Abstract—A large amount of transaction data containing associations between individuals and sensitive information flows everyday into data stores. Examples include web queries, credit card transactions, medical exam records, transit database records. The serial release of these data to partner institutions or data analysis centers is a common situation. In this paper we show that, in most domains, correlations among sensitive values associated to the same individuals in different releases can be easily mined, and used to violate users’ privacy by adversaries observing multiple data releases. We provide a formal model for privacy attacks based on this sequential background knowledge, as well as on background knowledge on the probability distribution of sensitive values over different individuals. We show how sequential background knowledge can be actually obtained by an adversary, and used to identify with high confidence the sensitive values associated with an individual. A defense algorithm based on Jensen-Shannon divergence is proposed, and extensive experiments show the superiority of the proposed technique 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.

I. INTRODUCTION

Large amounts of transaction data related to individuals are continuously acquired, and stored in the repositories of industry and government institutions. Examples include online service requests, web queries, credit card transactions, transit database records, medical exam 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 inter-organizational 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 non-aggregated form.

A very 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 dataset that a sequence of medical exam results within a certain time frame has higher probability to be observed than another sequence.

Related work has either focused on anonymization techniques dealing with multiple data releases, or on privacy protection techniques taking into account 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 [1], [2], [3], [4], [5], [6]. The work that is closest to ours is probably the one presented in [5], in which sensitive values are divided in transient values that may freely change with time, and persistent values that never change. However, the proposed 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. In [6] a technique is proposed to defend against attacks based on the observation of serial data having transient sensitive values; however, background knowledge on transition probabilities is not considered in that work. On the contrary, our privacy preserving technique captures non-uniform transition probabilities. Our running example in Section [7] shows that the anonymizations proposed in related works are not effective when an adversary can obtain background knowledge on the transition probabilities.

Techniques considering background knowledge have also been proposed, and they can be classified according to two main categories: a) models based on logic assertions and rules [7]; and b) models based on probabilistic tools [3], [9]. However, these techniques are devised for a single release of the data, and, as it is shown in Section [10], they are ineffective when an adversary having background knowledge on sequences of sensitive values may observe multiple releases.

In this paper we formally model privacy attacks based on background knowledge extended to serial data releases. We present a new probabilistic defense technique taking into account possible 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) attributes, but generalization is performed with a new goal: minimizing the differ-
We propose privacy attacks that current anonymization techniques are not resistant to these sensitive values and sequences of sensitive values. We show on background knowledge about the probability distributions that this defense is effective under different combinations of the knowledge of the adversary and the defender.

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

(i) 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.

(ii) We propose JS-reduce as a new probabilistic defense technique based on Jensen-Shannon divergence.

(iii) Through an experimental evaluation on a large dataset, 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 trade-off between achieved privacy and data utility.

The paper is structured as follows. In Section II the privacy problem is presented through an example in the medical domain that illustrates the privacy attacks enabled by background knowledge, and the inadequacy of state of the art techniques. In Section III we formally model the privacy attack, as well as the considered forms of background knowledge. In Section IV we show how an adversary can actually extract background knowledge, and revise his knowledge in order to perform the attack. In Section V we propose our JS-reduce defense algorithm that is experimentally evaluated in Section VI. Section VII concludes the paper.

**II. 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 example also shows the inadequacy of state of the art techniques, and serves as a running example for the rest of the paper.

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. 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).

We consider analysis that require individual transactions; i.e., no aggregation is allowed.

We consider analysis that require individual transactions; i.e., no aggregation is allowed.  

\[ BK^{\text{seq}} \] 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”. \( BK^{\text{seq}} \) 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 \( BK^{\text{seq}} \) is reported in Table II(b) 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 IV-A both

\[ 1 \] We consider analysis that require individual transactions; i.e., no aggregation is allowed.

\[ 2 \] MAM = mammography, CX = chest X-ray, BCM = breast cancer marker, PNE = pneumonia
TABLE II
ADVERSARY’S BACKGROUND KNOWLEDGE

(a) Sensitive values background knowledge at \( \tau_1 \)

| 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 | 0.00003       |
| Doris | 52  | F      | 12031     | CX-pos | 0.00003       |
| 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 \( \tau_1 \) | Ex-res at \( \tau_2 \) | \( p(\text{Ex-res} | \tau_1) \) |
|------------------------|------------------------|----------------|
| 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           |

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 \( \tau_1 \) the adversary’s confidence about the exam results of tuples respondents after observing the data released at time \( \tau_1 \) (e.g., “The probability that Alice is the respondent of a tuple with Ex-res = MAM-pos released at \( \tau_1 \) is 0.5”).

Consider the original transaction data at time \( \tau_1 \) (first week) and \( \tau_2 \) (second week) shown in Tables II(a) and II(c), respectively, and the corresponding generalized transaction data in Tables II(b) and II(d). Note that these generalized views satisfy state of the art techniques for privacy preservation. In particular, they satisfy \( l \)-diversity \( [10] \) with \( l = 2 \), \( m \)-invariance \( [1] \) with \( m = 2 \), as well as the privacy properties proposed in \([4, 5, 11]\). However, we show that the release of these views can lead to a serious privacy threat. Consider tuples released at \( \tau_1 \) 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 adversary cannot exploit \( BK^{sv} \) (reported in Table II(a)) to infer whether Alice or Betty is the respondent of the tuple with private value MAM-pos. Hence, his posterior knowledge after having observed tuples released at \( \tau_1 \) 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 \( \tau_2 \) (in Table II(d) 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 \( \tau_1 \) and \( BK^{seq} \) to derive a new kind of knowledge, which we name revised sensitive values background knowledge (\( RBK^{sv} \)) at \( \tau_2 \). This knowledge represents the revision of sensitive values background knowledge computed based on the history of released views, and on sequential background knowledge. The actual method for computing \( RBK^{sv} \) is shown in Section IV; here we give an intuition of the adversary reasoning. Since the exam result of Alice at \( \tau_1 \) is either MAM-pos or CX-neg, and the one at \( \tau_2 \) is either BCM-pos or PNE-pos, 4 possible sequences of sensitive values about Alice exist. Among these sequences, according to \( BK^{seq} \), the one having MAM-pos at \( \tau_1 \) and BCM-pos at \( \tau_2 \) 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 \( \tau_1 \) and PNE-pos at \( \tau_2 \). Through this kind of reasoning the adversary revises his sensitive values background knowledge, associating high confidence to the fact that at \( \tau_2 \) 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.

III. MODELLING ATTACKS BASED ON BACKGROUND AND REVISED KNOWLEDGE

In this section we formally model privacy attacks based on background and revised knowledge available to an adversary.

A. Problem definition

We denote by \( V_i \) a view on the original transaction data at time \( \tau_i \), and by \( V_i^* \) the generalization of \( V_i \) released by the data publisher. We denote by \( H_j = (V_1^*, V_2^*, \ldots, V_j^*) \) a history of released generalized views. We assume that the schema remains unchanged throughout the release history, and we partition the view columns into a set \( A^0 = \{A_1, A_2, \ldots, A_m\} \) of quasi-identifier attributes, and into a single private attribute \( S \). For the sake of 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, \( t[A] \) the projection of tuple \( t \) onto \( A \).

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 $G(\cdot)$ is publicly known.
- The adversary may have external information about respondents’ personal data. For example, for each QL-group $Q$, the adversary may know its set of respondents.
- The adversary may observe a history $\mathcal{H}_j^∗$ of anonymized views.
- The adversary may have background knowledge on sensitive values $BK_{sv}$ and $BK_{seq}$ as formally defined in Sections III-B and III-C, respectively.

Note that the first two assumptions are shared by most work on anonymity. As illustrated in Section III, 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.

B. Sensitive values background knowledge ($BK_{sv}$)

Sensitive values background knowledge represents the a-priori probability of associating an individual to a sensitive value. $BK_{sv}$ 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

$$\Upsilon = \{ (p_1, \ldots, p_n) \mid \sum_{1 \leq i \leq n} p_i = 1 \ (0 \leq p_i \leq 1) \}$$

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 $s_j$.

C. Sequential background knowledge ($BK_{seq}$)

We model the sensitive value referring to a respondent $r$ by means of the discrete random variable $S$ having values in $D[S]$. Hence, sequential background knowledge is a function that returns the probability distribution of $S$ at $t_j$ given a sequence $\Lambda = (s_1, s_2, \ldots, s_j)$ of past observations at $T = (t_1, t_2, \ldots, t_{j-1})$.

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

D. Posterior ($PK_{sv}$) and revised sensitive values background knowledge ($RBK_{sv}$)

As intuitively described in the running example of Section III, posterior knowledge at $t_j$ represents the adversary’s confidence about the association between a respondent and sensitive values after the observation of view $V_j^∗$. For the sake of readability, we denote $PK_{sv}^\tau$ at $t_j$ by $PK_{sv}^\tau$.

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

A method to compute $PK_{sv}^\tau$ is described in Section IV-B.

After observing view $V_{j-1}^*$, an adversary may exploit posterior knowledge at $t_1, t_2, \ldots, t_{j-1}$, together with sequential background knowledge $BK_{seq}$, to derive new information about the probability distribution of $S$ at $t_j$. We call this information **revised sensitive values background knowledge** at $t_j$ (denoted as $RBK_{sv}^\tau$); it is essentially the revision of sensitive values background knowledge due to the observation of a history of released tuples. $RBK_{sv}^\tau$ can be used by an adversary to calculate posterior knowledge after the observation of $V_{j}^*$.

The revised sensitive values background knowledge is a function $RBK_{sv}^\tau$ having the same domain and co-domain as function $PK_{sv}^\tau$ defined in Definition 3. The method to compute $RBK_{sv}^\tau$ is described in Section IV-C.

E. The privacy attack

The inference method adopted by an adversary to reconstruct the sensitive association is depicted in Figure 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 IV-A. When the first view $V_1^*$ is released at time $t_1$, the adversary computes posterior knowledge $PK_{sv}^1$ based on $V_1^*$ and on $BK_{sv}$, a method for posterior knowledge computation is presented in Section IV-B. Then, the adversary computes revised sensitive values background knowledge $RBK_{sv}^1$, based on $PK_{sv}^1$ and on sequential background knowledge $BK_{seq}$. A technique for
knowledge revision is illustrated in Section IV-C. Hence, when view \( V^*_2 \) is released, the adversary computes \( PK^sv_2 \) based on \( V^*_2 \) and on \( RBK^sv_2 \). Then, the knowledge revision cycle continues with the computation of \( RBK^sv_2 \) based on \( PK^sv_2 \) and \( BK^seq \), and so on. When \( V^*_i \) includes a tuple of respondent \( r \), and no tuples of \( r \) appeared in \( H_{t-1}^r \), \( RBK^sv_2(r, \tau_i) \) cannot be computed, since no historical information about \( r \)’s tuples is available; in this case \( BK^sv \) is used instead of \( RBK^sv(r, \tau_i) \).

IV. 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.

A. 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 [7], [8], [9]). 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 [7], [9]. At the time of writing, these techniques are limited to the calculation of \( BK^sv \). However, based on a sequence \( H_i \) of original views, sequential pattern mining (SPM) methods [12] can be used to calculate a function \( IE-BK^seq \) that approximates the exact \( BK^seq \). That function is incrementally refined as long as new original views are available. A number of different SPM techniques have been proposed in the last years for different application domains (e.g., [13], [14], [15], among many others). Hence, the choice of the most appropriate SPM algorithm strongly depends on the domain of the data. In Section IV-C we illustrate the algorithm we adopted to calculate \( IE-BK^seq \) 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 \( SPM-BK^seq \) 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 (e.g., [16], [17], [18], [19], just to name a few). Given this knowledge, it is easy to design a function \( DK-BK^seq \), which approximates the exact \( BK^seq \).

B. Computing posterior knowledge

In order to compute \( PK^sv \), 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 \( c : Q \rightarrow R \), i.e., a one-to-one correspondence between elements in \( Q \) and elements in \( R \). 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 III(a) released at \( \tau_2 \) 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 \( R BK^sv \) (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:

\[
p_m = \frac{\sum_{\forall c_j \in C, c_j(r) = r \land \{c_j\}[s] = s_m} d_j}{\sum_{\forall c_j \in C} d_j}.
\]

Example 2: Continuing Example 1 according to \( RBK^sv_2 \) (Table III(b)), 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 \( p_m = \frac{0.62}{0.62 + 0.07} \approx 0.9 \). The values of \( PK^sv \) at \( \tau_2 \) are shown in Table III(c).

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 \( k! \). For this reason, an
TABLE III
Adversary’s posterior and revised knowledge

| 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|

(a) $PK^{sv}$ at $\tau_1$

(b) $RBK^{sv}$ at $\tau_2$

(c) $PK^{sv}$ at $\tau_2$

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

approximate algorithm is the natural candidate for the computation of posterior knowledge. In our experimental evaluation, we calculate posterior knowledge by the $\Omega$-estimate method proposed by Li et al. [9].

C. Computing revised knowledge

In order to compute revised sensitive values background knowledge at $\tau_1$ ($i > 1$) 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 $BK^{svq}$. Let $V^*_r = \{V^*_1, V^*_2, \ldots, V^*_s\}$ be the history of released views containing a tuple of $r$, and $S_i$ the random variable representing the sensitive value of $r$’s tuple released at $\tau_1$. Then, by applying the conditioning rule, we have:

$$P(S_i) = \sum_{\lambda \in \Lambda} (BK^{svq}(\lambda, T, r, \tau_1) \cdot P(\lambda)),$$

where $T = \langle \tau_1, \tau_2, \ldots, \tau_{i-1} \rangle$, $\Lambda$ 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 $\tau_j$ 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:

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

Example 3: Considering our running example, the adversary reuses his sensitive values background knowledge after observing view $V^*_2$ to obtain $RBK^{sv}_2$ as follows. The probability $p(Alice, s, \tau_1)$ that Alice is the respondent of a tuple released at $\tau_1$ having sensitive value $s$ is given by $PK^{sv}_1$ (Table III(a)). Moreover, we represent by $\bar{p}(BCM-pos \mid s)$ the probability that an individual is the respondent of a tuple released at $\tau_2$ with sensitive value BCM-pos provided that the same individual was the respondent of a tuple released at $\tau_1$ with sensitive value $s$; this conditional probability is given by $BK^{svq}$ (Table II(b)). Then, the marginal probability of Alice to be the respondent of one tuple with BCM-pos at $\tau_2$ can be calculated as:

$$p(Alice, BCM-pos, \tau_2) =$$

$$= \sum_{s \in D[S]} \left( p(Alice, s, \tau_1) \cdot \bar{p}(BCM-pos \mid s) \right) =$$

$$= p(Alice, MAM-pos, \tau_1) \cdot \bar{p}(BCM-pos \mid MAM-pos) +$$

$$+ p(Alice, CX-neg, \tau_1) \cdot \bar{p}(BCM-pos \mid CX-neg) =$$

$$= 0.5 \cdot 0.6 + 0.5 \cdot 0.02 = 0.31.$$

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 $PK^{sv}_1$ is 0. Analogously, the adversary calculates that, according to $RBK^{sv}_2$, 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 III(b)).

V. JS-reduce defense

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

A. 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 IV-B and to the attack we are considering, this means reducing the confidence of the adversary in discriminating a configuration $\tilde{c}$ among the possible ones, based on his knowledge $RBK^{sv}$. The goal of JS-reduce is to create QI-groups whose tuple respondents have similar $RBK^{sv}(BRK^{sv})$ distributions. Indeed, if the respondents of tuples in a QI-group are indistinguishable with respect to $RBK^{sv}(BRK^{sv})$, 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 $k$-closeness, in order to protect against well-known identity- and attribute-disclosure attacks, respectively. Note that JS-reduce can be easily extended to enforce additional privacy models.

B. Defending against sequential background knowledge attacks

In order to measure the similarity of probability distributions $RBK^{sv}(BRK^{sv})$, we adopt Jensen-Shannon divergence (JS) [20]. With respect to other distance measures among probability distributions, this function has three important properties: i) it can be computed on a set of more than two distributions; ii) it is always a definite number; iii) it is symmetric with respect to the order of the arguments. Suppose that $P = \{p_1, \ldots, p_n\}$ is a set of probability distributions such that each element has form: $p_i = \langle p_{i1}, \ldots, p_{in} \rangle$. Suppose also that $\pi^1, \ldots, \pi^n$ denote the weights of the probability
distributions, and that \( \sum_{i=1}^{n} \pi_i = 1 \). Then the JS divergence among distributions in \( P \) is:

\[
JS(P) = H(\pi) - \sum_{i=1}^{n} \pi_i \cdot H(\pi^i),
\]

where \( H(\pi) \) is the Shannon entropy of \( \pi = (p_1, \ldots, p_n) \). In our case, each \( \pi^i \) corresponds to the background knowledge about a tuple respondent; since this probability \( \pi^i \) 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 without anonymizing data. Hence, similarly to the knowledge revision cycle presented in Section IV, the defense technique (graphically illustrated in Figure 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 IV-A.

C. The JS-reduce algorithm

The pseudo-code of the JS-reduce algorithm is shown in Algorithm 1. The algorithm takes as input: i) a sequence \( H_n = (V_1, \ldots, V_n) \) of original views; ii) the set \( R \) of respondents of tuples in \( H_n \), as well as their QI values; iii) sensitive values background knowledge \( BK^sv \) and sequential background knowledge \( BK^{seq} \); iv) the minimum level \( k \) of \( k \)-anonymity, threshold \( t \) of \( t \)-closeness, and threshold \( j \) of JS divergence.

Input: Sequence \( H_n = (V_1, \ldots, V_n) \), 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: \( V_n^* \)

1) JS-reduce \((H_n, R, BK^sv, BK^{seq}, k, t, j) \)

2) begin

3) forall \( r \in R \) do

4) \( RBK^sv(r) \leftarrow BK^sv(r) \)

5) end

6) forall \( h = 1 \) to \( n \) do

7) \( V_n^r \leftarrow \text{Generalize} \( V_h, RBK^sv_h, t, j, k \) \)

8) forall \( r \in R_h \) do

9) \( PK_{Computation}(V^*, RBK^sv_h, r) \)

10) \( RBK^sv_{h+1}(r) \leftarrow \text{BKRevision}(PK_{Computation}(r), BK^{seq}_h, r) \)

11) end

12) end

13) return \( V_n^* \)

14) end

Input: The anonymized release \( V_n^* \), the set \( RBK^sv_h \) of revised background knowledge for each respondent of a tuple in \( V_n^* \), respondent \( r \)

Output: \( PK^{Computation}(V^*, RBK^{seq}_h, r) \)

1) PKComputation \((V^*, RBK^{seq}_h, r) \)

2) begin

3) QI-group \( Q \leftarrow Q' \times V_n^* \) s.t. \( r \) is the respondent of one tuple in \( Q' \)

4) \( C \leftarrow \{ c_j | \text{c_j is a valid configuration for} Q \} \)

5) forall \( c_j \in C \) do

6) confidence degree \( d_j \leftarrow 0 \)

7) forall \( r' \cdot s.t. \exists r \in Q | c_j(t) = r' \) do

8) \( d_j \leftarrow d_j + RBK^sv_h(r', t[S]) \)

9) end

10) forall \( s \in D[S] \) do

11) \( p(r, s) \leftarrow \frac{\sum_{v \in \text{c_j}(t) \in c_j(t)=r' \cdot s.t. \exists r \in Q | c_j(t) = r'}{\sum_{v \in \text{c_j}(t) \in c_j(t)=r' \cdot s.t. \exists r \in Q | c_j(t) = r'}} \)

12) end

13) return \( PK^{Computation}(r) \)

14) end

Input: The set of posterior knowledge of respondent \( r \)

\( PK^{Computation}(r) \leftarrow \{ PK^{Computation}(r) \} \)

Output: \( PK^{seq}_h(r) \)

1) BKRevision \((PK^{Computation}(r), BK^{seq}_h, r) \)

2) begin

3) \( A \leftarrow \{ A = (s_1, \ldots, s_k) | s_j \text{ is a possible sensitive value for} r \text{ released at} \tau_j \} \)

4) forall \( \lambda \in A \) do

5) \( P(\lambda) \leftarrow 1 \)

6) forall \( s_j \in \lambda \) do

7) \( P(\lambda) \leftarrow P(\lambda) \cdot PK^{Computation}(r, s_j) \)

8) end

9) forall \( s \in D[S] \) do

10) \( \bar{p}(s | \lambda) \) is the conditional probability given by \( BK^{seq} \)

11) \( p(s) \leftarrow \sum_{\lambda \in A} \bar{p}(s | \lambda) \cdot P(\lambda) \)

12) end

13) \( RBK^sv_h(r) \leftarrow \{ p(s), \forall s \in D[S] \} \)

14) return \( RBK^sv_h(r) \)

15) end

Algorithm 1: JS-reduce algorithm
PKComputation receives as input: the code is shown in Algorithm 2. The reason, we devised an approximate algorithm, whose pseudo-code is shown in Algorithm 2. The Generalize procedure receives as input: i) the original view $V_h$; ii) revised sensitive values background knowledge at $\tau_h$; iii) 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 $V_h$.

As proposed in [23], 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 (lines 4 to 6) computes its Hilbert index considering the multi-dimensional space having the QI attributes as dimensions. Then, tuples in $V_h$ are re-ordered with respect to their Hilbert index, obtaining an auxiliary list $\tilde{V}_h$ (line 7). The procedure adds to a group $Q$ a tuple from the ordered list $\tilde{V}_h$, and checks if the cardinality of the group is greater than the $k$-anonymity threshold $k$, and if the $t$-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 $\tilde{V}_h$, 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: i) it is likely to find groups of tuples satisfying privacy constraints, and ii) we limit the generalization of QI values. Then, if the required privacy constraints are satisfied, a new QI-group is created (line 12) by procedure CreateQIG: the QI values are substituted with intervals including the QI values of each tuple; the same procedure is repeated with the remaining tuples. Otherwise (if constraints are violated), the next tuple in $\tilde{V}_h$ is added to the group until the constraints are satisfied (line 10).

As explained in Section V, it may happen that a few tuples cannot be grouped into a QI-group (line 16) during the first phase. In the current version of the algorithm, those tuples are suppressed in order to guarantee the privacy constraints in the whole view. However, the algorithm can be easily modified to apply other solutions; e.g., based on the creation of counterfeit tuples.

### VI. EXPERIMENTAL EVALUATION

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

#### A. Experimental setup

To the best of our knowledge, all the datasets used for experimental evaluation of proposed privacy defenses for serial data publication were created from non-temporally characterized sets of tuples, in which each tuple was randomly assigned to a release. Clearly, these datasets are not realistic for

| l | t | B | j |
|---|---|---|---|
| 1-div. | [2.8] | 0.2 | - |
| 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, 0.6] | 0.6 |

### TABLE IV

PRIVACY PARAMETERS USED IN THE EXPERIMENTS
investigating the use that an adversary can make of temporal correlations. The dataset used in our experiments has been synthetically created based on domain knowledge extracted from the medical literature; in particular, studies reported in \[16, 17, 18, 19\]. 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 $\tau_j$ to obtain a given result $ex-res_i$ given a sequence of results of exams performed by that person in the previous weeks. $BK^{sv}$ was calculated dividing age and weight into 3 sub-intervals (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 dataset has been made available from our group and can be used to replicate our experiments, or as a testbed for any research about sequential background knowledge\[1\].

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 dataset 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 3 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 (5 stages of liver disease, 4 of the HIV syndrome, 3 of Alzheimer, and 5 of sepsis), as well as two sensitive values to describe the deceased and discharged events.

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: a) distinct $l$-diversity (each QI-group must contain at least $l$ tuples having different sensitive values), b) $t$-closeness \[24\], and c) $(B,t)$-privacy \[9\]. We used the Mondrian framework \[21\] to generalize the views in the history according to each of the latter methods, while we used Algorithm 1 to apply the JS-reduce defense. Experiments were performed on a 2.4GHz workstation with 4GB RAM. The time required for anonymizing a view with the JS-reduce algorithm varied from a few minutes to a maximum of 43 minutes, depending on the chosen privacy parameters; this is an acceptable time since in many cases anonymization is performed offline.

For each considered technique, we made experiments with different values of the corresponding privacy parameters. Figure 3 shows the average semiperimeter of QI-groups generated by the different techniques using the values shown in Table IV (bold numbers indicate the parameters used in the following experiments). A smaller semiperimeter corresponds to a better quality of released data.

\begin{figure}
\centering
\begin{subfigure}{0.24\textwidth}
    \centering
    \includegraphics[width=\textwidth]{l-div.png}
    \caption{(a) $l$-div.}
\end{subfigure}
\hspace{0.05\textwidth}
\begin{subfigure}{0.24\textwidth}
    \centering
    \includegraphics[width=\textwidth]{t-clos.png}
    \caption{(b) $t$-clos.}
\end{subfigure}
\hspace{0.05\textwidth}
\begin{subfigure}{0.24\textwidth}
    \centering
    \includegraphics[width=\textwidth]{b-t-priv.png}
    \caption{(B,t)-priv.}
\end{subfigure}
\hspace{0.05\textwidth}
\begin{subfigure}{0.24\textwidth}
    \centering
    \includegraphics[width=\textwidth]{js-red.png}
    \caption{JS-red.}
\end{subfigure}
\caption{QI generalization}
\end{figure}

\begin{algorithm}
\caption{SPM-$BK^{seq}$ extraction}
\begin{algorithmic}[1]
\Input{History of original views $H_r = \langle V_1, \ldots, V_r \rangle$, 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 $H_r$.}
\State $SPM(H_r, seq, s)$ begin
\For{$h = 1$ to $r$}
\ForAll{all respondent $u$ of a tuple in $V_h$}
\For{$j = h$ to $1$}
\State $seq_j = seq.$ of past $j$ sensitive values of $u$ in $H_h$
\EndFor
\State $seq.numOcc = seq_j.numOcc + 1$
\EndFor
\EndFor
\End
\For{$(seq.numOcc == 0)$ then return 0}
\Else\State $sequence = (seq, s)$
\State $sequence.numOcc$
\EndFor
\end{algorithmic}
\end{algorithm}

B. 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 $\tau_i$ containing $N$ tuples, we measured the average

\[\text{average}\]

\footnote{The semiperimeter of a QI-group is the sum of the normalized lengths of the interval of each QI value of tuples in it.}
adversary gain $\rho$ as follows:

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

where: $p(r_j, s_{ij}, \tau_i)$ is the value of posterior knowledge computed based on background knowledge for respondent $r_j$ and her actual private value $s_{ij}$ at $\tau_i$; $Q_{ij}$ is the QI-group of $V_i^*$ containing the tuple whose respondent is $r_j$; and $m(s_{ij})$ is the number of tuples $t$ in $Q_{ij}$ such that $t[S_i] = s_{ij}$. Intuitively, the adversary gain represents the amount of information obtained with the use of background knowledge with respect to a privacy attack based only on the observation of the frequency of sensitive values in the QI-group.

C. 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-BK_{seq}**. Since it was the subject of related studies (e.g., [7], [9]), the first kind of background knowledge we consider is the one directly extracted from the data to be released. IE-BK_{seq} can be calculated by applying sequential pattern mining (SPM) techniques on the history of original (i.e., non-anonymized) data; at each time $\tau_i$, IE-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 [3].

- **Mined knowledge SPM-BK_{seq}**. In practice, an adversary may approximate BK_{seq} by applying SPM techniques on an external corpus of non-anonymized data. We created a data corpus using the same model that we used to generate our dataset; the corpus consists in a history of 24 views containing 5,000 tuples each. SPM-BK_{seq} was calculated by applying Algorithm [3] to that corpus.

- **Domain knowledge DK-BK_{seq}**. Since the dataset we used was generated based on domain knowledge, in our experiments DK-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 domain knowledge may only approximate the exact BK_{seq}. Hence, we also considered another kind of domain knowledge, whose temporal extent is limited to a number $n$ of past observations. We denote this knowledge as $n$-steps DK-BK_{seq}, and we consider $n = 1$, $n = 2$, and $n = 3$.

Figure [4] shows the adversary gain when views are anonymized using existing techniques, and the adversary may exploit the different kinds of sequential background knowledge. Results show that existing techniques are not effective against the attacks identified in this paper. Indeed, with each kind of background knowledge, the adversary gain grows very rapidly during the first 6/8 releases, exceeding the value of 0.4.

For each considered anonymization technique, the form of background knowledge that determines the highest adversary gain is full DK-BK_{seq}, since in our experiments it corresponds to the exact BK_{seq}. Hence, we considered approximate DK-BK_{seq} in order to better evaluate the role of domain knowledge. Results illustrated in Figures [5(a)] and [5(b)] show that even attacks based on approximate DK-BK_{seq} are effective against existing anonymization techniques; attacks exploiting 3-steps DK-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_{seq} (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 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, having a different temporal characterization.

D. Effectiveness of the JS-reduce defense

Experimental results reported in Figure [5(c)] show that, when views are anonymized with the JS-reduce technique, 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 Figure [6(a)] we considered the case in which the defender has background knowledge DK-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-SPM-BK_{seq} in Figure [6] we denote that the adversary’s SPM-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-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-BK_{seq}.

We also considered the unfortunate case in which the adversary has more accurate background knowledge than the defender. Results illustrated in Figures [6(b)] and [6(c)] show the adversary gain when the defender’s background knowledge is IE-BK_{seq} and SPM-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 sensible privacy protection even in the worst case; indeed, the adversary gain always 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 \( \tau_j \) the value of the posterior probability \( \mathbb{P}(r | \tau_j) \) computed by the adversary for the actual private value of \( r \) at \( \tau_j \). The average adversary confidence about a generalized view \( V_j^* \) is the average of the adversary confidence regarding respondents of tuples in \( V_j^* \). Figure 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 \( DK-BK^{seq} \)). 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, 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 Figure 9 as expected, the lower the JS threshold value, the lower the adversary gain.
E. 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 two well-known metrics: average semiperimeter, and Global Certainty Penalty (GCP) [25] (a metric taking into account the level of generalization of QI values). Figure 3 shows the average semiperimeter of QI-groups generated by the considered techniques (JS-reduce is based on $DK-BK_{seq}$). As it can be seen, JS-reduce outperforms the other techniques. These results are confirmed by a comparison in terms of GCP (Figure 8(a)).

Then, we compared the utility of transaction data generalized by the different techniques in terms of the 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. The imprecision in query answering was calculated in terms of the median error. The results reported in Figure 8(b) show the superiority of JS-reduce with respect to the other techniques; this result is due to the use of the data quality-oriented generalization algorithm presented in Section VII-D.

Finally, we evaluated the number of tuples that were suppressed by JS-reduce in order to enforce the privacy requirements. Results show that a very few number of tuples were suppressed; i.e., at most 12 ($< 0.25\%$) at each release.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we demonstrated that the correlation of sensitive values in subsequent data releases can be used as adversarial background knowledge to violate users’ privacy. We showed that an adversary can actually obtain this knowledge by different methods. Since serial release of transaction data is a common situation, the considered problem poses a very practical challenge. We proposed a defense algorithm based on Jensen-Shannon divergence, and we showed through an extensive experimental evaluation that other applicable solutions are not effective, while our JS-reduce defense provides strong privacy protection and good data quality, even when the adversary has more accurate background knowledge than the defender.

Future work includes studying the effect on privacy preservation of compromised tuples; i.e., possibly very few tuples whose respondent is known to the adversary. Moreover, specific application domains (e.g., streaming data) often require anonymization to be performed online; hence, a further line of investigation consists in devising protection techniques having very low computational complexity.

ACKNOWLEDGMENTS

The authors would like to thank Kristen LeFevre for providing an implementation of the Mondrian framework; Tiancheng Li, Ninghui Li and Jian Zhang for providing software modules for $(B,t)$-privacy; and Andrea Bianchini for his extensive programming work.

REFERENCES

[1] X. Xiao and Y. Tao, “m-Invariance: Towards Privacy Preserving Republification of Dynamic Datasets,” in Proc. of SIGMOD’07. ACM Pub., 2007, pp. 689–700.
[2] B. C. M. Fung, K. Wang, A. W.-C. Fu, and J. Pei, “Anonymity for Continuous Data Publishing,” in Proc. of EDBT’08. ACM Pub., 2008, pp. 264–275.
[3] J. Li, B. C. Ooi, and W. Wang, “Anonymizing Streaming Data for Privacy Protection,” in Proc. of ICDE’08. IEEE Comp. Soc., 2008, pp. 1367–1369.
[4] B. Zhou, Y. Han, J. Pei, B. Jiang, Y. Tao, and Y. Jia, “Continuous Privacy Preserving Publishing of Data Streams,” in Proc. of EDBT’09. ACM Pub., 2009, pp. 648–659.
Y. Bu, A. W. C. Fu, R. C. W. Wong, L. Chen, and J. Li, “Privacy Preserving Serial Data Publishing by Role Composition,” in Proc. of VLDB’08. VLDB Endow., 2008, pp. 845–856.

R. C.-W. Wong, A. W.-C. Fu, J. Liu, K. Wang, and Y. Xu, “Global privacy guarantee in serial data publishing,” in Proc. of ICDE’10. IEEE Comp. Soc., 2010, pp. 956–959.

T. Li and N. Li, “Injector: Mining Background Knowledge for Data Anonymization,” in Proc. of ICDE’08. IEEE Comp. Soc., 2008, pp. 446–455.

R. Agrawal and R. Srikant, “Mining sequential patterns,” in Proc. of ICDE’95. IEEE Computer Society, 1995, pp. 3–14.

J. Ayres, J. Flannick, J. Gehrke, and T. Yu, “Sequential pattern mining using a bitmap representation,” in Proc. of SIGKDD’02. ACM, 2002, pp. 429–435.

K. Wang, Y. Xu, and J. X. Yu, “Scalable sequential pattern mining for biological sequences,” in Proc. of CIKM’04. ACM, 2004, pp. 178–187.

J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu, “Mining sequential patterns by pattern-growth: The prefixspan approach,” IEEE Trans. Knowl. Data Eng., vol. 16, no. 11, pp. 1424–1440, 2004.

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,” Clin. Inf. Dis., vol. 27, no. 1, pp. 185–190, 1998.

G. D. Biase, G. D’Amico, A. D. Girolamo, J. Janssen, S. Iacobelli, N. Tinari, and R. Manca, “A Stochastic Model for the HIV/AIDS Dynamic Evolution,” Math. Prob. Eng., 2007.

R. S. Remis, “A Study to Characterize the Epidemiology of Hepatitis C Infection in Canada,” Pub. Health Agency of Canada, Tech. Rep., 2002.

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. J. of Geriatr. Psychiatry, vol. 19, no. 3, pp. 266–270, 2004.

J. Lin, “Divergence Measures based on the Shannon Entropy,” IEEE Trans. Inf. Theory, vol. 37, no. 1, pp. 145–151, 1991.

K. LeFevre, D. J. DeWitt, and R. Raghu, “Mondrian Multidimensional k-Anonymity,” in Proc. of ICDE’06. IEEE Comp. Soc., 2006.

A. Meyerson and R. Williams, “On the Complexity of Optimal k-Anonymity,” in Proc. of PODS’04. ACM Pub., 2004, pp. 223–228.

G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, “Fast Data Anonymization with Low Information Loss,” in Proc. of VLDB’07. ACM Pub., 2007, pp. 758–769.

N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,” in Proc. of ICDE’07. IEEE Comp. Soc., 2007, pp. 106–115.

J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu, “Utility-based Anonymization Using Local Recoding,” in Proc. of KDD’06. ACM Pub., 2006, pp. 785–790.