TLKC-Privacy Model for Process Mining

Majid Rafiei[0000−0001−7161−6927]✉, Miriam Wagner[0000−0002−6941−037X], and Wil M.P. van der Aalst[0000−0002−0955−6940]

Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany

Abstract. Process mining aims to provide insights into the actual processes based on event data. These data are widely available and often contain private information about individuals. Consider for example healthcare information systems recording highly sensitive data related to diagnosis and treatment activities. Process mining should reveal insights in the form of annotated models, yet, at the same time, should not reveal sensitive information about individuals. In this paper, we discuss the challenges regarding directly applying existing well-known privacy-preserving techniques to event data. We introduce the $TLKC$-privacy model for process mining that provides privacy guarantees in terms of group-based anonymization. It extends and customizes the LKC-privacy model presented to deal with high-dimensional, sparse, and sequential trajectory data. Experiments on real-life event data demonstrate that our privacy model maintains a high utility for process discovery and performance analyses while preserving the privacy of the cases.

Keywords: Responsible process mining · Privacy preservation · Process discovery · Performance analyses

1 Introduction

Event logs are used by process mining algorithms to discover and analyze the real processes. An event log is a collection of events and such information is widely available in current information systems [1]. Each event is described by its attributes and typical attributes required for process mining algorithms are *case id*, *activity*, *timestamp*, and *resource*. The minimal requirements for process mining are that any event can be related to both a case and activity and that the events that belong to a case are ordered, which is often done by means of timestamps [1]. Therefore, *timestamps* play a crucial role in process mining algorithms and need to be stored and processed. However, the event data containing accurate timestamps (in milliseconds) are highly sensitive.

Moreover, some of the event attributes may refer to individuals, e.g., in the health-care context, the *case id* may refer to the patient whose data is recorded, and the *resource* may refer to the employees performing activities for the patients, e.g., nurses or surgeons. When the individuals' data are explicitly or implicitly included, privacy issues arise. According to regulations such as the European General Data Protection Regulation (GDPR) [21], organizations are obliged to consider the privacy of individuals.

Regarding the four main attributes of events, two different perspectives for privacy in process mining can be considered; *resource perspective* and *case perspective*. The *resource perspective* refers to the privacy of the individuals performing the activities, and the *case perspective* considers the privacy of the individuals whose data is recorded and analyzed. Depending on the context, the relative importance of these perspectives may vary. However, often the *case perspective* is more important than the *resource perspective*. For example, in the health-care context, the activity performers could be publicly available. However, what happens for a specific patient and her/his personal information should be kept private. In this paper, we focus on the *case perspective*.

There are many activities and techniques in process mining such as *process discovery*, *conformance checking*, *social network analyses*, *prediction*, etc. However, the three basic types of process mining are; *process discovery*, *conformance checking*, and *enhancement* [1]. The proposed privacy model focuses on process discovery and a subfield of enhancement called performance analyses. Since the event data used by process mining algorithms are high-dimensional sparse data, privacy preservation with high data utility is significantly challenging.

The aim of this paper is to provide a privacy-preserving model for process mining protecting the privacy of *cases*, yet, at the same time, maintains the utility of the process discovery and performance analyses. The utility is preserved in terms of similarity of the results provided by the privacy-preserving approach to the results obtained from the original data. We introduce $TLKC$-privacy model, which exploits some restrictions regarding the availability of the background knowledge in the real world to deal with process mining-specific challenges. Our model is an extension for the LKC-privacy model [16,8], which was presented to deal with privacy challenges of the trajectory data. The LKC-privacy model generalizes several traditional privacy models, such as k-anonymity, confidence bounding, (α,k)-anonymity, and l-diversity, which are inherited by our model. We evaluate our approach with respect to the typical trade-off between privacy guarantees and the loss of accuracy. The approach is evaluated on a real-life event data belonging to a hospital (Sepsis) containing infrequent behavior. Our experiments show that our approach maintains a high utility, assuming realistic background knowledge while using tunable privacy parameters.

The rest of the paper is organized as follows. In Section 2, we explain the motivation and challenges. In Section 3, formal models are presented for event log and attack scenarios. We explain the $TLKC$-privacy model in Section 4. In Section 5, the implementation and evaluation are described. Section 6 outlines related work, and Section 7 concludes the paper.

2 Motivation

To motivate the necessity to deal with privacy issues in process mining, we illustrate the problem with an example in health-care context. Suppose that Table 1 shows a part of an event log recorded by an information system in a hospital. Assuming that an adversary knows that patient's data are in the

Table 1: Sample event log (each row represents an event).

Case Id	Activity	Timestamp	Resource	Age	Disease
1	Registration (RE)	01.01.2019-08:30:00	Employee1	22	Flu
1	Visit (V)	01.01.2019-08:45:00	Doctor1	22	Flu
2	Registration (RE)	01.01.2019-08:46:00	Employee1	30	Infection
3	Registration (RE)	01.01.2019-08:50:00	Employee1	32	Infection
4	Registration (RE)	01.01.2019-08:55:00	Employee4	29	Poisoning
1	Release (RL)	01.01.2019-08:58:00	Employee2	22	Flu
5	Registration (RE)	01.01.2019-09:00:00	Employee1	35	Cancer
6	Registration (RE)	01.01.2019-09:05:00	Employee4	35	Hypotension
4	Visit (V)	01.01.2019-09:10:00	Doctor2	29	Poisoning
5	Visit (V)	01.01.2019-09:20:00	Doctor4	35	Cancer
4	Infusion (IN)	01.01.2019-09:30:00	Nurse2	29	Poisoning
2	Hospitalization (HO)	01.01.2019-09:46:00	Employee3	30	Infection
3	Hospitalization (HO)	01.01.2019-10:00:00	Employee3	32	Infection
5	Hospitalization (HO)	01.01.2019-09:55:00	Employee6	35	Cancer
2	Blood Test (BT)	01.01.2019-10:00:00	Nurse1	30	Infection
5	Blood Test (BT)	01.01.2019-10:10:00	Nurse2	35	Cancer
3	Blood Test (BT)	01.01.2019-10:15:00	Nurse1	32	Infection
6	Visit (V)	01.01.2019-10:20:00	Doctor3	35	Hypotension
4	Release (RL)	01.01.2019-10:30:00	Employee2	29	Poisoning
6	Release (RL)	01.01.2019-14:20:00	Employee2	35	Hypotension
2	Blood Test (BT)	01.02.2019-08:00:00	Nurse2	30	Infection
2	Visit (V)	01.02.2019-10:00:00	Doctor2	30	Infection
3	Visit (V)	01.02.2019-10:15:00	Doctor3	32	Infection
2	Release (RL)	01.02.2019-14:00:00	Employee2	30	Infection
3	Release (RL)	01.02.2019-14:15:00	Employee5	32	Infection
5	Release (RL)	01.02.2019-16:00:00	Employee5	35	Cancer

event log (as a *case*), with little information about the activities having been done for the patient, the adversary is easily able to connect the patient to the corresponding *Case Id* and find the complete sequence of activities having been performed for the patient. For example, if the adversary knows that two blood tests have been performed for the patient, the only matching case is case 2. We call this attack *case linkage* attack. Note that the complete sequence of activities having been done for a patient is considered as the sensitive person-specific information which can be disclosed by the *case linkage* attack. Moreover, if we consider some attributes in the event log as sensitive, e.g., diagnosis and test results, the adversary can go further and link the sensitive information as well. For example, the disease that belongs to case 2 is infection. This attack is called *attribute linkage*. Note that the *attribute linkage* attack does not necessarily need to be done after the *case linkage*, i.e., if more than one case corresponds to the adversaries knowledge while all the cases have the same value as the sensitive attribute, the *attribute linkage* could happen without a successful *case linkage*.

Many privacy models, such as k-anonymity and its extensions [12], have been introduced to deal with the aforementioned attacks in the context of relational databases. In these privacy models, the data attributes are classified into four main categories including; *explicit identifier*, *quasi-identifier*, *sensitive attributes*, and *non-sensitive attributes*. The *explicit identifier* is a set of attributes containing information that explicitly identifies the data owner, the *quasi-identifier* is a set of attributes that could potentially identify the data owner, the *sensitive attributes* consist of sensitive person-specific information such as disease, and the *non-sensitive attributes* contain all the attributes that do not fall into the previous three categories [3]. In the group-based privacy models, the idea is to disorient potential linkages by generalizing the records into equivalence classes having the same values on the *quasi-identifier*. These privacy models are effec-

tive for anonymizing relational data. However, they are not easily applicable to event data due to some specific properties of event data.

In process mining, the *explicit identifiers* do not need to be stored and processed. By identifier, we often refer to a dummy identifier, e.g., incremental IDs, created to distinguish cases. As already mentioned, the minimal required information for process mining is the sequence of activities having been performed for each case, known as a *trace*. Therefore, an event log can be defined as a multiset of traces, i.e., a multiset of sequences of activities. Considering this minimal required information, the first challenge is that a trace can be considered as a *quasi-identifier* and, at the same time, as a *sensitive attribute*. In other words, a complete sequence of activities belonging to a case, is sensitive person-specific information, at the same time, part of a trace, i.e., only some of the activities, can be utilized as a *quasi-identifier* to identify the trace owner.

The *quasi-identifier* role of traces in process mining causes significant challenges for group-based anonymization techniques because of two specific properties of event data; *high variability* and *Pareto distribution*. In an event log the variability of traces is high because: (1) There could be tens of different activities happening in any order, (2) One activity or a bunch of activities could happen repetitively, and (3) Some traces could contain a few activities compared to all possible activities. In an event log, trace variants are often distributed similarly to the Pareto distribution, i.e., few trace variants are frequent and many trace variants are unique. Enforcing k-anonymity on little-overlapping traces in a high-dimensional space is a significant challenge, and the majority part of the data have to be suppressed in order to achieve the desired anonymization.

3 Preliminaries (Formal Models)

In this section, we provide formal models for event logs and possible attacks. These formal models will be used in the remainder for describing the approach.

3.1 Event Log Model

For a given set A, A^* is the set of all finite sequences over A, and $\mathcal{B}(A)$ is the set of all multisets over the set A. A finite sequence over A of length n is a mapping $\sigma \in \{1, ..., n\} \rightarrow A$, represented as $\sigma = \langle a_1, a_2, ..., a_n \rangle$ where $\sigma_i = a_i = \sigma(i)$ for any $1 \leq i \leq n$. $|\sigma|$ denotes the length of the sequence. For $\sigma_1, \sigma_2 \in A^*$, $\sigma_1 \sqsubseteq \sigma_2$ if σ_1 is a subsequence of σ_2, e.g., $\langle a, b, c, x \rangle \sqsubseteq \langle z, x, a, b, b, c, a, b, c, x \rangle$. For $\sigma \in A^*$, $\{a \in \sigma\}$ is the set of elements in σ, and $[a \in \sigma]$ is the multiset of elements in σ, e.g., $[a \in \langle x, y, z, x, y \rangle] = [x^2, y^2, z]$. For $x = (a_1, a_2, ..., a_n) \in A_1 \times A_2 \times ... \times A_n$, $\pi_k(x) = a_k$, i.e., the k-th element of the tuple. For $\sigma \in (A_1 \times A_2 \times ... \times A_n)^*$, $\pi_k(\sigma) = \langle \pi_k(x) \mid x \in \sigma \rangle$, i.e., the sequence projected on the k-th element. For example, $\pi_1(\langle (a_1, t_1), (a_2, t_2), ..., (a_n, t_n) \rangle) = \langle a_1, a_2, ..., a_n \rangle$. These notations can be combined, e.g., $[a \in \pi_k(\sigma)]$ is the multiset of elements for the sequence projected on the k-th element.

Table 2: A simple event log derived from Table 1 (each row represents a simple process instance).

Case Id	Simple Trace	Disease
1	⟨(RE,01-08:30),(V,01-08:45),(RL,01-08:58)⟩	Flu
2	⟨(RE,01-08:46),(HO,01-09:46),(BT,01-10:00),(BT,02-08:00),(V,02-10:00),(RL,02-14:00)⟩	HIV
3	⟨(RE,01-08:50),(HO,01-10:00),(BT,01-10:15),(V,02-10:15),(RL,02-14:15)⟩	Infection
4	⟨(RE,01-08:55),(V,01-09:10),(IN,01-09:30),(RL,01-10:30)⟩	Poisoning
5	⟨(RE,01-09:00),(V,01-09:20),(HO,01-09:55),(BT,01-10:10),(RL,02-16:00)⟩	Cancer
6	⟨(RE,01-09:05),(V,01-10:20),(RL,01-14:20)⟩	Hypotension

Definition 1 (Event, Event Log). *An event is a tuple* $e = (a, r, c, t, d_1, ..., d_m)$*, where* $a \in \mathcal{A}$ *is the activity associated with the event,* $r \in \mathcal{R}$ *is the resource, who is performing the activity,* $c \in \mathcal{C}$ *is the case id,* $t \in \mathcal{T}$ *is the event timestamp, and* d_1*,...,*d_m *is a list of additional attributes values, where for any* $1 \leq i \leq m, d_i \in \mathcal{D}_i$ *(domain of attributes). We call* $\xi = \mathcal{A} \times \mathcal{R} \times \mathcal{C} \times \mathcal{T} \times \mathcal{D}_1 \times ... \times \mathcal{D}_m$ *the event universe. An* ***event log*** *is* $EL \subseteq \xi$ *where each event can appear only once, i.e., events are uniquely identifiable by their attributes.*

Definition 2 (Simple Process Instance, Simple Trace, Simple Event). *We define* $\mathcal{P} = \mathcal{C} \times (\mathcal{A} \times \mathcal{T})^* \times \mathcal{S}$ *as the universe of all simple process instances, where* $\mathcal{S}$ *is the domain of the sensitive attribute. Each simple process instance* $p = (c, \sigma, s) \in \mathcal{P}$ *represents a* ***simple trace*** $\sigma = \langle (a_1, t_1), (a_2, t_1), ..., (a_n, t_n) \rangle$*, which is a sequence of* ***simple events****, containing activities and timestamps, belonging to the case* c *with* s *as the sensitive attribute value.*

Definition 3 (Simple Event Log). *Let* $\mathcal{P} = \mathcal{C} \times (\mathcal{A} \times \mathcal{T})^* \times \mathcal{S}$ *be the universe of simple process instances. A simple event log is* $EL \subseteq \mathcal{P}$ *such that if* $(c_1, \sigma_1, s_1) \in EL$*,* $(c_2, \sigma_2, s_2) \in EL$*, and* $c_1 = c_2$*, then* $\sigma_1 = \sigma_2$ *and* $s_1 = s_2$*.*

Table 2 shows a simple event log derived from Table 1, where timestamps are represented as "day-hour:minute". In this event log, "Disease" is the attribute which is considered as the sensitive attribute. In the remainder, by event log, trace, and event, we refer to Definition 2 and Definition 3.

3.2 Attack Model

Considering the typical scenario of data collection and data publishing [7], we assume the *trusted model*, where the *data holder* (here, a hospital) is trustworthy. However, the *data recipient* (here, a process miner) is not trustworthy, i.e., a process miner may attempt to identify sensitive information from record owners. In this subsection, we explain the real attack scenarios based on the *quasi-identifier* role of traces. Note that the examples used in the following definitions are based on Table 2.

Definition 4 (Background Knowledge 1 - bk_{set}^{EL}**).** *In the first scenario, we assume that the adversary knows a subset of activities having been done for the case, and this information can lead to the case (attribute) linkage attack. Let* EL *be an event log, we formalize this background knowledge by a function* $bk_{set}^{EL} : 2^{\mathcal{A}} \rightarrow 2^{EL}$*. For* $A \subseteq \mathcal{A}$*,* $bk_{set}^{EL}(A) = \{(c, \sigma, s) \in EL \mid A \subseteq \{a \in \pi_1(\sigma)\}\}$*.*

For example, if the adversary knows that $\{V, IN\}$ is the subset of activities having been done for a case, the only matching case is case 4. Therefore, the whole sequence of activities and the sensitive attribute are disclosed.

Definition 5 (Background Knowledge 2 - bk^{EL}_{mult}). *In this scenario, we assume that the adversary knows not only a subset of activities having been done for the case, but also the frequency of each activity. Let EL be an event log, we formalize this background knowledge by a function $bk^{EL}_{mult} : \mathcal{B}(\mathcal{A}) \rightarrow 2^{EL}$. For $B \in \mathcal{B}(\mathcal{A})$, $bk^{EL}_{mult}(B) = \{(c, \sigma, s) \in EL \mid B \subseteq [a \in \pi_1(\sigma)]\}$.*

For example, if the adversary knows that $[HO^1, BT^2]$ is the multiset of activities having been performed for a case, the only matching case is case 2. Consequently, the whole sequence of activities and the diseases are disclosed.

Definition 6 (Background Knowledge 3 - bk^{EL}_{seq}). *In this scenario, we assume that the adversary knows a subsequence of activities having been done for the case, and this information can lead to the case (attribute) linkage attack. Let EL be an event log, we formalize this background knowledge by a function $bk^{EL}_{seq} : \mathcal{A}^* \rightarrow 2^{EL}$. For $\sigma \in \mathcal{A}^*$, $bk^{EL}_{seq}(\sigma) = \{(c, \sigma', s) \in EL \mid \sigma \sqsubseteq \pi_1(\sigma')\}$.*

For example, if the adversary knows that $\langle RE, V, HO \rangle$ is the subsequence of activities having been performed for a case, the only matching case is case 5. Note that case 3 and case 5 have the same set of activities and by assuming bk^{EL}_{set}, the adversary is not able to single out a case, and since the matching cases have different values as the sensitive attribute, the adversary cannot certainly deduce the actual value of the sensitive attribute.

As can be seen, case 1 and case 6 are not distinguishable according to the defined types of background knowledge, i.e., the *case linkage* attack is not possible. However, by considering the timestamps, another attack scenario can be considered. In order to avoid revealing the exact timestamps of events, we assume that the timestamps are relative rather than absolute.

Definition 7 (Relative Timestamps). *Let $\sigma = \langle (a_1, t_1), (a_2, t_2), ..., (a_n, t_n) \rangle$ be a trace and t_0 be an initial timestamp, $rel(\sigma) = \langle (a_1, t'_1), (a_2, t'_2), ..., (a_n, t'_n) \rangle$ is the trace with relative timestamps such that $t'_1 = t_0$ and for each $1 < i \leq n$, $t'_i = t_i - t_1 + t_0$.*

Definition 8 (Background Knowledge 4 - bk^{EL}_{rel}). *In this scenario, we assume that the adversary knows not only a subsequence of activities, but also the time difference between the activities. Let EL be an event log, we formalize this background knowledge by a function $bk^{EL}_{rel} : (\mathcal{A} \times \mathcal{T})^* \rightarrow 2^{EL}$. For $\sigma \in (\mathcal{A} \times \mathcal{T})^*$, $bk^{EL}_{rel}(\sigma) = \{(c, \sigma', s) \in EL \mid rel(\sigma) \sqsubseteq rel(\sigma')\}$.*

For example, case 1 and case 6 have the same sequence of activities. However, if the adversary knows that for a victim case, it took almost four hours to get released after visiting by a doctor, the corresponding possible cases narrow down to only one case, which is case 6. The defined types of background knowledge can be categorized from more general and easily achievable to more specific

and difficult to achieve, i.e., bk^{EL}_{set} is the most general and easier to gain by an adversary, and bk^{EL}_{rel} is the most specific one. Corresponds to the four defined types of background knowledge and considering a trace in an event log, we define four types of quasi-identifiers w.r.t. the trace and four matching sets for the trace.

Definition 9 (Trace-based Quasi-identifiers - QID^{σ}_{set}, QID^{σ}_{mult}, QID^{σ}_{seq}, QID^{σ}_{rel}). *Let EL be an event log and σ be a trace such that $(c, \sigma, s) \in EL$. Given the four defined types of background knowledge, $QID^{\sigma}_{set} = \{a \in \pi_1(\sigma)\}$, $QID^{\sigma}_{mult} = [a \in \pi_1(\sigma)]$, $QID^{\sigma}_{seq} = \pi_1(\sigma)$, and $QID^{\sigma}_{rel} = rel(\sigma)$.*

Definition 10 (Matching Sets - EL^{σ}_{set}, EL^{σ}_{mult}, EL^{σ}_{seq}, EL^{σ}_{rel}). *Let EL be an event log and σ be a trace such that $(c, \sigma, s) \in EL$. Given the four defined types of background knowledge, $EL^{\sigma}_{set} = \{bk^{EL}_{set}(A) \mid A \subseteq QID^{\sigma}_{set}\}$, $EL^{\sigma}_{mult} = \{bk^{EL}_{mult}(B) \mid B \subseteq QID^{\sigma}_{mult}\}$, $EL^{\sigma}_{seq} = \{bk^{EL}_{seq}(\sigma') \mid \sigma' \sqsubseteq QID^{\sigma}_{seq}\}$, and $EL^{\sigma}_{rel} = \{bk^{EL}_{rel}(\sigma') \mid rel(\sigma') \sqsubseteq QID^{\sigma}_{rel}\}$.*

4 TLKC-Privacy Model

Regular k-anonymity and its extended privacy models assume that an adversary could use all of the quasi-identifier attributes as background knowledge to launch the attacks. However, in reality, it is almost impossible for an adversary to acquire all the information of a target victim, and it requires non-trivial effort to gather each piece of background knowledge. The LKC-privacy model exploits this limitation and assume that the adversary's background knowledge is bounded by at most L values of the quasi-identifier.

Based on the bounded background knowledge, proposed by the LKC-privacy model [16], we introduce $TLKC$-privacy model for process mining. In the LKC-privacy model, L refers to the power of background knowledge, i.e., the length of a sequence, K refers to the k in the k-anonymity definition, and C refers to the bound of confidence regarding the sensitive attribute values in an equivalence class. In the $TLKC$-privacy model $T \in \{seconds, minutes, hours, days\}$ is added which refers to the accuracy of timestamps. For example, when $T = hours$, the accuracy of timestamps is limited at *hours* level. We denote $EL(T)$ as the event log with the accuracy of timestamps at the level T. The general idea of $TLKC$-privacy is to ensure that the background knowledge with maximum length L in $EL(T)$ is shared by at least K cases, and the confidence of inferring any sensitive value in S given the quasi-identifier is not greater than C.

Definition 11 ($TLKC$-Privacy). *Let EL be an event log, L be the maximum length of background knowledge, $T \in \{seconds, minutes, hours, days\}$ be the accuracy of timestamps, and type $\in \{set, mult, seq, rel\}$. $EL(T)$ satisfies $TLKC$-privacy if and only if for any trace $\sigma \sqsubseteq \sigma'$, $(c, \sigma', s) \in EL$, and $0 < |\sigma| \leq L$:*

- $|EL(T)^{\sigma}_{type}| \geq K$, *where* $K \in \mathbb{N}_{>0}$, *and*
- $Pr(s|QID^{\sigma}_{type}) = \frac{|\{s \in \pi_3(p) | p \in EL(T)^{\sigma}_{type}\}|}{|EL(T)^{\sigma}_{type}|} \leq C$ *for any* $s \in S$, *where* $0 < C \leq 1$ *is a real number as the confidence threshold.*

$TLKC$-privacy inherits several properties from LKC-privacy that makes it suitable for anonymizing high-dimensional sparse event data. First, it provides a major relaxation from traditional k-anonymity based on a reasonable assumption that the adversary has restricted knowledge. Second, it generalizes several privacy models including; k-anonymity, confidence bounding, (α, k)-anonymity, and l-diversity. Third, it provides the flexibility to adjust the trade-off between data privacy and data utility, and between an adversary's power and data utility.

4.1 Utility Measure

The measure of data utility depends on the task which is supposed to be performed. However, in process mining, and specifically for process discovery, we want to preserve the maximal frequent traces which are defined as follows.

Definition 12 (Maximal Frequent Trace - MFT). *Let EL be an event log. For a given minimum support threshold Θ, a non-empty trace $\sigma \sqsubseteq \sigma'$ such that $(c, \sigma', s) \in EL$ is maximal frequent in the EL if σ is frequent, i.e., the frequency of σ is greater than or equal to Θ, and no supertrace of σ is frequent in the EL.*

The goal of data utility is to preserve as many MFT as possible. We denote the set of MFT in an event log EL by MFT_{EL}, which is much smaller than the set of frequent traces in the event log EL. Note that any subtrace of an MFT is also a frequent trace, and once all the MFT have been discovered, the support counts of any frequent subtrace can be computed by scanning the data once.

4.2 The Algorithm

The first step is to find all traces that violate the given $TLKC$-privacy requirement. We define a violating trace as follows.

Definition 13 (Violating Trace). *Let EL be an event log, $\sigma \sqsubseteq \sigma'$ such that $(c, \sigma', s) \in EL$, L be the maximum length of the background knowledge, $T \in \{seconds, minutes, hours, days\}$ be the accuracy of timestamps, type $\in \{set, mult, seq, rel\}$, and $0<|\sigma|\leq L$. σ is violating with respect to $TLKC$-privacy requirements if $|EL(T)^{\sigma}_{type}|< K$ or $Pr(s|QID^{\sigma}_{type}) > C$ for any $s \in S$.*

An event log satisfies $TLKC$-privacy, if all violating traces w.r.t. the given privacy requirement are removed. A naïve approach is to determine all possible violating traces and remove them. However, this approach is inefficient because of the numerous number of violating traces, even for a weak privacy requirement.

Table 3: A simple event log with relative timestamps for monotonic property.

Case Id	Trace	Disease
1	σ_1=⟨(RE,01-00:00:00),(V,01-01:02:00)⟩	Flu
2	σ_2=⟨(RE,01-00:00:00),(V,01-01:02:00),(RL,01-01:10:00)⟩	Flu
3	σ_3=⟨(RE,01-00:00:00),(V,01-01:02:00),(RL,01-01:10:00)⟩	HIV

Algorithm 1: TLKC-Privacy Algorithm

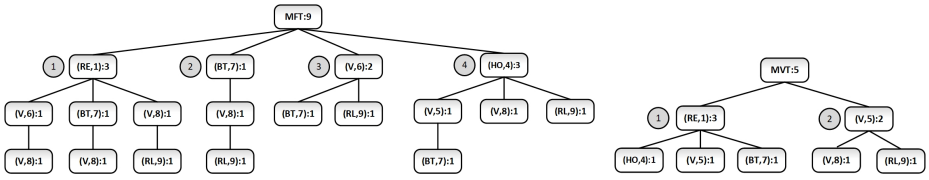
Input: Original event log EL
Input: T, L, K, C, and Θ
Input: Sensitive values S
Output: Anonymized event log EL' which satisfies $TLKC$-privacy
1 generate MFT_{EL} and MVT_{EL};
2 generate MFT_{tree} and MVT_{tree} as the prefix trees for MFT_{EL} and MVT_{EL};
3 **while** *there is node (event) in* MVT_{tree} **do**
4 select an event (node) e_w that has the highest score to suppress;
5 delete all the MVT and MFT containing the event e_w from MVT_{tree} and MFT_{tree};
6 update $Socre(e)$ for all the remaining events (nodes) in MVT_{tree};
7 add e_w to the suppression set Sup_{EL};
8 **end**
9 **foreach** $e \in Sup_{EL}$ **do**
10 suppress all instances of e from EL;
11 **end**
12 return suppressed EL as EL';

In [16], the authors demonstrate that LKC-privacy is not monotonic w.r.t. L, which holds for $TLKC$-privacy as well. The anonymity threshold K is monotonic w.r.t. L, i.e., if $L' \leq L$ and $C = 100\%$, an event log EL satisfying $TLKC$-privacy must satisfy $TL'KC$-privacy. However, confidence threshold C is not monotonic w.r.t. L, i.e., if σ is non-violating trace, its subtrace may or may not be a non-violating trace. For example, in Table 3, for $L = 3$ and $C = 75\%$, trace σ_2 satisfies $Pr(Flu|\sigma_2) \leq 75\%$. However, its subtrace σ_1 with $L' = 2$ does not satisfy $Pr(Flu|\sigma_1) \leq 75\%$. Therefore, in order to satisfy the second condition in Definition 11, it is insufficient to ensure that every trace σ in EL satisfies $Pr(s|QID^{\sigma}_{type}) \leq C$ for $|\sigma| = L$, and the condition should hold for $0 < |\sigma| \leq L$. To this end, the *minimal violating traces* are defined.

Definition 14 (Minimal Violating Trace - MVT). *Let EL be an event log, a violating trace $\sigma \sqsubseteq \sigma'$ such that $(c, \sigma', s) \in EL$ is a minimal violating trace in the EL if every proper subtrace of σ is not a violating trace in the EL.*

Every violating trace in an event log is either an MVT or it contains an MVT. Therefore, if an event log EL contains no MVT, then EL contains no violating trace. We denote the set of MVT in an event log EL by MVT_{EL}, which is much smaller than the set of violating traces in the event log EL. A greedy function $Score : \xi \rightarrow \mathbb{R}_{>0}$ is defined to choose an event e to suppress such that it maximizes the number of removed minimal violating traces (privacy gain), but minimizes the number of removed maximal frequent traces (utility loss). For $e \in \xi$, $Score(e) = {}^{PG(e)}/_{UL(e)+1}$. $PG(e)$ is the number of MVT containing the event e, and $UL(e)$ is the number of MFT containing the event e. In order to avoid diving by zero (when e does not belong to any MFT), 1 is added to the denominator. The event e with the highest score is called the *winner* event, denoted by e_w. Algorithm 1 summarizes all the steps of $TLKC$-privacy.

Suppose that Table 4 shows a simple event log EL where timestamps are represented by integer values as hours. The first line in Algorithm 1 generates the set of maximal frequent traces (MFT_{EL}) and the set of minimal violating traces (MVT_{EL}) from the event log EL with $T = hours$, $L = 2$, $K = 2$, $C = 50\%$, $\Theta =$



(a) MFT_{tree} (b) MVT_{tree}

Fig. 1: The MFT_{tree} and MVT_{tree} generated for the event log Table 4 with $T = days$, $L = 2$, $K = 2$, $C = 50\%$, $\Theta = 25\%$, $S = Disease$, and bk_{rel}^{EL}.

25%, *Disease* as the sensitive attribute S, and bk_{rel}^{EL} as the background knowledge. $MFT_{EL} = \{\langle(RE,1),(V,6),(V,8)\rangle, \langle(RE,1),(BT,7),(V,8)\rangle, \langle(RE,1),(V,8),(RL,9)\rangle, \langle(HO,4),V,5),(BT,7)\rangle, \langle(BT,7),(V,8),(RL,9)\rangle, \langle(V,6),(BT,7)\rangle, \langle(V,6),(RL,9)\rangle, \langle(HO,4),(V,8)\rangle, \langle(HO,4),(RL,9)\rangle\}$, and $MVT_{EL} = \{\langle(RE,1),(HO,4)\rangle, \langle(RE,1),(V,5)\rangle, \langle(RE,1),(BT,7)\rangle, \langle(V,5),(V,8)\rangle, \langle(V,5),(RL,9)\rangle\}$.

Figure 1 shows the MFT_{tree} and MVT_{tree} generated by line 2 in Algorithm 1, where each root-to-leaf path represents one trace, and each node represents an event in a trace with the frequency of occurrence. Table 5 shows the initial $Score(e)$ of every event (node) in the MVT_{tree}. Line 4 determines the winner event e_w which is $(V,5)$. Line 5 deletes all the MVT and MFT containing the winner event e_w, i.e., subtree 2 and the path $\langle(RE,1),(V,5)\rangle$ of subtree 1 in the MVT_{tree} as well as the path $\langle(HO,4),(V,5),(BT,7)\rangle$ of subtree 4 in the MFT_{tree} are removed and frequencies get updated. Line 6 updates the scores based on the new frequencies of events. Table 6 shows the remaining events in MVT_{tree} with the updated scores. Line 7 adds the winner event to a suppression set Sup_{EL}. Lines 4-7 is repeated until there is no node in MVT_{tree}. According to Table 6 the next winner event is $(RE,1)$, and after deleting all the MVT and MFT containing this event, MVT_{tree} is empty. Therefore, at the end of the ***while*** loop, the suppression set $Sup_{EL} = \{(V,5),(RE,1)\}$. The ***foreach*** loop suppresses all the instances of the events (*global suppression*) in the Sup_{EL} from the EL, and the last line returns the suppressed EL as the anonymized event log EL' which is shown by Table 7. Table 8 shows the result by applying the traditional k-anonymity with $k = 2$ on the event log Table 4. One can see that even for a weak privacy requirement, much information needs to be suppressed compared to the results provided by $TLKC$-privacy.

Table 4: A simple event log where timestamps are represented by integer values.

Case Id	Trace	Disease
1	$\langle(RE,1),(HO,4),(V,5),(BT,7),(V,8)\rangle$	Cancer
2	$\langle(BT,7),(V,8),(RL,9)\rangle$	Infection
3	$\langle(HO,4),(V,5),(BT,7),(RL,9)\rangle$	Poisoning
4	$\langle(RE,1),(V,6),(V,8),(RL,9)\rangle$	Infection
5	$\langle(HO,4),(V,8),(RL,9)\rangle$	Poisoning
6	$\langle(V,6),(BT,7),(RL,9)\rangle$	Flu
7	$\langle(RE,1),(BT,7),(V,8),(RL,9)\rangle$	Flu
8	$\langle(RE,1),(V,6),(BT,7),(V,8)\rangle$	Cancer

Table 5: The initial scores for the events in Fig. 1b.

	$(RE,1)$	$(HO,4)$	$(V,5)$	$(BT,7)$	$(V,8)$	$(RL,9)$
$PG(e)$	3	1	3	1	1	1
$UL(e)+1$	4	4	2	5	6	5
$Score(e)$	0.75	0.25	1.50	0.20	0.16	0.20

Table 6: The first updated scores.

	$(RE,1)$	$(HO,4)$	$(BT,7)$
$PG(e)$	2	1	1
$UL(e)+1$	4	3	4
$Score(e)$	0.5	0.33	0.25

Table 7: The anonymized event log for Table 4 with $T = days$, $L = 2$, $K = 2$, $C = 50\%$, $\Theta = 25\%$, $S = Disease$, and bk^{EL}_{rel}.

Case Id	Trace	Disease
1	$\langle(HO,4),(BT,7),(V,8)\rangle$	Cancer
2	$\langle(BT,7),(V,8),(RL,9)\rangle$	Infection
3	$\langle(HO,4),(BT,7),(RL,9)\rangle$	Poisoning
4	$\langle(V,6),(V,8),(RL,9)\rangle$	Infection
5	$\langle(HO,4),(V,8),(RL,9)\rangle$	Poisoning
6	$\langle(V,6),(BT,7),(RL,9)\rangle$	Flu
7	$\langle(BT,7),(V,8),(RL,9)\rangle$	Flu
8	$\langle(V,6),(BT,7),(V,8)\rangle$	Cancer

Table 8: The traditional 2-anonymity event log for Table 4.

Case Id	Trace	Disease
1	$\langle(BT,7),(V,8)\rangle$	Cancer
2	$\langle(BT,7),(V,8),(RL,9)\rangle$	Infection
3	$\langle(BT,7),(RL,9)\rangle$	Poisoning
4	$\langle(V,8),(RL,9)\rangle$	Infection
5	$\langle(V,8),(RL,9)\rangle$	Poisoning
6	$\langle(BT,7),(RL,9)\rangle$	Flu
7	$\langle(BT,7),(V,8),(RL,9)\rangle$	Flu
8	$\langle(BT,7),(V,8)\rangle$	Cancer

5 Evaluation

We evaluate our proposed privacy protection model by applying it on a real-life event log and exploring the effect on the accuracy of the process discovery and performance analysis compared to the ground truth. As the ground truth we use the original process model discovered from the original event log. We employed *Sepsis Case* [13] to conduct our experiments due to some challenging features that it has for process discovery. This event data is a hospital event log containing 16 unique activities, 1050 traces, and 846 variants, which are unique traces, i.e., 80% of traces are unique. The maximum number of traces per variant is 35, the maximum trace length is 185, on average the traces contain 14.5 events, i.e., the average length of traces is 14.5. Note that we provide privacy guarantees w.r.t. the power of background knowledge (L), i.e., all the subtraces having the maximal length L should fulfill the $TLKC$-privacy requirements (Definition 11). Since 80% of traces are unique, this event log is significantly challenging for privacy-preserving process discovery algorithms [6,14].

Overall 1536 experiments have been done for four different types of background knowledge, 384 per each background knowledge, using $T \in \{hours, minutes\}$, $L \in \{2,4,8,16\}$, $K \in \{10,20,40,80\}$, $C \in \{0.2,0.3,0.4,0.5\}$, and $\Theta \in \{0.7,0.8,0.9\}$. We consider "disease" and "age" as the sensitive case attributes in the *Sepcis* event log. The confidence value C should not be greater than 0.5, i.e., there are at least two different sensitive values for a victim case. We convert the numerical attributes to categorical using *Boxplots* such that all the values greater than the upper quartile are categorized as *high*, the values less than the lower quartile are categorized as *low*, and the values in between are categorized as *middle*. Regarding the number of unique activities in this event log, it is not realistic to consider the power of background knowledge greater than 16. This is the maximal *set* background knowledge, i.e., an adversary knows all the activities that can be done. Moreover, the length of 75% of the traces in this event log is maximal 16. We consider two settings as representatives to interpret the results in detail; *weak setting* and *strong setting*. For the *weak setting*, we use $T = hours$, $L = 2$, $K = 10$, $C = 0.5$, and $\Theta \in \{0.7,0.8,0.9\}$. For the *strong*

setting, we use $T = minutes$, $L = 8$, $K = 80$, $C = 0.2$, and $\Theta \in \{0.7, 0.8, 0.9\}$. The implementation as a Python program is available on Github.[1]

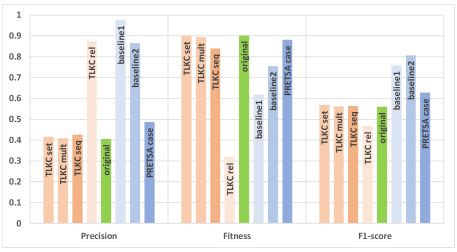
5.1 Process Discovery

To evaluate the effect of applying our method on the accuracy of discovered models, we consider three main questions. **Q1:** How accurately do the discovered process models capture the behavior of the original event log? **Q2:** How similar are the discovered process models to the original process model in terms of some quality measures? **Q3:** How is the content of the original event log preserved by the privacy model? To answer Q1, we first discover a process model M' from an anonymized event log EL'. Then, for M', we calculate *fitness*, *precision*, and *f1-score* [1], as some model quality measures, w.r.t. the original event log EL. *Fitness* quantifies the extent to which the discovered model can reproduce the traces recorded in the event log. *Precision* quantifies the fraction of the traces allowed by the model which is not seen in the event log, and *f1-score* combines the fitness and precision $f1\text{-}score = {}^{2\times precision\times fitness}/_{precision+fitness}$. To answer Q2, we discover two process models; the original process model M from the original event log EL and a process model M' from an anonymized event log EL'. Then, we calculate *fitness*, *precision*, and *f1-score* of M and M' w.r.t. EL. At the end, we compare the results to analyze the similarity of the quality measures. We use the *inductive miner infrequent* [10] with the default parameters as the process discovery algorithm. To answer Q3, we compare the number of *variants*, which are the unique traces in the event log, after applying our method with the actual number of variants. Note that applying privacy-preserving algorithms may result in high *precision* and probably high *f1-score*. However, high values for some quality measures do not necessarily mean that the privacy-preserving algorithm preserves the data utility, since the aim is to provide as similar results as possible not to improve the quality of discovered models.

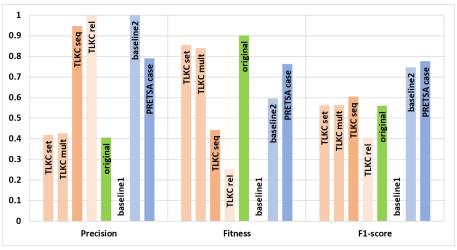
As we discuss in Section 6, $PRETSA$ is the only similar algorithm which applies k-anonymity and t-closeness on event data for privacy-aware process discovery. However, $PRETSA$ focuses on the *resource perspective* of privacy while we focus on the *case perspective* of privacy. To compare our method with similar methods, we have developed a variant of $PRETSA$ algorithm $PRETSA_{case}$ where only the k-anonymity part is considered, and the focus is on the privacy of *cases* rather than *resources*. The background knowledge assumed by $PRETSA$ is a prefix of the sequence of executed activities. We have also developed two naïve baseline algorithms. *baseline*1 is a naïve k-anonymity algorithm, where we remove all the traces that occur less than k times in the event log. *baseline*2 considers k-anonymity and maps each violating trace to the most similar non-violating subtrace by removing events. For the baseline algorithms and $PRETSA_{case}$ only K is considered from the *settings*.

Figure 2a shows how the mentioned quality measures are affected by applying our method with the weak setting (average of three experiments regarding

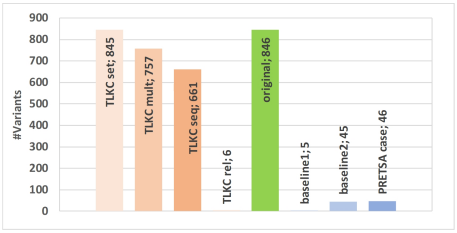
[1] https://github.com/Widderiru/TLKC-privacy/tree/master/home_version



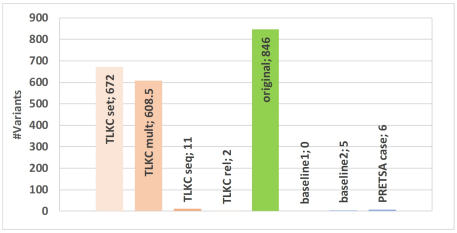
(a) The measures with the weak setting.



(b) The measures with the strong setting.



(c) The #variants with the weak setting.

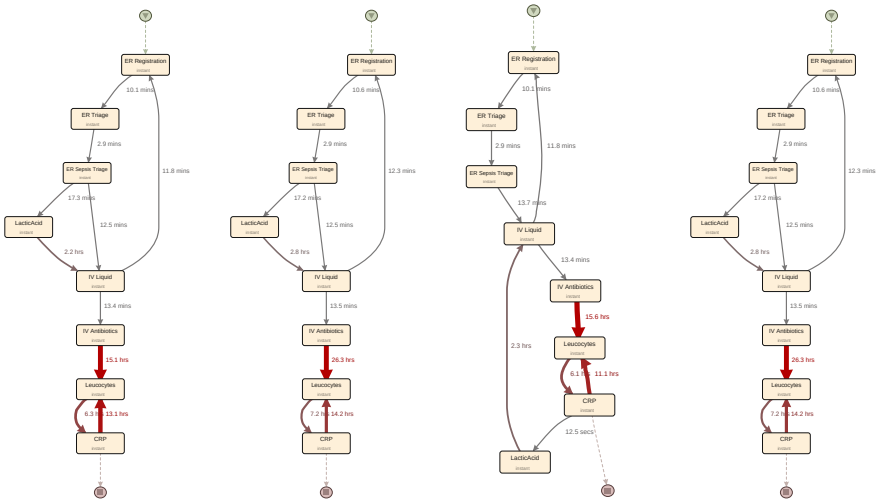


(d) The #variants with the strong setting.

Fig. 2: The number of variants and quality measures comparison between the four variants of $TLKC$-privacy, the original results, $PRETSA_{case}$, and the baseline algorithms.

Θ), while we consider four variants of our privacy model based on the introduced types of background knowledge including; $TLKC_{set}$, $TLKC_{mult}$, $TLKC_{seq}$, and $TLKC_{rel}$. We compare the measures with the results from the original process model, two baseline algorithms, and $PRETSA_{case}$. If we only consider Q1, the baseline algorithms should be marked as the best ones, since they result in better *f1-score* values. However, as can be seen in Fig. 2c, the baseline algorithms remove many variants from the original event log. Consequently, the corresponding anonymized event logs contain significantly less behavior compared to the original event log, and the resulting models have high *precision*, which in turn results in high *f1-score*. Figure 2a and Fig. 2c show that the results from our privacy model are considerably similar to the original results, except for $TLKC_{rel}$. $TLKC_{rel}$ removes many variants compared to the other variants which is not surprising regarding the assumed background knowledge which is considerably strong, but, difficult to achieve in reality.

Figure 2b and Fig. 2d show the same experiments based on the mentioned quality measures with the strong setting (average of three experiments regarding Θ). Figure 2d shows that even for the strong setting, our privacy model preserves a considerably high amount of content of the original event log considering more general types of background knowledge (bk^{EL}_{set} and bk^{EL}_{mult}). However, $TLKC_{seq}$ preserves fewer variants with the strong setting which results in high precision. Note that the baseline algorithms and $PRETSA_{case}$ do not protect event data against the *attribute linkage* attack and provide weaker privacy guarantees.



(a) DFG'-$TLKC_{set}$ (b) DFG-$TLKC_{set}$ (c) DFG'-$TLKC_{mult}$ (d) DFG-$TLKC_{mult}$

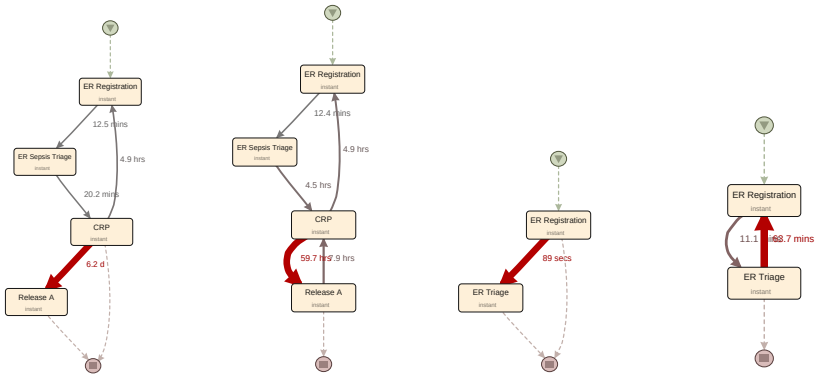
Fig. 3: The performance-annotated DFGs from the projected event log (DFG) and an anonymized event log (DFG') for $TLKC_{set}$ and $TLKC_{mult}$ with the strong setting where $\Theta = 0.7$.

5.2 Performance

The effect on performance analyses is evaluated by analyzing the bottlenecks w.r.t. the mean duration of cases between activities. Since the privacy-preserving algorithm may have removal activities, we cannot compare the bottlenecks in the original process model with the bottlenecks in a process model discovered from an anonymized event log. Therefore, we first project the original event log on the activities existing in the anonymized event log. Then, we discover a performance-annotated directly follows graph DFG from the projected event log and compare it with the performance-annotated directly follows graph DFG' from the anonymized event log. A DFG is a graph where the nodes represent activities and the arcs represent causalities. Activities "a" and "b" are connected by an arrow when "a" is frequently followed by "b" [11].

Here, we show the results for the strong setting with $\Theta = 0.7$ in Fig. 3 and Fig. 4.[2] As can be seen, the bottlenecks in DFG and DFG' are the same for all the variants except for $TLKC_{rel}$, where the assumed background knowledge is significantly strong and only a few variants remain after applying the method. Note that the mean duration of the cases are different in DFG and DFG' because of the use of relative timestamps in the anonymized event logs. This experiment shows the similarity of the results in terms of real process models.

[2]These results have been provided by Disco (https://fluxicon.com/disco/) with the sliders set to the maximal number of activities and the minimal paths.



(a) DFG'-$TLKC_{seq}$ (b) DFG-$TLKC_{seq}$ (c) DFG'-$TLKC_{rel}$ (d) DFG-$TLKC_{rel}$

Fig. 4: The performance-annotated DFGs from the projected event log (DFG) and an anonymized event log (DFG') for $TLKC_{seq}$ and $TLKC_{rel}$ with the strong setting where $\Theta = 0.7$.

6 Related Work

During the last decades, privacy issues have received increasing attention. The privacy challenges in process mining are more similar to the privacy-preserving sequential pattern mining [9,4] and anonymizing trajectory data [17,16]. The privacy model, presented in this paper, extends the LKC-privacy model [16], both in the parameters and the type of background knowledge, to be fitted in the context of process mining. In process mining research, confidentiality and privacy received less attention. In [2], *Responsible Process Mining* (RPM) is introduced as the sub-discipline focusing on possible negative side-effects of applying process mining. RPM addresses concerns related to Fairness, Accuracy, Confidentiality, and Transparency (FACT). In [15], the authors propose a privacy-preserving system design for process mining, where a user-centered view is considered to track personal data. In [19,20], a framework is introduced, which provides a generic scheme for confidentiality in process mining. In [18], the aim is to provide a privacy-preserving method for discovering roles from event data.

Most related to our work are [6] and [14], where the authors propose privacy-preserving techniques for process discovery. Therefore, we pinpoint the differences with $TLKC$-privacy model. In [6], the authors apply k-anonymity and t-closeness [12] on event data to preserve the privacy of *resources* while we focus on the *case perspective*. Also, the assumed background knowledge is a prefix of sequence of activities which is restrictively specific. In [14], the authors employ the notion of differential privacy [5]. This research focuses on *case perspective* of privacy in process mining which is similar to our research from this point of view. However, the type of privacy guarantee is noise-based. As shown in [14], applying the noise-based privacy guarantees on event data is challenging

when the process models are unstructured and the majority of traces are unique. Moreover, noise-based techniques do not preserve the *truthfulness* of values at the case level [8], i.e., for some cases there is no corresponding individual in real life. Also, the performance aspect is not considered by this research.

7 Conclusions

In this paper, we introduced two perspectives for privacy in process mining (*case perspective* and *resource perspective*), and we discussed privacy challenges in process mining. We demonstrated that existing well-known privacy-preserving techniques cannot be directly applied to event data. We introduced the $TLKC$-privacy model for process mining which is an extension for the LKC-privacy model. Our proposed model preserves the privacy of the cases whose data is processed in process mining, particularly for process discovery and performance analyses. It counteracts both the *case linkage* and the *attribute linkage* attacks.

We implemented four variants of $TLKC$-privacy w.r.t. the four different types of background knowledge. All the variants have been evaluated based on a real-life event log which is highly challenging for process discovery techniques in terms of unique traces ratio. 384 experiments were performed per each type of background knowledge, and the results were given for a weak and a strong setting. Our experiments demonstrate that $TLKC$-privacy model preserves the data utility in terms of similarity of the results to the actual results. Specifically, for the more general types of background knowledge. Moreover, we showed that how the cost of privacy increases w.r.t. the strength of background knowledge.

For the *multiset* variant of $TLKC$ ($TLKC_{mult}$) many potential minimal violating traces with the length longer than one can be generated by the presented algorithm, which results in long computation times. In the future, smarter pruning algorithms could be explored to generate a smaller potential set of minimal violating traces. Moreover, some algorithms could be designed to automatically generate reasonable values for the parameters used by our algorithm.

Acknowledgment

We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.M.P.: Responsible data science: using event data in a "people friendly" manner. In: International Conference on Enterprise Information Systems. pp. 3–28. Springer (2016)
3. Aggarwal, C.C., Philip, S.Y.: Privacy-preserving data mining: models and algorithms. Springer Science & Business Media (2008)

4. Bonomi, L., Xiong, L.: A two-phase algorithm for mining sequential patterns with differential privacy. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM (2013)
5. Dwork, C.: Differential privacy: A survey of results. In: Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings. pp. 1–19 (2008)
6. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: PRETSA: event log sanitization for privacy-aware process discovery. In: International Conference on Process Mining, ICPM 2019, Aachen, Germany, June 24-26, 2019. pp. 1–8 (2019)
7. Fung, B.C., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Computing Surveys (Csur) **42**(4) (2010)
8. Fung, B.C., Wang, K., Fu, A.W.C., Philip, S.Y.: Introduction to privacy-preserving data publishing: Concepts and techniques. Chapman and Hall/CRC (2010)
9. Kapoor, V., Poncelet, P., Trousset, F., Teisseire, M.: Privacy preserving sequential pattern mining in distributed databases. In: Proceedings of the 15th ACM international conference on Information and knowledge management. ACM (2006)
10. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Business Process Management Workshops - BPM International Workshops. pp. 66–78 (2013)
11. Leemans, S.J., Fahland, D., van der Aalst, W.M.P.: Scalable process discovery and conformance checking. Software & Systems Modeling **17**(2), 599–631 (2018)
12. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20 (2007)
13. Mannhardt, F.: Sepsis cases-event log. Eindhoven University of Technology (2016)
14. Mannhardt, F., Koschmider, A., Baracaldo, N., Weidlich, M., Michael, J.: Privacy-preserving process mining - differential privacy for event logs. Business & Information Systems Engineering **61**(5), 595–614 (2019)
15. Michael, J., Koschmider, A., Mannhardt, F., Baracaldo, N., Rumpe, B.: User-centered and privacy-driven process mining system design for IoT. In: Information Systems Engineering in Responsible Information Systems. pp. 194–206 (2019)
16. Mohammed, N., Fung, B.C., Hung, P.C., Lee, C.k.: Anonymizing healthcare data: A case study on the blood transfusion service. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1285–1294. KDD '09, ACM, New York, NY, USA (2009)
17. Nergiz, M.E., Atzori, M., Saygin, Y.: Towards trajectory anonymization: a generalization-based approach. In: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS (2008)
18. Rafiei, M., van der Aalst, W.M.P.: Mining roles from event logs while preserving privacy. In: Business Process Management Workshops - BPM 2019 International Workshops, Vienna, Austria. pp. 676–689 (2019)
19. Rafiei, M., von Waldthausen, L., van der Aalst, W.M.P.: Ensuring confidentiality in process mining. In: Proceedings of the 8th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2018), Seville, Spain (2018)
20. Rafiei, M., von Waldthausen, L., van der Aalst, W.M.P.: Supporting confidentiality in process mining using abstraction and encryption. In: Data-Driven Process Discovery and Analysis. pp. 101–123. Springer International Publishing (2020)
21. Voss, W.G.: European union data privacy law reform: General data protection regulation, privacy shield, and the right to delisting. Business Lawyer **72**(1) (2016)