Towards Quantifying Privacy in Process Mining

Majid Rafiei[0000−0001−7161−6927] 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 employs event logs to provide insights into the actual processes. Event logs are recorded by information systems and contain valuable information helping organizations to improve their processes. However, these data also include highly sensitive private information which is a major concern when applying process mining. Therefore, privacy preservation in process mining is growing in importance, and new techniques are being introduced. The effectiveness of the proposed privacy preservation techniques needs to be evaluated. It is important to measure both sensitive data protection and data utility preservation. In this paper, we propose an approach to quantify the effectiveness of privacy preservation techniques. We introduce two measures for quantifying disclosure risks to evaluate the sensitive data protection aspect. Moreover, a measure is proposed to quantify data utility preservation for the main process mining activities. The proposed measures have been tested using various real-life event logs.

Keywords: Responsible process mining · Privacy preservation · Privacy quantification · Data utility · Event logs

1 Introduction

Process mining bridges the gap between traditional model-based process analysis (e.g., simulation), and data-centric analysis (e.g., data mining) [1]. The three basic types of process mining are *process discovery*, where the aim is to discover a process model capturing the behavior seen in an event log, *conformance checking*, where the aim is to find commonalities and discrepancies between a process model and an event log, and *process re-engineering* (*enhancement*), where the idea is to extend or improve a process model using event logs.

An event log is a collection of events. Each event has the following mandatory attributes: a *case identifier*, an *activity name*, a *timestamp*, and optional attributes such as *resources* or *costs*. In the human-centered processes, case identifiers refer to individuals. For example, in a patient treatment process, the case identifiers refer to the patients whose data are recorded. Moreover, other attributes may also refer to individuals, e.g., *resources* often refer to persons performing activities. When event logs explicitly or implicitly include personal data, *privacy concerns* arise which should be taken into account w.r.t. regulations such as the European General Data Protection Regulation (GDPR).

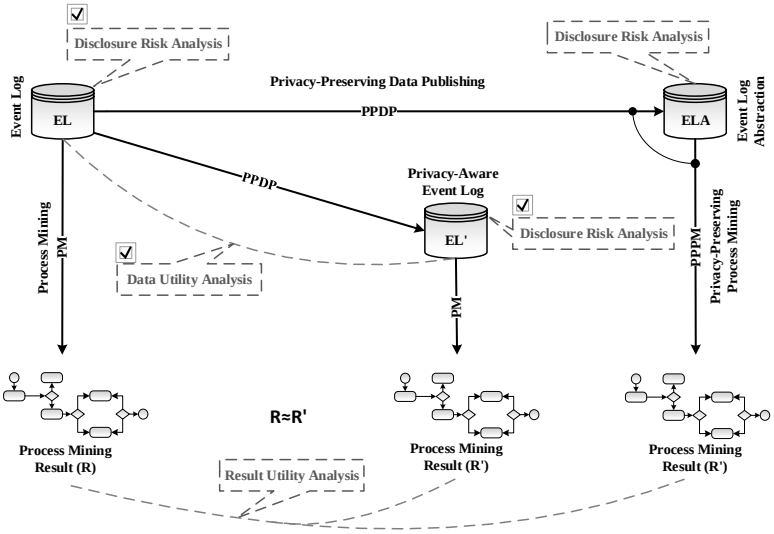


Fig. 1: Overview of privacy-related activities in process mining. Privacy preservation techniques are applied to event logs to provide desired privacy requirements. The aim is to protect sensitive personal data, yet, at the same time, preserve data utility, and generate as similar as possible results to the original ones. The parts indicated by dashed callouts show the analyses that need to be performed to evaluate the effectiveness of privacy preservation techniques.

The *privacy* and *confidentiality* issues in process mining are recently receiving more attention and various techniques have been proposed to protect sensitive data. Privacy preservation techniques often apply anonymization operations to modify the data in order to fulfill desired privacy requirements, yet, at the same time, they are supposed to preserve data utility. To evaluate the effectiveness of these techniques, their effects on *sensitive data protection* and *data utility preservation* need to be measured. In principle, privacy preservation techniques always deal with a trade-off between data utility and data protection, and they are supposed to balance these aims.

Figure 1 shows the general view of privacy in process mining including two main activities: *Privacy-Preserving Data Publishing* (PPDP) and *Privacy-Preserving Process Mining* (PPPM). PPDP aims to hide the identity and the sensitive data of record owners in event logs to protect their privacy. PPPM aims to extend traditional process mining algorithms to work with the non-standard event data so-called *Event Log Abstraction* (ELA) [16] that might result from PPDP techniques. *Abstractions* are intermediate results, e.g., a directly follows graph could be an intermediate result of a process discovery algorithm. Note that PPPM algorithms are tightly coupled with the corresponding PPDP techniques.

In this paper, our main focus is on the analyses indicated by the check-boxes in Fig. 1. Note that *disclosure risk analysis* is done for a single event log, while for *data/result utility analysis*, the original event log/result need to be

compared with the privacy-aware event log/result. We consider simple event logs containing basic information for performing two main process mining activities: *process discovery* and *conformance checking*. We introduce two measures for quantifying disclosure risks in a simple event log: *identity (case) disclosure* and *attribute (trace) disclosure*. Using these measures, we show that even simple event logs could disclose sensitive information. We also propose a measure for quantifying *data utility* which is based on the *earth mover's distance*. So far, the proposed privacy preservation techniques in process mining use the *result utility* approach to demonstrate the utility preservation aspect which is not as precise and general as the *data utility* approach, since it is highly dependent on the underlying algorithms. We advocate the proposed measures by assessing their functionality for quantifying the disclosure risks and data utility on real-life event logs before and after applying a privacy preservation technique with different parameters.

The remainder of the paper is organized as follows. Section 2 outlines related work. In Section 3, formal models for event logs are presented. We explain the measures in Section 4. The experiments are described in Section 5, and Section 6 concludes the paper.

2 Related Work

In process mining, the research field of confidentiality and privacy is growing in importance. In [2], *Responsible Process Mining* (RPM) is introduced as the sub-discipline focusing on possible negative side-effects of applying process mining. In [12], the authors propose a privacy-preserving system design for process mining, where a user-centered view is considered to track personal data. In [18], a framework is introduced providing a generic scheme for confidentiality in process mining. In [14], the authors introduce a privacy-preserving method for discovering roles from event data. In [6], the authors apply k-anonymity and t-closeness on event data to preserve the privacy of *resources*. In [11], the notion of *differential privacy* is employed to preserve the privacy of *cases*. In [17], the TLKC-privacy model is introduced to deal with high variability issues in event logs for applying group-based anonymization techniques. In [5], a secure multi-party computation solution is proposed for preserving privacy in an inter-organizational setting. In [13], the authors analyze data privacy and utility requirements for healthcare event data, and the suitability of privacy-preserving techniques is assessed. In [16], privacy metadata in process mining are discussed and a privacy extension for the XES standard (https://xes-standard.org/) is proposed.

Most related to our work is [22], where a uniqueness-based measure is proposed to evaluate the re-identification risk of event logs. Privacy quantification in data mining is a well-developed field where the effectiveness of privacy preservation techniques is evaluated from different aspects such as *dissimilarity* [3], *information loss* [7], *discernibility* [8], and etc. We utilize the experiences achieved in this field and propose a trade-off approach as suggested in [4].

3 Preliminaries

In this section, we provide formal definitions for event logs used in the remainder. An event log is a collection of events, composed of different attributes, such that they are uniquely identifiable. In this paper, we consider only the mandatory attributes of events including *case identifier*, *activity name*, and *timestamp*. Accordingly, we define a simple event, trace, and event log. In the following, we introduce some basic concepts and notations.

Let A be a set. A^* is the set of all finite sequences over A, and $\mathcal{B}(A)$ is the set of all multisets over the set A. For $A_1, A_2 \in \mathcal{B}(A)$, $A_1 \subseteq A_2$ if for all $a \in A$, $A_1(a) \leq A_2(a)$. 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 $1 \leq i \leq n$, and $|\sigma| = n$. $a \in \sigma \Leftrightarrow a = a_i$ for $1 \leq i \leq n$. 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., $[a \in \langle x, y, z, x, y \rangle] = [x^2, y^2, z]$.

Definition 1 (Simple Event). *A simple event is a tuple $e = (c, a, t)$, where $c \in \mathcal{C}$ is the case identifier, $a \in \mathcal{A}$ is the activity associated to event e, and $t \in \mathcal{T}$ is the timestamp of event e. $\pi_X(e)$ is the projection of event e on the attribute from domain X, e.g., $\pi_{\mathcal{A}}(e) = a$. We call $\xi = \mathcal{C} \times \mathcal{A} \times \mathcal{T}$ the event universe.*

Definition 2 (Simple Trace). *Let ξ be the universe of events. A trace $\sigma = \langle e_1, e_2, ..., e_n \rangle$ in an event log is a sequence of events, i.e., $\sigma \in \xi^*$, s.t., for each $e_i, e_j \in \sigma$: $\pi_{\mathcal{C}}(e_i) = \pi_{\mathcal{C}}(e_j)$, and $\pi_{\mathcal{T}}(e_i) \leq \pi_{\mathcal{T}}(e_j)$ if $i < j$. A simple trace is a trace where all the events are projected on the activity attribute, i.e., $\sigma \in \mathcal{A}^*$.*

Definition 3 (Simple Event Log). *A simple event log is a multiset of simple traces, i.e., $L \in \mathcal{B}(\mathcal{A}^*)$. We assume each trace in an event log belongs to an individual and $\sigma \neq \langle\rangle$ if $\sigma \in L$. $A_L = \{a \in \mathcal{A} \mid \exists_{\sigma \in L} a \in \sigma\}$ is the set of activities in the event log L. $\tilde{L} = \{\sigma \in L\}$ is the set of unique traces (variants) in the event log L. We denote $\mathcal{U}_L$ as the universe of event logs.*

Definition 4 (Trace Frequency). *Let L be an event log, $f_L \in \tilde{L} \rightarrow [0, 1]$ is a function which retrieves the relative frequency of a trace in the event log L, i.e., $f_L(\sigma) = {}^{L(\sigma)}/_{|L|}$ and $\sum_{\sigma \in \tilde{L}} f_L(\sigma) = 1$.*

Definition 5 (Event Log Entropy). *$ent \in \mathcal{U}_L \rightarrow \mathbb{R}_{\geq 0}$ is a function which retrieves the entropy of traces in an event log, s.t., for $L \in \mathcal{U}_L$, $ent(L) = -\sum_{\sigma \in \tilde{L}} f_L(\sigma) log_2 f_L(\sigma)$. We denote $max\_ent(L)$ as the maximal entropy achieved when all the traces in the event log are unique, i.e., $|\tilde{L}| = |L|$.*

4 Privacy Quantification

We employ a *risk-utility* model for quantifying privacy in process mining where *disclosure risk* and *utility loss* are measured to assess the effectiveness of privacy preservation techniques before and after applying the techniques.

4.1 Disclosure Risk

In this subsection, we introduce *identity/case disclosure* and *attribute/trace disclosure* for quantifying disclosure risk of event logs. Identity disclosure quantifies how uniquely the trace owners, i.e., cases, can be re-identified. Attribute disclosure quantifies how confidently the sensitive attributes of cases (as individuals) can be specified. As discussed in [17], traces play the role of both quasi-identifiers and sensitive attributes. That is, a complete sequence of activities, which belongs to a case, is sensitive person-specific information. At the same time, knowing a part of this sequence, as background knowledge, can be exploited to re-identify the trace owner. In a simple event log, traces, i.e., sequence of activities, are the only available information. Therefore, *attribute disclosure* can be seen as *trace disclosure*.

In the following, we define *set*, *multiset*, and *sequence* as three types of background knowledge based on traces in simple event logs that can be exploited for uniquely re-identifying the trace owners or certainly specifying their complete sequence of activities. Moreover, we consider a size for different types of background knowledge as their power, e.g, the *set* background knowledge of size 3 is more powerful than the same type of background knowledge of size 2. Note that the assumed types of background knowledge are the most general ones, and more types can be explored. However, the general approach will be the same.

Definition 6 (Background Knowledge 1 - Set). *In this scenario, we assume that an adversary knows a subset of activities performed for the case, and this information can lead to the identity or attribute disclosure. Let L be an event log, and A_L be the set of activities in the event log L. We formalize this background knowledge by a function $proj^L_{set} \in 2^{A_L} \to 2^L$. For $A \subseteq A_L$, $proj^L_{set}(A) = [\sigma \in L \mid A \subseteq \{a \in \sigma\}]$. We denote $cand^l_{set}(L) = \{A \subseteq A_L \mid |A| = l \wedge proj^L_{set}(A) \neq []\}$ as the set of all subsets over the set A_L of size l for which there exists matching traces in the event log.*

Definition 7 (Background Knowledge 2 - Multiset). *In this scenario, we assume that an adversary knows a sub-multiset of activities performed for the case, and this information can lead to the identity or attribute disclosure. Let L be an event log, and A_L be the set of activities in the event log L. We formalize this background knowledge by a function $proj^L_{mult} \in \mathcal{B}(A_L) \to 2^L$. For $A \in \mathcal{B}(A_L)$, $proj^L_{mult}(A) = [\sigma \in L \mid A \subseteq [a \in \sigma]]$. We denote $cand^l_{mult}(L) = \{A \in \mathcal{B}(A_L) \mid |A| = l \wedge proj^L_{mult}(A) \neq []\}$ as the set of all sub-multisets over the set A_L of size l for which there exists matching traces in the event log.*

Definition 8 (Background Knowledge 3 - Sequence). *In this scenario, we assume that an adversary knows a subsequence of activities performed for the case, and this information can lead to the identity or attribute disclosure. Let L be an event log, and A_L be the set of activities in the event log L. We formalize this background knowledge by a function $proj^L_{seq} \in A^*_L \to 2^L$. For $\sigma \in A^*_L$, $proj^L_{seq}(\sigma) = [\sigma' \in L \mid \sigma \sqsubseteq \sigma']$. We denote $cand^l_{seq}(L) = \{\sigma \in A^*_L \mid |\sigma| = l \wedge proj^L_{seq}(\sigma) \neq []\}$ as the set of all subsequences of size (length) l, based on the activities in A_L, for which there exists matching traces in the event log.*

Example 1 (background knowledge) *Let* $L = [\langle a,b,c,d\rangle^{10}, \langle a,c,b,d\rangle^{20}, \langle a,d,b,d\rangle^{5}, \langle a,b,d,d\rangle^{15}]$ *be an event log.* $A_L = \{a,b,c,d\}$ *is the set of unique activities, and* $cand^2_{set}(L) = \{\{a,b\},\{a,c\},\{a,d\},\{b,c\},\{b,d\},\{d,c\}\}$ *is the set of candidates of the set background knowledge of size* 2. *For* $A = \{b,d\} \in cand^2_{set}(L)$ *as a candidate of the set background knowledge of size* 2, $proj^L_{set}(A) = [\langle a,b,c,d\rangle^{10}, \langle a,c,b,d\rangle^{20}, \langle a,d,b,d\rangle^{5}, \langle a,b,d,d\rangle^{15}]$. *For* $A = [b,d^2]$ *as a candidate of the multiset background knowledge,* $proj^L_{mult}(A) = [\langle a,d,b,d\rangle^{5}, \langle a,b,d,d\rangle^{15}]$. *Also, for* $\sigma = \langle b,d,d\rangle$ *as a candidate of the sequence background knowledge,* $proj^L_{seq}(\sigma) = [\langle a,b,d,d\rangle^{15}]$.

As Example 1 shows, the strength of background knowledge from the weakest to the strongest w.r.t. the type is as follows: *set*, *multiset*, and *sequence*, i.e., given the event log L, $proj^L_{seq}(\langle b,d,d\rangle) \subseteq proj^L_{mult}([b,d^2]) \subseteq proj^L_{set}(\{b,d\})$.

Identity (Case) Disclosure We use the uniqueness of traces w.r.t. the background knowledge of size l to measure the corresponding case disclosure risk in an event log. Let L be an event log and $type \in \{set, mult, seq\}$ be the type of background knowledge. The case disclosure based on the background knowledge *type* of size l is calculated as follows:

$$cd^l_{type}(L) = \sum_{x \in cand^l_{type}(L)} \frac{1/|proj^L_{type}(x)|}{|cand^l_{type}(L)|} \tag{1}$$

Equation (1) calculates the average uniqueness based on the candidates of background knowledge, i.e., $x \in cand^l_{type}(L)$. Note that we consider equal weights for the candidates of background knowledge. However, they can be weighted based on the various criteria, e.g., the sensitivity of the activities included. One can also consider the worst case, i.e., the maximal uniqueness, rather than the average value.

Example 2 (insufficiency of case disclosure analysis) *Consider* $L_1 = [\langle a,b,c,d\rangle, \langle a,c,b,d\rangle, \langle a,b,c,c,d\rangle, \langle a,b,b,c,d\rangle]$ *and* $L_2 = [\langle a,b,c,d\rangle^4, \langle e,f\rangle^4, \langle g,h\rangle^4]$ *as two event event logs.* $A_{L_1} = \{a,b,c,d\}$ *and* $A_{L_2} = \{a,b,c,d,e,f,g,h\}$ *are the set of unique activities in* L_1 *and* L_2, *respectively.* $cand^1_{set}(L_1) = \{\{a\},\{b\},\{c\},\{d\}\}$ *and* $cand^1_{set}(L_2) = \{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\},\{h\}\}$ *are the set of candidates of the set background knowledge of size* 1. *Both event logs have the same value as the case disclosure for the set background knowledge of size* 1 *(*$cd^1_{set}(L_1) = cd^1_{set}(L_2) = 1/4$*). However, in* L_2, *the complete sequence of activities performed for a victim case is disclosed by knowing only one activity without uniquely identifying the corresponding trace.*

Example 2 clearly shows that measuring the uniqueness alone is insufficient to demonstrate disclosure risks in event logs and the uncertainty in the set of sensitive attributes matching with the assumed background knowledge need to be measured, as well. In the following, we define a measure to quantify the uncertainty in the set of matching traces. Note that, the same approach can be exploited to quantify the disclosure risk of any other sensitive attribute matching with some background knowledge.

Attribute (Trace) Disclosure We use the entropy of matching traces w.r.t. background knowledge of size l to measure the corresponding trace disclosure risk in an event log. Let L be an event log and $type \in \{set, mult, seq\}$ be the type of background knowledge. The trace disclosure based on the background knowledge $type$ of size l is calculated as follows:

$$td^l_{type}(L)=1 - \sum_{x \in cand^l_{type}(L)} \frac{ent(proj^L_{type}(x))/max\_ent(proj^L_{type}(x))}{|cand^l_{type}(L)|} \quad (2)$$

In (2), $max\_ent(proj^L_{type}(x))$ is the maximal entropy for the matching traces based on the type and size of background knowledge, i.e., uniform distribution of the matching traces. As discussed for (1), in (2), we also assume equal weights for the candidates of background knowledge. However, one can consider different weights for the candidates. Also, the worst case, i.e., the minimal entropy, rather than the average entropy can be considered.

The trace disclosure of the event logs in Example 2 is as follows: $td^1_{set}(L_1) = 0$ (the multiset of matching traces has the maximal entropy) and $td^1_{set}(L_2) = 1$ (the entropy of matching traces is 0). These results distinguish the disclosure risk of the event logs.

4.2 Utility Loss

In this subsection, we introduce a measure based on the *earth mover's distance* [19] for quantifying the utility loss after applying a privacy preservation technique to an event log. The *earth mover's distance* describes the distance between two distributions. In an analogy, given two piles of earth, it expresses the effort required to transform one pile into the other. First, we introduce the concept of reallocation indicating how an event log is transformed into another event log. Then, we define a trace distance function expressing the cost of transforming one trace into another one. Finally, we introduce the utility loss measure that indicates the entire cost of transforming an event log to another one using the introduced reallocation and distance functions.

Reallocation Let L be the original event log and L' be an anonymized event log derived from the original event log. We introduce $r \in \tilde{L} \times \tilde{L}' \rightarrow [0, 1]$ as a function that indicates the movement of frequency between two event logs. $r(\sigma, \sigma')$ describes the relative frequency of $\sigma \in \tilde{L}$ that should be transformed to $\sigma' \in \tilde{L}'$. To make sure that a reallocation function properly transforms L into L', the frequency of each $\sigma \in \tilde{L}$ should be considered, i.e., for all $\sigma \in \tilde{L}$, $f_L(\sigma) = \sum_{\sigma' \in \tilde{L}'} r(\sigma, \sigma')$. Similarly, the probability mass of traces $\sigma' \in \tilde{L}'$ should be preserved, i.e., for all $\sigma' \in \tilde{L}'$, $f_{L'}(\sigma') = \sum_{\sigma \in \tilde{L}} r(\sigma, \sigma')$. We denote $\mathcal{R}$ as the set of all reallocation functions which depends on L and L'.

Trace Distance A trace distance function $d \in \mathcal{A}^* \times \mathcal{A}^* \rightarrow [0, 1]$ expresses the distance between traces. This function is 0 if and only if two traces are

Table 1: The dissimilarity between two event logs based on the earth mover's distance assuming r_s as a reallocation function and d_s as the normalized Levenshtein distance.

$r_s \cdot d_s$	$\langle a,b,c,d\rangle$	$\langle a,c,b,d\rangle$	$\langle a,e,c,d\rangle^{49}$	$\langle a,e,b,d\rangle^{49}$
$\langle a,b,c,d\rangle^{50}$	$0.01 \cdot 0$	$0 \cdot 0.5$	$0.49 \cdot 0.25$	$0 \cdot 0.5$
$\langle a,c,b,d\rangle^{50}$	$0 \cdot 0.5$	$0.01 \cdot 0$	$0 \cdot 0.5$	$0.49 \cdot 0.25$

equal, i.e., $d(\sigma,\sigma') = 0 \iff \sigma = \sigma'$. This function should also be symmetrical, i.e., $d(\sigma,\sigma') = d(\sigma',\sigma)$. Different distance functions can be considered satisfying these conditions. We use the *normalized string edit distance* (Levenshtein) [9].

Utility Loss Let L be an original event log, and L' be an anonymized event log derived from the original event log. Several reallocation functions might exist. However, the *earth mover's distance* problem aims to express the shortest distance between the two event logs, i.e., the least mass movement over the least distance between traces. Therefore, the difference between L and L' using a reallocation function r is the inner product of reallocation and distance. The data utility preservation is considered as $du(L,L') = 1 - \min_{r\in\mathcal{R}} ul(r,L,L')$.

$$ul(r,L,L') = r \cdot d = \sum_{\sigma\in\tilde{L}} \sum_{\sigma'\in\tilde{L}'} r(\sigma,\sigma')d(\sigma,\sigma') \tag{3}$$

Example 3 (using earth mover's distance to calculate dissimilarity between event logs) *Let* $L = [\langle a,b,c,d\rangle, \langle a,c,b,d\rangle, \langle a,e,c,d\rangle^{49}, \langle a,e,b,d\rangle^{49}]$ *and* $L' = [\langle a,b,c,d\rangle^{50}, \langle a,c,b,d\rangle^{50}]$ *be the original and aninymized event logs, respectively. Table 1 shows the calculations assuming* r_s *as a reallocation function and* d_s *as the normalized Levenshtein distance, e.g.,* $r_s(\langle a,b,c,d\rangle, \langle a,e,c,d\rangle) = 0.49$ *and* $d_s(\langle a,b,c,d\rangle, \langle a,e,c,d\rangle) = 0.25$. $ul(r_s,L,L') = 0.24$ *and* $du(L,L') = 0.76$.

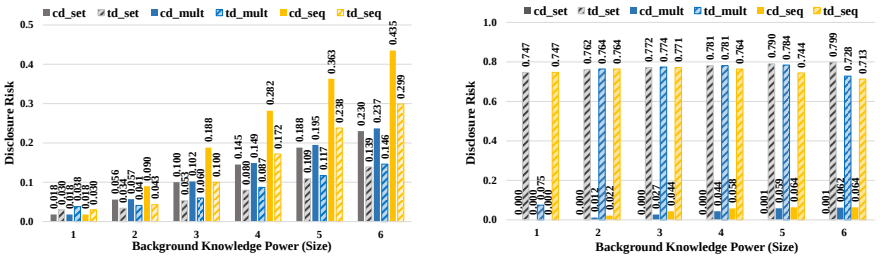
5 Experiments

In this section, we demonstrate the experiments on real-life event logs to advocate the proposed measures. We employ two human-centered event logs, where the *case identifiers* refer to individuals. Sepsis-Cases [10] is a real-life event log containing events of sepsis cases from a hospital. BPIC-2017-APP [21] is also a real-life event log pertaining to a loan application process of a Dutch financial institute. We choose these event logs because they are totally different w.r.t. the uniqueness of traces. Table 2 shows the general statistics of these event logs. Note that *variants* are the unique traces, and $trace\_uniquness = \#variants/\#traces$. The implementation as a Python program is available on GitHub.[1]

Table 2: The general statistics of the event logs used in the experiments.

Event Log	#traces	#variants	#events	#unique_activities	trace_uniqueness
Sepsis-Cases [10]	1050	845	15214	16	80%
BPIC-2017-APP [21]	31509	102	239595	10	0.3%

[1] https://github.com/m4jidRafiei/privacy_quantification



(a) Sepsis-Cases [10]. (b) BPIC-2017-APP [21].

Fig. 2: Analyses of the case disclosure (*cd*) and the trace disclosure (*td*) based on the three types of background knowledge (i.e., *set*, *mult*, and *seq*) when we vary the background knowledge power (size) from 1 to 6. For example, in the Sepsis-Cases event log, the case disclosure risk of the background knowledge *seq* (*cd_seq*) of size 3 is 0.188.

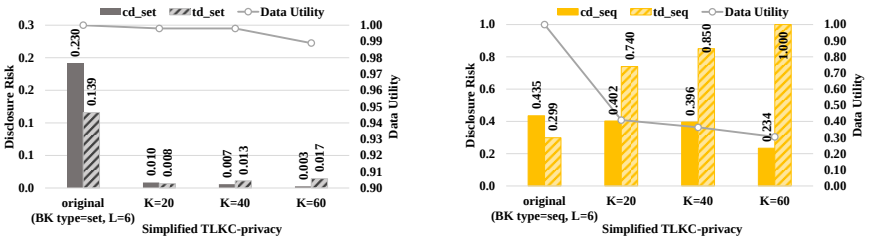
5.1 Disclosure Risk Analysis

In this subsection, we show the functionality of the proposed measures for disclosure risk analysis. To this end, we consider three types of background knowledge (*set*, *multiset*, and *sequence*) and vary the background knowledge power (size) from 1 to 6. Figure 2a shows the results for the Sepsis-Cases event log where the uniqueness of traces is high. As shown, the disclosure risks are higher for the more powerful background knowledge w.r.t. the *type* and *size*.

Figure 2b demonstrates the results for the BPIC-2017-APP event log, where the uniqueness of traces is low. As shown, the case disclosure risk is low, which is expected regarding the low uniqueness of traces. However, the trace disclosure risk is high which indicates low entropy (uncertainty) of the traces. Moreover, for the stronger background knowledge w.r.t. the size, one can assume a higher case disclosure risk. However, the trace disclosure risk is correlated with the entropy of the sensitive attribute values and can be a high value even for weak background knowledge. The above-mentioned analyses clearly show that uniqueness alone cannot reflect the actual disclosure risk in an event log.

5.2 Utility Loss Analysis

In this subsection, we demonstrate the functionality of the proposed measure in Section 4.2 for quantifying data utility preservation after applying a privacy preservation technique. We use PPDP-PM [15] as a privacy preservation tool for process mining to apply the TLKC-privacy model [17] to a given event log. The TLKC-privacy model is a group-based privacy preservation technique which provides a good level of flexibility through various parameters such as the type and size (power) of background knowledge. The T in this model refers to the accuracy of timestamps in the privacy-aware event log, L refers to the power



(a) Using *set* as background knowledge. (b) Using *sequence* as background knowledge.

Fig. 3: The utility loss and disclosure risk analyses for the Sepsis-Cases event log where the background knowledge is *set* or *sequence*, and the power (size) of background knowledge is 6.

of background knowledge[2], K refers to the k in the k-anonymity definition [20], and C refers to the bound of confidence regarding the sensitive attribute values in an equivalence class.

Assuming *set* (Definition 6) and *sequence* (Definition 8) as the types of background knowledge, we apply the TLKC-privacy model to the Sepsis-Cases event log with the following parameters: L = 6 (as the maximum background knowledge power in our experiments), K = $\{20, 40, 60\}$, C = 1 (there is no additional sensitive attribute in a simple event log), and T is set to the maximal precision (T has no effect on a simple event log). That is, the TLKC-privacy model is simplified to k-anonymity where the *quasi-identifier* (background knowledge) is the *set* or *sequence* of activities. Table 3 demonstrates the general statistics of the event logs before and after applying the privacy preservation technique.

Figure 3a shows disclosure risk and data utility analyses for the background knowledge *set*, and Fig. 3b shows the same analyses for the background knowledge *sequence*. In both types of background knowledge, the data utility value decreases. For the stronger background knowledge, i.e., *sequence*, the utility loss is much higher which is expected w.r.t. the general statistics in Table 3. However, the data utility for the weaker background knowledge remains high which again complies with the general statistics. Note that since we apply k-anonymity (sim-

Table 3: The general statistics before and after applying the TLKC-privacy model.

Event Log			#traces	#variants	#events	#unique_activities
Original Sepsis-Cases			1050	845	15214	16
Anonymized Sepsis-Cases	BK type=set BK size (L)=6	K=20	1050	842	15103	12
		K=40	1050	842	14986	11
		K=60	1050	818	14809	11
	BK type=seq BK size (L)=6	K=20	1050	34	3997	6
		K=40	1050	33	4460	5
		K=60	1050	18	3448	4

[2]Note that this L is identical to the l introduced as the power (size) of background knowledge and should not be confused with L as the event log notation.

plified TLKC-model) only *case disclosure*, which is based on the uniqueness of traces, decreases. Moreover, for the *sequence* background knowledge, higher values for K result in more similar traces. Therefore, the *trace disclosure* risk, in the anonymized event logs, drastically increases. These analyses demonstrate that privacy preservation techniques should consider different aspects of disclosure risk while balancing data utility preservation and sensitive data protection.

6 Conclusion

Event logs often contain highly sensitive information, and regarding the rules imposed by regulations, these sensitive data should be analyzed responsibly. Therefore, privacy preservation in process mining is recently receiving more attention. Consequently, new measures need to be defined to evaluate the effectiveness of the privacy preservation techniques both from the sensitive data protection and data utility preservation point of views. In this paper, using a trade-off approach, we introduced two measures for quantifying disclosure risks: *identity/case disclosure* and *attribute/trace disclosure*, and one measure for quantifying *utility loss*. The introduced measures were applied to two real-life event logs. We showed that even simple event logs could reveal sensitive information. Moreover, for the first time, the effect of applying a privacy preservation technique on *data utility* rather than *result utility* was explored. The *data utility* measure is based on the *earth mover's distance* and can be extended to evaluate the utility w.r.t. the different perspectives of process mining, e.g., *time*, *resource*, etc.

Acknowledgment

Funded under the Excellence Strategy of the Federal Government and the Länder. We also 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. Bertino, E., Fovino, I.N., Provenza, L.P.: A framework for evaluating privacy preserving data mining algorithms. Data Min. Knowl. Discov. **11**(2), 121–154 (2005)
4. Bertino, E., Lin, D., Jiang, W.: A survey of quantification of privacy preserving data mining algorithms. In: Aggarwal, C.C., Yu, P.S. (eds.) Privacy-Preserving Data Mining - Models and Algorithms, Advances in Database Systems, vol. 34, pp. 183–205. Springer (2008)

5. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Secure multi-party computation for inter-organizational process mining. In: Enterprise, Business-Process and Information Systems Modeling - 21st International Conference, BPMDS. Springer (2020)
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 (2019)
7. Iyengar, V.S.: Transforming data to satisfy privacy constraints. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 279–288. ACM (2002)
8. Jr., R.J.B., Agrawal, R.: Data privacy through optimal k-anonymization. In: Proceedings of the 21st International Conference on Data Engineering, ICDE (2005)
9. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
10. Mannhardt, F.: Sepsis cases-event log. Eindhoven University of Technology (2016)
11. 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)
12. 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)
13. Pika, A., Wynn, M.T., Budiono, S., ter Hofstede, A.H., van der Aalst, W.M.P., Reijers, H.A.: Privacy-preserving process mining in healthcare. International Journal of Environmental Research and Public Health **17**(5), 1612 (2020)
14. 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)
15. Rafiei, M., van der Aalst, W.M.P.: Practical aspect of privacy-preserving data publishing in process mining. In: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2020 co-located with the 18th International Conference on Business Process Management (BPM 2020). CEUR-WS.org (2020)
16. Rafiei, M., van der Aalst, W.M.P.: Privacy-preserving data publishing in process mining. In: Business Process Management Forum - BPM Forum 2020, Seville, Spain, September 13-18. pp. 122–138. Springer (2020)
17. Rafiei, M., Wagner, M., van der Aalst, W.M.P.: TLKC-privacy model for process mining. In: Research Challenges in Information Science - 14th International Conference, RCIS. pp. 398–416. Springer International Publishing (2020)
18. Rafiei, M., von Waldthausen, L., van der Aalst, W.M.P.: Supporting condentiality in process mining using abstraction and encryption. In: Data-Driven Process Discovery and Analysis - 8th IFIP WG 2.6 International Symposium, SIMPDA 2018, and 9th International Symposium, SIMPDA 2019, Revised Selected Papers (2019)
19. Rüschendorf, L.: The wasserstein distance and approximation theorems. Probability Theory and Related Fields **70**(1), 117–129 (1985)
20. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(05), 557–570 (2002)
21. Van Dongen, B.F.: BPIC 2017. Eindhoven University of Technology (2017)
22. von Voigt, S.N., Fahrenkrog-Petersen, S.A., Janssen, D., Koschmider, A., Tschorsch, F., Mannhardt, F., Landsiedel, O., Weidlich, M.: Quantifying the re-identification risk of event logs for process mining - empiricial evaluation paper. In: Advanced Information Systems Engineering, CAiSE (2020)