error

Fig. 8. Data quality evaluation.

after a few anonymized views have been released, the adversary can predict with high confidence the exact sensitive values of tuples respondents. We also performed specific experiments to evaluate the impact on privacy protection of the *JS* divergence threshold for the JS-reduce defense. Results are illustrated in Fig. 7; as expected, the lower the JS threshold value, the lower the adversary gain.

7.5 Data Utility

In order to evaluate data utility, we considered both general utility measures, and accuracy of aggregate query answering. General utility is evaluated in terms of *Global Certainty Penalty* (GCP) [27], a well-known metric taking into account the level of generalization of QI values. Fig. 8a shows the average GCP value of QI-groups generated by the considered techniques (JS-reduce is based on $DK\text{-}BK^{seq}$). As it can be seen, JS-reduce outperforms the other techniques; however, in terms of data quality, results obtained with our technique are not directly comparable with the ones of the other techniques, since the latter were based on a Mondrian implementation, which is not data-quality oriented. With data-quality oriented implementations of those techniques, we expect to obtain a data quality very close to ours.

Then, we evaluated the utility of transaction data generalized by our technique in terms of precision in answering aggregate queries (e.g., "*count the number of individuals in the table whose QI-values belong to certain ranges*"). Queries were randomly generated according to different values of expected selectivity, i.e., expected ratio of tuples to be returned by the query. For each value of expected selectivity, 10,000 random queries were evaluated. Results reported in Fig. 8b show that the median error is very low, even with low values of expected selectivity (i.e., very specific aggregate queries). Finally, results show that a very few number of tuples were suppressed by JS-reduce to enforce privacy requirements; i.e., at most 12 ($<0.25\%$) at each release.

8 Conclusions

In this paper, we showed that the correlation of sensitive values in subsequent releases can be used as background knowledge to violate users' privacy. We showed that an adversary can actually obtain this knowledge by different methods. We proposed a defense algorithm and we showed through an extensive experimental evaluation that other applicable solutions are not effective, while our defense provides strong privacy protection and good data quality, even when the adversary has more accurate background knowledge than the defender. Our framework is seamlessly extensible with additional forms of probabilistic inference, since the JS-reduce technique relies on a background knowledge revision process that is not tied to a specific inference method.

Acknowledgments

The authors would like to thank Kristen LeFevre for providing an implementation of the Mondrian framework, and Tiancheng Li, Ninghui Li, and Jian Zhang for providing modules for (B, t)-privacy. This work was partially supported by Italian MIUR under grants PRIN-2007F9437X and InterLink II04C0EC1D, and by the US National Science Foundation under grant CNS-0716567.

References

[1] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. 11th Int'l Conf. Data Eng. (ICDE '95)*, pp. 3-14, 1995.

[2] Y. Bu, A. Wai, C. Fu, R.C.W. Wong, L. Chen, and J. Li, "Privacy Preserving Serial Data Publishing by Role Composition," *Proc. VLDB Endowment*, vol. 1, pp. 845-856, 2008.

[3] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure Anonymization for Incremental Data Sets," *Proc. Third VLDB Workshop Secure Data Management (SDM '06)*, pp. 48-63, 2006.

[4] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan, "CASTLE: Continuously Anonymizing Data Streams," *IEEE Trans. Dependable and Secure Computing*, vol. 8, no. 3, pp. 337-352, May-June 2011.

[5] T.-H. Hubert Chan, E. Shi, and D. Song, "Private and Continual Release of Statistics," *Proc. 37th Int'l Colloquium Conf. Automata, Languages and Programming (ICALP '10)*, pp. 405-417, 2010.

[6] W. Du, Z. Teng, and Z. Zhu, "Privacy-MaxEnt: Integrating Background Knowledge in Privacy Quantification," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08)*, pp. 459-472, 2008.

[7] C. Dwork, "Differential Privacy," *Proc. 33rd Int'l Colloquium on Automata, Languages and Programming (ICALP '06)*, pp. 1-12, 2006.

[8] C. Dwork, M. Naor, T. Pitassi, and G.N. Rothblum, "Differential Privacy under Continual Observation," *Proc. 42nd ACM Symp. Theory of Computing (STOC '10)*, pp. 715-724, 2010.

[9] G. Di Biase et al., "A Stochastic Model for the HIV/AIDS Dynamic Evolution," *Math. Problems in Eng.*, 2007.

[10] J.-L. Fuh, R.-F. Pwu, S.-J. Wang, and Y.-H. Chen, "Measuring Alzheimer' s Disease Progression with Transition Probabilities in the Taiwanese Population," *Int'l J. Geriatric Psychiatry*, vol. 19, no. 3, pp. 266-270, 2004.

[11] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast Data Anonymization with Low Information Loss," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07)*, pp. 758-769, 2007.

[12] D. Kifer and A. Machanavajjhala, "No Free Lunch in Data Privacy," *Proc. Int'l Conf. Management of Data (SIGMOD '11)*, pp. 193-204, 2011.

[13] K. LeFevre, D.J. DeWitt, and R. Raghu, "Mondrian Multidimensional k-Anonymity," *Proc. 22nd Int'l Conf. Data Eng. (ICDE '06)*, 2006.

[14] J. Li, B.C. Ooi, and W. Wang, "Anonymizing Streaming Data for Privacy Protection," *Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE '08)*, pp. 1367-1369, 2008.

[15] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy beyond k-Anonymity and l-Diversity," *Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE '07)*, pp. 106-115, 2007.

[16] T. Li and N. Li, "Injector: Mining Background Knowledge for Data Anonymization," *Proc. IEEE 24th Int'l Conf. Data Eng. (ICDE '08)*, pp. 446-455, 2008.
[17] T. Li, N. Li, and J. Zhang, "Modeling and Integrating Background Knowledge in Data Anonymization," *Proc. IEEE 25th Int'l Conf. Data Eng. (ICDE '09)*, pp. 6-17, 2009.
[18] J. Lin, "Divergence Measures based on the Shannon Entropy," *IEEE Trans. Information Theory*, vol. 37, no. 1, pp. 145-151, Jan. 1991.
[19] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "*l*-Diversity: Privacy Beyond *k*-Anonymity," *ACM Trans. Knowledge Discovery from Data*, vol. 1, no. 1, article 3, 2007.
[20] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J.Y. Halpern, "Worst-Case Background Knowledge for Privacy-Preserving Data Publishing," *Proc. 23rd Int'l Conf. Data Eng. (ICDE '07)*, pp. 126-135, 2007.
[21] A. Meyerson and R. Williams, "On the Complexity of Optimal *k*-Anonymity," *Proc. 23rd ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (PODS '04)*, pp. 223-228, 2004.
[22] M.S. Rangel-Frausto, D. Pittet, T. Hwang, R.F. Woolson, and R.P. Wenzel, "The Dynamics of Disease Progression in Sepsis: Markov Modeling Describing the Natural History and the Likely Impact of Effective Antisepsis Agents," *Clinical Infectious Diseases*, vol. 27, no. 1, pp. 185-190, 1998.
[23] R.S. Remis, "A Study to Characterize the Epidemiology of Hepatitis C Infection in Canada," technical report, Health Agency of Canada, 2002.
[24] P. Samarati, "Protecting Respondents' Identities in Microdata Release," *IEEE Trans. Knowledge Data Eng.*, vol. 13, no. 6, pp. 1010-1027, Nov./Dec. 2001.
[25] R. Chi-Wing Wong, A. Wai-Chee Fu, J. Liu, K. Wang, and Y. Xu, "Global Privacy Guarantee in Serial Data Publishing," *Proc. 26th Int'l Conf. Data Eng. (ICDE '10)*, pp. 956-959, 2010.
[26] X. Xiao and Y. Tao, "*m*-Invariance: Towards Privacy Preserving Re-Publication of Dynamic Data Sets," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '07)*, pp. 689-700, 2007.
[27] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. Wai-Chee Fu, "Utility-Based Anonymization Using Local Recoding," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD' 06)*, pp. 785-790, 2006.

Daniele Riboni received the PhD degree in computer science in 2007. He holds a postdoctoral position at the University of Milan, and collaborates with the University La Sapienza, Rome. His research interests include mainly context-awareness, data management, mobile and pervasive computing, and privacy. He has published numerous papers in international journals and conference proceedings. He belongs to the organizing committee of the IEEE Workshop on Context Modeling and Reasoning (CoMoRea).

Linda Pareschi received the PhD degree in computer science from the University of Milan in 2009. She has been a postdoctoral fellow at the same University. Her main research interests include mobile and pervasive computing, context awareness, and privacy.

Claudio Bettini is a professor of computer science at the University of Milan, Italy, where he leads the EveryWare research laboratory. He is in the Steering Committee of the TIME conference (International Conference Temporal Representation and Reasoning) and he has been associate editor of *IEEE Transactions on Knowledge and Data Engineering* and of the *VLDB Journal*. His research interests include areas of temporal and spatial data management, mobile and pervasive computing, privacy and security. On these topics, he has extensively published in top conferences and journals. He is a member of the ACM SIGMOD and the IEEE Computer Society.