Quantifying Temporal Privacy Leakage in Continuous Event Data Publishing*

Majid Rafiei[1] ✉, Gamal Elkoumy[2], and Wil M.P. van der Aalst[1]

[1] Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
[2] University of Tartu, Tartu, Estonia

Abstract. Process mining employs event data extracted from different types of information systems to discover and analyze actual processes. Event data often contain highly sensitive information about the people who carry out activities or the people for whom activities are performed. Therefore, privacy concerns in process mining are receiving increasing attention. To alleviate privacy-related risks, several privacy preservation techniques have been proposed. Differential privacy is one of these techniques which provides strong privacy guarantees. However, the proposed techniques presume that event data are released in only one shot, whereas business processes are continuously executed. Hence, event data are published repeatedly, resulting in additional risks. In this paper, we demonstrate that continuously released event data are not independent, and the correlation among different releases can result in privacy degradation when the same differential privacy mechanism is applied to each release. We quantify such privacy degradation in the form of temporal privacy leakages. We apply continuous event data publishing scenarios to real-life event logs to demonstrate privacy leakages.

Keywords: privacy preservation · differential privacy · process mining · privacy leakage · event data.

1 Introduction

Process mining forms a family of techniques used to analyze operational processes of organizations. These techniques use event logs extracted from information systems. An event log contains sequences of events, and each event reflects the execution of an activity with some attributes, e.g., the timestamp at which the activity was performed or the case for which the activity was performed. Some event attributes may refer to individuals, e.g., patients or customers, thus raising privacy concerns.

Data regulations, e.g., GDPR [1], limit the analysis of sensitive event logs. To circumvent such restrictions, Privacy-Preserving Process Mining (PPPM) [8] proposes techniques to guarantee privacy preservation, e.g., *Differential Privacy*

(a) An event log containing trace variants with their frequencies, e.g., $trace_1$ happened 4 times.

	$i=1$	$i=2$	$i=3$	...
$trace_1$	$\langle\blacktriangleright,a,b,c\rangle^4$	$\langle\blacktriangleright,a,b,c,f\rangle^4$	$\langle\blacktriangleright,a,b,c,f,\blacksquare\rangle^4$	...
$trace_2$	$\langle\blacktriangleright,d,a,b\rangle^2$	$\langle\blacktriangleright,d,a,b,c\rangle^2$	$\langle\blacktriangleright,d,a,b,c,g\rangle^2$	...
$trace_3$	$\langle\blacktriangleright,d,a,f\rangle^2$	$\langle\blacktriangleright,d,a,f,c\rangle^2$	$\langle\blacktriangleright,d,a,f,c,h\rangle^2$	...
$trace_4$	$\langle\blacktriangleright,a,b,f\rangle^4$	$\langle\blacktriangleright,a,b,f,c\rangle^4$	$\langle\blacktriangleright,a,b,f,c,\blacksquare\rangle^4$	...

(b) The actual frequency of trace variants at each release point.

	$i=1$	$i=2$	$i=3$	...
...	...	...	...	...
$\langle\blacktriangleright,a,b,c\rangle$	4	0	0	...
$\langle\blacktriangleright,a,b,c,f\rangle$	0	4	0	...
$\langle\blacktriangleright,a,b,c,f,\blacksquare\rangle$	0	0	4	...
$\langle\blacktriangleright,a,b,f\rangle$	4	0	0	...
$\langle\blacktriangleright,a,b,f,c\rangle$	0	4	0	...
$\langle\blacktriangleright,a,b,f,c,\blacksquare\rangle$	0	0	4	...
$\langle\blacktriangleright,d,a,f\rangle$	2	0	0	...
$\langle\blacktriangleright,d,a,f,c\rangle$	0	2	0	...
$\langle\blacktriangleright,d,a,f,c,h\rangle$	0	0	2	...
$\langle\blacktriangleright,d,a,b\rangle$	2	0	0	...
$\langle\blacktriangleright,d,a,b,c\rangle$	0	2	0	...
$\langle\blacktriangleright,d,a,b,c,g\rangle$	0	0	2	...
...	...	...	...	...

(c) Differentially private frequency of trace variants at each release point.

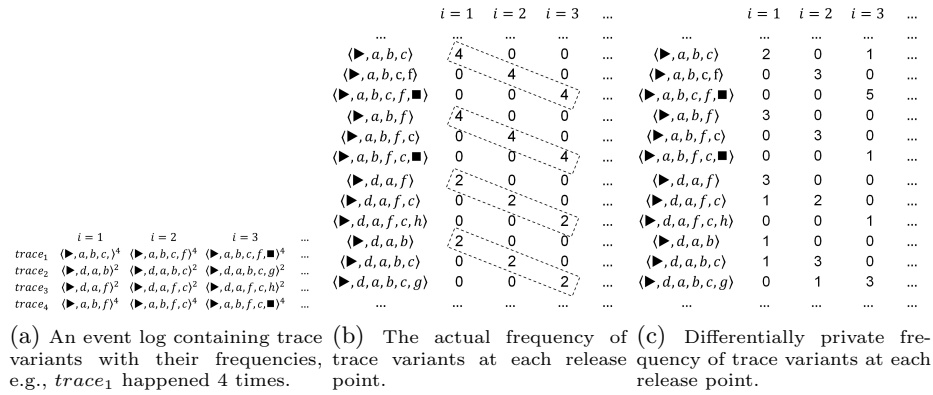|	$i=1$	$i=2$	$i=3$	...
...	...	...	...	...
$\langle\blacktriangleright,a,b,c\rangle$	2	0	1	...
$\langle\blacktriangleright,a,b,c,f\rangle$	0	3	0	...
$\langle\blacktriangleright,a,b,c,f,\blacksquare\rangle$	0	0	5	...
$\langle\blacktriangleright,a,b,f\rangle$	3	0	0	...
$\langle\blacktriangleright,a,b,f,c\rangle$	0	3	0	...
$\langle\blacktriangleright,a,b,f,c,\blacksquare\rangle$	0	0	1	...
$\langle\blacktriangleright,d,a,f\rangle$	3	0	0	...
$\langle\blacktriangleright,d,a,f,c\rangle$	1	2	0	...
$\langle\blacktriangleright,d,a,f,c,h\rangle$	0	0	1	...
$\langle\blacktriangleright,d,a,b\rangle$	1	0	0	...
$\langle\blacktriangleright,d,a,b,c\rangle$	1	3	0	...
$\langle\blacktriangleright,d,a,b,c,g\rangle$	0	1	3	...
...	...	...	...	...

Fig. 1: Continuous event data release under temporal correlations.

(DP) [4] or *group-based* privacy preservation techniques, i.e., k-anonymity and its extensions [17]. DP works based on a noise injection mechanism that injects noise into published data to ensure that modifying a single user's record in the original data has a small impact on the published data. Such an impact is bounded by ϵ, so-called *privacy budget*. The smaller values of ϵ result in more noise injection and less privacy leakage.

Process mining techniques, such as *process discovery* and *conformance checking*, discover and analyze the control-flow of a process which is based on the distribution of *trace variants*, i.e., the control-flow aspect of an event log. A trace variant is a sequence of activities performed for a case. Various privacy mechanisms have been proposed to anonymize the control-flow aspect of an event log [15,17,11,9]. These approaches consider only a one-shot data release. However, business processes are continuously executed, stressing the need for Continuous Event Data Publishing (CEDP) [18]. In CEDP, events are collected up to a certain point in time or meeting a certain condition and published in the form of event logs. This publishing scenario is done continuously based on a *time-window*, e.g., daily, or a *count-window*, e.g., each new release contains one new event per trace.

CEDP may result in violating the provided privacy guarantees provided for separate releases of event logs if there are correlations among continuously released event logs. For example, consider the continuous release of the event data in Fig. 1. An organization collects its business process event data in the form of trace variants frequencies up to a release point and publishes the differentially private trace variant frequencies. Suppose that each case, e.g., a patient, contributes to only one trace variant at each release point and the trace variant of a case is the sensitive information that needs to be protected. Note that the trace variant of a case is considered sensitive information because it contains the entire sequence of activities performed for the case. For example, in the healthcare context, the activities are treatment-related, and sequences of activities can be exploited to determine the health conditions of cases, e.g., their diseases.

In order to provide an ϵ-DP guarantee, one needs to hide the participation of an individual in the released output. To this end, the output gets noisified. The amount of noise is determined by the privacy parameter ϵ and the *sensitivity* of a query. The sensitivity indicates how much uncertainty is required to hide the contribution of one individual to the query. Here, the query is the frequency of each trace variant. Since the modification of only one frequency value, i.e., the contribution of one individual, at a specific release point i in Fig. 1b affects only one trace variant, the sensitivity is set to 1. Adding noise drawn from a *Laplacian distribution* with scale $1/\epsilon$, where the sensitivity value is in the numerator, to perturb each frequency achieves ϵ-DP at each release point [5], as in Fig. 1c.

However, that may not be true with the existence of *temporal correlations*. For example, as shown in Fig. 1b, the frequency of each trace variant at release point i is not independent of the frequency of its prefix at release point $i-1$. Thus, adding Laplacian noise with scale $1/\epsilon$ at the release point $i=3$ only achieves 3ϵ-DP, which is three times weaker than the first provided guarantee. One can interpret this situation based on *group differential privacy*, where correlated data are protected as a group [3]. Moreover, due to the nature of business processes, traces may have a particular subtrace pattern, such as "activity b always follows activity a". Such temporal correlations can be formulated as conditional probabilities to analyze their effect on the provided privacy guarantees by DP mechanisms [2].

In this paper, we adapt the approach introduced in [2]. In [2], the authors assume that probability matrices explaining the correlations between different releases are given. However, we exploit some characteristics of CEDP to obtain such probabilities. We show that different event data publishing scenarios can affect the correlations and the privacy leakage results. We also investigate the effect of specific event log characteristics on the correlations and privacy leakages. Our proposal utilizes a transition system to model traces in the form of states at each point of release. Particularly, we focus on a full-history transition system, so-called prefix automaton, where each state represents a prefix of a trace from the start point until the state. We utilize such a transition system to obtain conditional probabilities between states (traces) at each release point.

The paper is structured as follows. Section 2 discusses related work. Section 3 introduces basic notations and formal definitions. Section 4 demonstrates our approach to quantify temporal privacy leakage in CEDP. In Section 5, we provide experiments based on real-life public event logs. Section 6 concludes the paper and discusses some limitations of the approach.

2   Related Work

A plethora of studies has been conducted to provide privacy for process mining. In [8], the authors studied the requirements and challenges of providing privacy-preserving process mining. Several studies applied differential privacy to publish event logs. Mannhardt et al. [15] applied differential privacy to anonymize queries over event logs. PRIPEL [10] applies differential privacy to anonymize timestamps of event logs. SaCoFa [11] integrates differential privacy with event

log semantics to anonymize the control flow of event logs. In [9], the authors applied differential privacy to event logs in order to prevent singling out individuals using the prefixes/suffixes of their traces. However, all of the above mechanisms assume one-shot data publishing.

Dwork et al. first studied differential privacy under continual observation, and they presented user-level [7] and event level [6] privacy. Several studies have investigated applying differential privacy in continuous data publishing. Kellaris et al. [14] studied the problem of infinite sequences. Fan et al. [12] studied differential privacy with a real-time publishing setting. Cao et al. [2] quantified the risk of using differential privacy under temporal correlation to release continuous location data. A framework for quantifying risk when publishing only one event log for process mining has been studied in [16]. To the best of our knowledge, no study has presented a risk quantification for differential privacy in continuous event data publishing. Although, in [18], the authors have elaborated possible attacks against continuous anonymized event data publishing, that work focuses on group-based privacy preservation techniques.

3   Preliminaries

In this section, we provide formal definitions for *event logs*, *transition systems*, and *differential privacy*, which will be used to explain the approach.

3.1   Event Log

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 $\sigma \in \{1, ..., n\} \to A$, represented as $\sigma = \langle a_1, a_2, ..., a_n \rangle$ where $a_i = \sigma(i)$ for any $1 \leq i \leq n$. $|\sigma|$ denotes the length of the sequence. Given A and B as two multisets, $A \uplus B$ is the sum over multisets, e.g., $[a^2, b^3] \uplus [b^2, c^2] = [a^2, b^5, c^2]$. A multiset set A can be represented as a set of tuples $\{(a, A(a)) | a \in A\}$ where $A(a)$ is the frequency of $a \in A$. For $\sigma_1, \sigma_2 \in A^*$, $\sigma_1 \sqsubseteq \sigma_2$ if σ_1 is a subsequence of σ_2, e.g., $\langle z, x, a, b \rangle \sqsubseteq \langle z, x, a, b, b, c, a, b, c, x \rangle$.

Definition 1 (Event). *An 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 with the event e, and $t \in \mathcal{T}$ is the timestamp of the event e. We call $\xi = \mathcal{C} \times \mathcal{A} \times \mathcal{T}$ the universe of events. Given an event $e = (c, a, t) \in \xi$, $\pi_{case}(e) = c$, $\pi_{act}(e) = a$, and $\pi_{time}(e) = t$.*

Note that ▶ and ■ are artificial start and end activities included in $\mathcal{A}$, i.e., $\{\blacktriangleright, \blacksquare\} \subset \mathcal{A}$. We assume that the case identifiers are dummy identifiers referring to individuals such as patients, workers, customers, etc. These identifiers cannot be exploited to directly re-identify individuals.

Definition 2 (Trace, Trace Variant). *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, s.t., for each $e_i, e_j \in \sigma$, $1 \leq i < j \leq n$: $\pi_{case}(e_i) = \pi_{case}(e_j)$, and $\pi_{time}(e_i) \leq \pi_{time}(e_j)$. A trace variant is a trace where all the events are projected on the activity attribute, i.e., $\sigma \in \mathcal{A}^*$.*

Definition 3 (Event Log). *An event log L is a set of case identifiers and their corresponding trace variants, i.e., $L \subseteq \mathcal{C} \times \mathcal{A}^*$. If $(c_1, \sigma_1),(c_2, \sigma_2) \in L$ and $c_1 = c_2$, then $\sigma_1 = \sigma_2$. $\tilde{L} = [\sigma \mid (c,\sigma) \in L]$ is the multiset representation of traces in L, i.e., $\tilde{L} \in \mathcal{B}(\mathcal{A}^*)$. Given $(c,\sigma) \in L$, $\pi_{case}((c,\sigma)) = c$ and $\pi_{trace}((c,\sigma)) = \sigma$.*

For instance, $L_1 = [(c_1, \langle \blacktriangleright, a, b, c, f, \blacksquare \rangle), (c_2, \langle \blacktriangleright, a, b, f, c, \blacksquare \rangle), (c_3, \langle \blacktriangleright, d, a, b, c, g \rangle), (c_4, \langle \blacktriangleright, d, a, f, c, h \rangle)]$ is an event log with artificial start activities for all the traces, and artificial end activities for the *complete traces*, i.e., the traces of c_1 and c_2. The traces of c_3 and c_4 are called *partial traces*, i.e., traces that have not yet reached the end activity. Note that our definition of an event log represent the control-flow perspective that is the focus of this work. In general, events of an event log may contain more attributes, e.g., *resources*, who perform activities.

3.2   Transition System

In this paper, we aim to quantify the privacy degradation in CEDP due to the correlations among event logs in different release points. To this end, we need to adopt an event log representation that helps to study these correlations. We consider a full-history transition system, so-called prefix automaton, as the event log representation. A transition system is one of the most basic process modeling notations which consists of states and transitions. States are represented by circles having unique labels, and transitions are represented by directed arcs with activity labels. Each transition connects two states. Figure 2 shows a transition system for the event log L_1. The labels of states are specified by a state representation function, which is defined as follows.

Definition 4 (State). *Given $\sigma \in \mathcal{A}^*$ as a trace and $0 \leq k \leq |\sigma|$ as a number, which indicates the number of events of σ that have occurred, $state(\sigma, k)$ is a function that produces a state.*

We define $state_{hd}()$ as the state representation functions describing the current state by the history of the case, i.e., given $\sigma = \langle a_1, a_2, ..., a_n \rangle$ as a trace of length n, $state_{hd}(\sigma, k) = \langle a_1, a_2, ..., a_k \rangle$.

Definition 5 (Event Log Representation). *Let $L \subseteq \mathcal{C} \times \mathcal{A}^*$ be an event log and $state()$ be a state representation function. $TS_{L,state()} = (S, A, T)$ is a transition system that represents L based on $state()$ where:*

- *$S = \{state(\sigma, k) \mid (c,\sigma) \in L \wedge 0 \leq k \leq |\sigma|\}$ is the state space;*
- *$A = \{\sigma(k) \mid (c,\sigma) \in L \wedge 1 \leq k \leq |\sigma|\}$ is the set of activities;*
- *$T = [(state(\sigma, k), \sigma(k+1), state(\sigma, k+1)) \mid (c,\sigma) \in L \wedge 0 \leq k < |\sigma|]$ is the multiset of transitions;*
- *$S^{start} = \{state(\sigma, 0) \mid (c,\sigma) \in L\}$ is the set of start states; and*
- *$S^{end} = \{state(\sigma, |\sigma|) \mid (c,\sigma) \in L \wedge \sigma(|\sigma|) = \blacksquare\}$ is the set of end states.*

Using $state_{hd}()$ as a state representation function, one can create a transition system where states represent prefixes. Consider $L_1 = [(c_1, \langle \blacktriangleright, a, b, c, f, \blacksquare \rangle), (c_2, \langle \blacktriangleright, a, b, f, c, \blacksquare \rangle), (c_3, \langle \blacktriangleright, d, a, b, c, g \rangle), (c_4, \langle \blacktriangleright, d, a, f, c, h \rangle)]$ as an event log where c_1
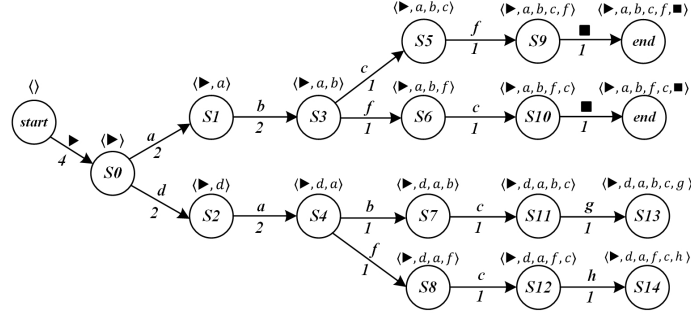
Fig. 2: The history transition system of the event log L_1. The circles represent states, and the arcs represent transitions with activity names as their labels. The numbers below arcs show the frequency of the corresponding transition.

and c_2 have complete traces, and c_3 and c_4 have partial traces. Figure 2 shows the history transition system, obtained by considering $state_{hd}()$ as the state representation function for the event log L_1. A history transition system can be converted to a probabilistic model to show the correlation between states as conditional probabilities. We utilize such representation of an event log to quantify the correlations between traces. Then, such correlations are used to quantify temporal privacy leakages of a DP mechanism in CEDP.

3.3   Differential Privacy

Differential privacy provides a formal definition of data privacy. The main idea of differential privacy is to randomize the data in such a way that an observer seeing the randomized output cannot tell if a specific individual's information was used in the computation [5]. Considering the distribution of trace variants as our sensitive event data, ϵ-DP can be defined as follows.

Definition 6 (ϵ-DP). *Let L_1 and L_2 be two neighbouring event logs that differ only in a single entry, e.g., $\tilde{L}_2 = \tilde{L}_1 \uplus [\sigma]$, for any $\sigma \in \mathcal{A}^*$, and let $\epsilon \in \mathbb{R}_{>0}$ be the privacy parameter. A randomized mechanism $\mathcal{M}:\mathcal{B}(\mathcal{A}^*) \to \mathcal{B}(\mathcal{A}^*)$ provides ϵ-DP if for any $(\sigma, f) \in \mathcal{A}^* \times \mathbb{N}_{>0}$ and for all $\tilde{L}' \in rng(\mathcal{M})$:*

$$log\frac{Pr((\sigma, f) \in \tilde{L}' \mid \mathcal{M}(\tilde{L}_1))}{Pr((\sigma, f) \in \tilde{L}' \mid \mathcal{M}(\tilde{L}_2))} \leq \epsilon$$

The parameter ϵ is called the *privacy budget* and represents the degree of privacy. The smaller the privacy budget, the stronger the privacy guarantees. A real-valued query q can be made differentially private by using a *Laplace mechanism* where the noise is drawn from a *Laplacian distribution* with scale $\Delta q/\epsilon$. Δq is called the sensitivity of the query q. Intuitively, Δq denotes the amount of uncertainty that one needs to incorporate into the output to hide the contribution of single occurrences at the ϵ-DP level. In our context, q is the frequency of a trace variant. Since one individual, i.e., a case, contributes to only one trace, the sensitivity is $\Delta q=1$ [15,11]. If an individual can appear in more
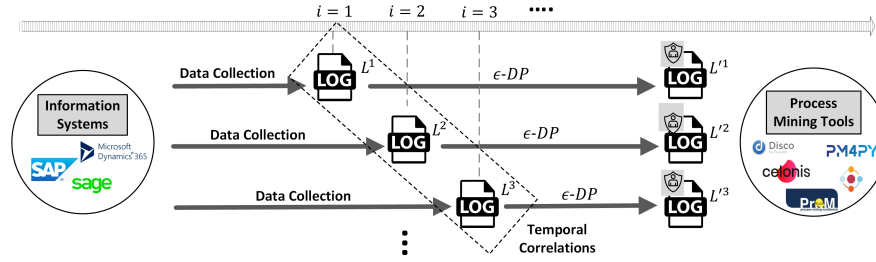
Fig. 3: The general overview of continuous event data publishing in process mining.

than one trace, the sensitivity needs to be accordingly increased assuming the same value for privacy parameter ϵ [5].

4 Continuous Event Data Publishing

Continuous data publishing can generally be classified into three main categories: *incremental*, *decremental*, and *dynamic* [13]. In incremental continuous data publishing, the raw data are cumulatively collected up to a release point, and they cannot be updated or deleted after the collection phase. In decremental continuous data publishing, the previously collected raw data can only be deleted in the later releases. Dynamic data publishing assumes that new raw data can be added to the previously collected data, and the previously collected data can be updated or deleted. In the context of process mining, the events generated by an information system are cumulatively collected, and they are not updated or deleted after generation, i.e., the continuous event data publishing is *incremental*. Figure 3 shows the general overview of continuous event data publishing using an ϵ-DP mechanism in process mining. Events recorded by information systems are collected up to a release point i, then ϵ-DP mechanisms are applied to provide privacy guarantees for each event log L^i.

The incremental nature of CEDP can be considered as the main reason of temporal correlations among event logs that need to be published at different release points. For example, the complete traces in an event log L^i appear in all the next releases $L^{i+1}, L^{i+2}, \cdots$. Moreover, each trace σ in an event log L^i has a prefix in all the previous releases $L^j, L^{j+1}, \cdots, L^{i-1}$, s.t., $j < i$ and L^j is the event log where the process of the case having the trace σ started. As these examples show, temporal correlations can be categorized into two main categories: *forward* and *backward*. Given L^i as an event log at release point i, the former considers temporal correlations between L^i and its next releases, and the latter concerns temporal correlations between L^i and its previous releases.

4.1 CEDP Scenarios

Different event data publishing scenarios can have a significant impact on the privacy leakage based on temporal correlations. In the following, we briefly ex-

plain some of the different possible scenarios. In general, CEDP scenarios can be based on a *time-window*, e.g., weekly, or a *count-window*, e.g., the number of new cases. Since time-window-based scenarios are not deterministic in terms of the amount of new data that can be published in each window, we focus on count-window-based scenarios to quantify the potential privacy degradation. One can consider different count-window-based scenarios. For example, an event log is released when there exist x new cases compared to the previous release, or when there exist x new events per trace, or when there exist up to x new events per trace, etc. We classify the count-window-based scenarios into two main types: *certain* and *uncertain*. The former specifies an exact number, e.g., x new events per trace. The latter specifies a bound, e.g., up to x events per trace. This classification allows us to assess the effects of certain and uncertain CEDP scenarios on temporal privacy leakages.

Since events are the smallest units of event logs, to propose a generic approach, we consider the following certain and uncertain scenarios: (S1) an event log is released when there exist exactly x new events per trace compared to the previous release, and (S2) an event log is released when there exist up to x new events per trace compared to the previous release. In practice, such bounds can be specified to keep the process mining findings updated. Note that in both scenarios, events can belong to a new case or an existing one. In Subsection 4.4, we demonstrate how to use transition systems to quantify the forward and backward privacy leakages considering these scenarios.

4.2   Notation Summary

For the sake of simplicity, we assume that the number of releases is λ, which does not need to be exactly specified. For a given event log L, $C_L \subset \mathcal{C}$ and $A_L \subset \mathcal{A}$ are considered as the finite set of dummy case identifiers and the set of activities that can appear in different releases of L, respectively. L^i denotes an event log that needs to be released at point $i \in [1, \lambda]$. L^i contains cases and their current states describing *full history*, i.e., traces. $\sigma_c^i \in \tilde{L}^i$ is the state of a case c at the release point i. Note that according to Definition 3, each case can only have one trace in an event log.

We consider $\mathcal{M}^i$ as the DP mechanism, which is applied to $\tilde{L}^i$ to randomize the count of trace variants. $rng(\mathcal{M}^i)$ denotes the set of all possible outputs that $\mathcal{M}^i$ can produce. For simplicity, $\mathcal{M}^i$ is considered to be the same DP mechanism, e.g., a Laplace mechanism, but maybe with different privacy budgets at each $i \in [1, \lambda]$. $\tilde{L}'^i \in rng(\mathcal{M}^i)$ denotes a differentially private output at release point i. In the following, we first demonstrate the potential privacy loss of $\mathcal{M}^i$ for a single release of event log at release point i. Then, we quantify the privacy leakage in the context of continuous releases when i varies from 1 to λ.

4.3   Privacy Leakage of a Single Release

Consider an adversary whose target is to identify the state of a case $c \in C_L$ at release point $i \in [1, \lambda]$. We assume that such an adversary has the knowledge of all the states at the given release point except the state of the target case c.

Definition 7 (Adversary without Temporal Correlations - $Ad^{L^i_c}$). *Let L^i be an event log that needs to be released at the point i and $\sigma^i_c \in \tilde{L}^i$ be the state of case c at the point i. $Ad^{L^i_c}$ denotes an adversary whose target is to identify σ^i_c. $L^i_c = \{(c', \sigma) \in L^i | c \neq c'\}$ is the background knowledge of such an adversary.*

$Ad^{L^i_c}$ observes $\tilde{L}'^i \in rng(\mathcal{M}^i)$ and tries to distinguish case c's state. The privacy leakage of the DP mechanism $\mathcal{M}^i$ can be formulated as follows, where $\sigma^i_c, \sigma'^i_c \in A^*_L$ are two different possible traces for the state of case c.

$$PL(Ad^{L^i_c}, \mathcal{M}^i) \coloneqq \sup_{\tilde{L}'^i, \sigma^i_c, \sigma'^i_c} log \frac{Pr(\tilde{L}'^i \mid \tilde{L}^i_c \uplus \sigma^i_c)}{Pr(\tilde{L}'^i \mid \tilde{L}^i_c \uplus \sigma'^i_c)} \tag{1}$$

$$PL(\mathcal{M}^i) \coloneqq \max_{c \in C_L} PL(Ad^{L^i_c}, \mathcal{M}^i) \tag{2}$$

Equation (2) is another formal representation of differential privacy that formulates the privacy budget as the supremum of privacy leakage, i.e., considering ϵ as the privacy budget, $PL(\mathcal{M}^i) = \epsilon$.

4.4 Privacy Leakage of Continuous Releases

We exploit a full-history transition system to calculate probabilities of visiting states and to generate *forward* and *backward* temporal correlations describing the probabilities for transitions between states. We obtain a transition system form the last collected event log that needs to be published.

Definition 8 (State Probability). *Let $TS_{L,state_{hd}()} = (S, A, T)$ be a history transition system based on an event log L, the probability of visiting a state $s \in S$ is as follows: $Pr(s) = |T'|/|L|$ where $T' = [(s_1, a, s_2) \in T | s_2 = s]$.*

For instance, in Fig. 2, $Pr(S3) = 2/4$. In the following, we define forward and backward temporal correlations based on scenarios S1 and S2 (see Subsection 4.1). Note that to simplify the notation, we abbreviate $\langle a_1, a_2, ..., a_n \rangle$ as $\langle \rangle_n$, and a sequence of event logs that need to be released, i.e., $L^1, L^2, ..., L^\lambda$, as $L^{1..\lambda}$.

Definition 9 (Forward Temporal Correlations - FTC). *Let $TS_{L,state_{hd}()} = (S, A, T)$ be a transition system based on an event log L. The forward temporal correlations are calculated based on the correlations between adjacent states. Given $s_1, s_2 \in S$ as two adjacent states, $Pr(s_2 = \langle \rangle_n | s_1 = \langle \rangle_{n-1}) = \frac{|T''|}{|T'|}$ where $T' = [(s, a, s') \in T | s = s_1]$ and $T'' = [(s, a, s') \in T | s = s_1 \wedge s' = s_2]$.*

- *The certain scenario with $x \in \mathbb{N}_{>0}$ new events:*
  Given $s_1 \in S \backslash S^{end}$, for all $s_2 \in S$, s.t., $s_1 \sqsubset s_2$, and $|s_2| - |s_1| = x$: $Pr(s_2 = \langle \rangle_n | s_1 = \langle \rangle_{n-x}) = \prod_{j=0}^{x-1} Pr(s'_2 = \langle \rangle_{n-j} | s'_1 = \langle \rangle_{n-(j+1)})$. Otherwise, $Pr(s_2 | s_1) = 0$. If $s_1 \in S^{end}$, $Pr(s_2 = s_1 | s_1) = 1$.
- *The uncertain scenario with up to $x \in \mathbb{N}_{>0}$ new events:*
  Given $s_1 \in S \backslash S^{end}$, let fd_{s_1} be the distance of the furthest state s_2 from s_1, s.t., $s_1 \sqsubset s_2$. $fm^x_{s_1} = min(x, fd_{s_1})$ is considered as the maximal forward move on the transition system starting from s_1. For all $s_2 \in S$, s.t., $s_1 \sqsubset s_2$, and for all $y \in [1, min(x, |s_2| - |s_1|)]$: $Pr(s_2 = \langle \rangle_n | s_1 = \langle \rangle_{n-y}) = 1/(fm^x_{s_1}+1) \times \prod_{j=0}^{y-1} Pr(s'_2 = \langle \rangle_{n-j} | s'_1 = \langle \rangle_{n-(j+1)})$, and for $y = 0$: $Pr(s_2 = \langle \rangle_n | s_1 = \langle \rangle_{n-y}) = 1/(fm^x_{s_1}+1)$. Otherwise, $Pr(s_2 | s_1) = 0$. If $s_1 \in S^{end}$, $Pr(s_2 = s_1 | s_1) = 1$.

For instance, in Fig. 2, given the certain scenario with $x{=}2$, we only consider the states that their distance from a given state is 2. If the given state is $S1$, $Pr(S5|S1){=}^1\!/_2$ and $Pr(S6|S1){=}^1\!/_2$. Other probabilities given $S1$ are considered to be zero. However, given the uncertain scenario with $x{=}2$, we explore all the states within the maximal distance 2. $fd_{S1}{=}4$ and $fm^x_{S1}{=}min(2,4)$. Thus, $Pr(S5|S1){=}^1\!/_3{\times}^1\!/_2$, $Pr(S6|S1){=}^1\!/_3{\times}^1\!/_2$, $Pr(S3|S1){=}^1\!/_3{\times}1$, and $Pr(S1|S1){=}^1\!/_3$. Other probabilities given $S1$ are considered to be zero. Note that in the uncertain scenario, we consider an equal chance for a case to stay in the same state, or move forward up to maximal x states. This is the reason for the division by $fm^x_{s_1}{+}1$.

Definition 10 (Backward Temporal Correlations - BTC). *Let $TS_{L,state_{hd}()}$ $=(S,A,T)$ be a transition system based on an event log L. The backward temporal correlations can be obtained using Bayesian inference based on FTC.*

- *The certain scenario with $x{\in}\mathbb{N}_{>0}$ new events:*
  Given $s_2{\in}S$, for all $s_1{\in}S$, s.t., $s_1{\sqsubset}s_2$, and $|s_2|{-}|s_1|{=}$ x: $Pr(s_1 = \langle\rangle_{n-x}|s_2 = \langle\rangle_n) = {}^{Pr(s_1=\langle\rangle_{n-x})\times Pr(s_2=\langle\rangle_n|s_1=\langle\rangle_{n-x})}\!/_{Pr(s_2=\langle\rangle_n)}$. Otherwise, $Pr(s_1|s_2) = 0$.
- *The uncertain scenario with up to $x{\in}\mathbb{N}_{>0}$ new events:*
  Given $s_2{\in}S$, let bd_{s_2} be the distance of the furthest state s_1 from s_2, s.t., $s_1{\sqsubset}s_2$, and let $bm^x_{s_2} = min(x,bd_{s_2})$ be the maximal backward move on the transition system starting from s_2. For all $s_1{\in}S$, s.t., $s_1{\sqsubset}s_2$, and for all $y{\in}[1,min(x,|s_2|{-}|s_1|)]$: $Pr(s_1 = \langle\rangle_{n-y}|s_2 = \langle\rangle_n) = {}^1\!/_{(bm^x_{s_2}+1)}{\times}{}^{Pr(s_1=\langle\rangle_{n-x})\times Pr(s_2=\langle\rangle_n|s_1=\langle\rangle_{n-x})}\!/_{Pr(s_2=\langle\rangle_n)}$, and for $y{=}0$: $Pr(s_1 = \langle\rangle_{n-y}|s_1 = \langle\rangle_n) = {}^1\!/_{(bm^x_{s_2}+1)}$. Otherwise, $Pr(s_1|s_2) = 0$.

For instance, in Fig. 2, given the certain scenario with $x{=}2$, $Pr(S1|S5) = \frac{^2\!/_{4}\times^1\!/_2}{^1\!/_4}$, and given the uncertain scenario with $x{=}2$, $Pr(S1|S5){=}^1\!/_3{\times}\frac{^2\!/_{4}\times^1\!/_2}{^1\!/_4}$. In the uncertain scenario, the previous state of a case can be the current state or any state within the maximal x distance, and this is the reason for the division by $bm^x_{s_2}{+}1$. Note that we incrementally update the transition system based on the last collected event log. Thus, the knowledge regarding correlations is gained based on all the available data up to the last release point.

Definition 11 (Adversary with Temporal Correlations - $Ad^{L^{1..\lambda}_c}$). *Let $L^{1..\lambda}$ be the sequence of event logs that need to be released. We denote $Ad^{L^{1..\lambda}_c}$ as an adversary who has knowledge of all case's states in the entire releases range from 1 to λ except the state of the victim case $c{\in}C_L$. The background knowledge of such an adversary is $L^{1..\lambda}_c = \bigcup_{i\in[1,\lambda]} L^i_c$ as well as the knowledge of temporal correlations. $Ad^{L^{1..\lambda}_c}_F$ ($Ad^{L^{1..\lambda}_c}_B$) denotes such an adversary with only forward (backward) temporal correlations.*

Given $c{\in}C_L$, $Ad^{L^{1..\lambda}_c}$ observes the differentially private outputs $\tilde{L}'^1, \tilde{L}'^2, \ldots, \tilde{L}'^\lambda$ of the DP mechanism $\mathcal{M}^i$ applied to $\tilde{L}^i$ at each release point $i{\in}[1,\lambda]$ and attempts to identify the state of the case c.

Definition 12 (Temporal Privacy Leakage - TPL). *Let $Ad^{L^{1..\lambda}_c}$ be an adversary with the knowledge of temporal correlations, $\mathcal{M}^i$ be a DP mechanism that is applied to each event log $\tilde{L}^i$, $i{\in}[1,\lambda]$, and $\tilde{L}'^i{\in}rng(\mathcal{M}^i)$ be the corresponding*

differentially private release at each release point. Considering $\sigma_c^i, \sigma'^i_c \in A_L^*$ as two different possible states for case $c \in C_L$, temporal privacy leakage of $\mathcal{M}^i$ w.r.t. $Ad^{L_c^{1..\lambda}}$ is defined as follows:

$$TPL(Ad^{L_c^{1..\lambda}}, \mathcal{M}^i) := \sup_{\tilde{L}'^1,\ldots,\tilde{L}'^\lambda,\sigma_c^i,\sigma'^i_c} \log \frac{Pr(\tilde{L}'^1,\ldots,\tilde{L}'^\lambda \mid \tilde{L}_c^i \uplus \sigma_c^i)}{Pr(\tilde{L}'^1,\ldots,\tilde{L}'^\lambda \mid \tilde{L}_c^i \uplus \sigma'^i_c)} \tag{3}$$

$$TPL(\mathcal{M}^i) := \max_{c \in C_L} TPL(Ad^{L_c^{1..\lambda}}, \mathcal{M}^i) \tag{4}$$

The above-defined temporal privacy leakage can be broken down into backward and forward privacy leakages, as defined in Definition 13 and Definition 14.

Definition 13 (Backward Privacy Leakage - BPL). *Backward privacy leakage of $\mathcal{M}^i$, $i \in [1, \lambda]$, w.r.t. $Ad_B^{L_c^{1..\lambda}}$ is defined as follows:*

$$BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^i) := \sup_{\tilde{L}'^1,\ldots,\tilde{L}'^i,\sigma_c^i,\sigma'^i_c} \log \frac{Pr(\tilde{L}'^1,\ldots,\tilde{L}'^i \mid \tilde{L}_c^i \uplus \sigma_c^i)}{Pr(\tilde{L}'^1,\ldots,\tilde{L}'^i \mid \tilde{L}_c^i \uplus \sigma'^i_c)} \tag{5}$$

$$BPL(\mathcal{M}^i) := \max_{c \in C_L} BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^i) \tag{6}$$

Definition 14 (Forward Privacy Leakage - FPL). *Forward privacy leakage of $\mathcal{M}^i$, $i \in [1, \lambda]$, w.r.t. $Ad_F^{L_c^{1..\lambda}}$ is defined as follows:*

$$FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^i) := \sup_{\tilde{L}'^i,\ldots,\tilde{L}'^\lambda,\sigma_c^i,\sigma'^i_c} \log \frac{Pr(\tilde{L}'^i,\ldots,\tilde{L}'^\lambda \mid \tilde{L}_c^i \uplus \sigma_c^i)}{Pr(\tilde{L}'^i,\ldots,\tilde{L}'^\lambda \mid \tilde{L}_c^i \uplus \sigma'^i_c)} \tag{7}$$

$$FPL(\mathcal{M}^i) := \max_{c \in C_L} FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^i) \tag{8}$$

From Equations (4), (6), and (8), we can conclude Equation (9), which shows that to quantify the temporal privacy leakage, we need to analyze BPL and FPL. We subtract $PL(\mathcal{M}^i)$ because it is included in both BPL and FPL.

$$TPL(\mathcal{M}^i) = BPL(\mathcal{M}^i) + FPL(\mathcal{M}^i) - PL(\mathcal{M}^i) \tag{9}$$

Equations (5) and (7) can be expanded based on Bayesian theorem to calculate backward and forward privacy leakages.

Quantifying BPL As shown in [2], using Bayesian theorem, $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^i)$ can be simplified as Eq. (10) (cf. Theorem 2 and Eq. (12) in [2]). Since CEDP is incremental, the trace of a case at release point $i-1$ cannot be longer than its trace at release point i. Thus, $A_\sigma^{\leq} = \{\sigma' \in A_L^* \mid |\sigma'| \leq |\sigma|\}$ is the domain of all possible previous steps.

$$BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^i) = \sup_{\substack{\tilde{L}'^1,\ldots,\tilde{L}'^{i-1}\\\sigma_c^i,\sigma'^i_c}} \log \frac{\sum\limits_{\sigma_c^{i-1} \in A_{\sigma_c^i}^{\leq}} Pr(\tilde{L}'^1,\ldots,\tilde{L}'^{i-1} \mid \tilde{L}_c^{i-1} \uplus \sigma_c^{i-1}) Pr(\sigma_c^{i-1}\mid\sigma_c^i)}{\sum\limits_{\sigma'^{i-1}_c \in A_{\sigma'^i_c}^{\leq}} \underbrace{Pr(\tilde{L}'^1,\ldots,\tilde{L}'^{i-1} \mid \tilde{L}_c^{i-1} \uplus \sigma'^{i-1}_c)}_{(a)} \underbrace{Pr(\sigma'^{i-1}_c\mid\sigma'^i_c)}_{(b)}}$$

$$+ \sup_{\tilde{L}'^i,\sigma_c^i,\sigma'^i_c} \log \frac{Pr(\tilde{L}'^i \mid \tilde{L}_c^i \uplus \sigma_c^i)}{\underbrace{Pr(\tilde{L}'^i \mid \tilde{L}_c^i \uplus \sigma'^i_c)}_{(c)}} \tag{10}$$

In Eq. (10), the part annotated with (a) refers to BPL at point $i-1$, (b) refers to the backward conditional probabilities for the case c given its state at release point i, and (c) is the privacy leakage of single release at point i without considering temporal correlations. Based on Eq. (10), if $i=1$, then $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^1)=PL(Ad^{L_c^1}, \mathcal{M}^1)$, and if $i>1$, $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^i)$ is as follows, where $AL_B(.)$ is a function to calculate the accumulated BPL.

$$BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^i)=AL_B(BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^{i-1})) + PL(Ad^{L_c^i}, \mathcal{M}^i) \qquad (11)$$

Equation (11) shows that BPL can be calculated recursively and may accumulate over time. According to Definition 10, and considering the certain scenario of event data publishing, the backward temporal correlation between states of a case is always on an extreme side, i.e., given σ_c^i as the state of a case $c \in C_L$ at the release point $i \in [1, \lambda]$, there exists a state for the case c at release point $i-x$, σ_c^{i-x}, s.t., $\sigma_c^{i-x} \sqsubset \sigma_c^i$, thus $Pr(\sigma_c^{i-x}|\sigma_c^i)=1$. Consequently, considering $i=2$, $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^2)$ is calculated as follows:

$$BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^2) = \sup_{\tilde{L}'^1, \sigma_c^1, \sigma_c'^1} log \frac{Pr(\tilde{L}'^1 \mid \tilde{L}_c^1 \uplus \sigma_c^1)}{Pr(\tilde{L}'^1 \mid \tilde{L}_c^1 \uplus \sigma'^1_c)} + \sup_{\tilde{L}'^2, \sigma_c^2, \sigma_c'^2} log \frac{Pr(\tilde{L}'^2 \mid \tilde{L}_c^2 \uplus \sigma_c^2)}{Pr(\tilde{L}'^2 \mid \tilde{L}_c^2 \uplus \sigma'^2_c)}$$

$$= PL(Ad^{L_c^1}, \mathcal{M}^1) + PL(Ad^{L_c^2}, \mathcal{M}^2)$$

If we consider ϵ as the privacy budget of the mechanism $\mathcal{M}^i$, i.e., for any $i \in [1, \lambda]$, $max_{c \in C_L} PL(Ad^{L_c^i}, \mathcal{M}^i)=\epsilon$. Then, $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^2)=2\epsilon$. Consequently, $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^3)=3\epsilon$, $BPL(Ad_B^{L_c^{1..\lambda}}, \mathcal{M}^4)=4\epsilon$, etc. Hence, BPL for CEDP considering the certain scenario is expected to linearly increase. We investigate this observation in our experiments.

Quantifying FPL Similar to the backward privacy leakage, the equation of the forward privacy leakage, i.e., Eq. (7), can also be simplified as Eq. (12) (cf. Theorem 2 and Eq. (14) in [2]). Since continuous event data publishing is incremental, the trace of a case at release point $i+1$ cannot be shorter than its trace at release point i. Thus, $A_\sigma^\geq = \{\sigma' \in A_L^* \mid |\sigma| \leq |\sigma'|\}$.

$$FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^i)= \sup_{\substack{\tilde{L}'^{i+1}, \ldots, \tilde{L}'^\lambda \\ \sigma_c^i, \sigma_c'^i}} log \frac{\sum\limits_{\sigma_c^{i+1} \in A_{\sigma_c^i}^\geq} Pr(\tilde{L}'^{i+1}, \ldots, \tilde{L}'^\lambda \mid \tilde{L}_c^{i+1} \uplus \sigma_c^{i+1})Pr(\sigma_c^{i+1}|\sigma_c^i)}{\sum\limits_{\sigma'^{i+1}_c \in A_{\sigma'^i_c}^\geq} \underbrace{Pr(\tilde{L}'^{i+1}, \ldots, \tilde{L}'^\lambda \mid \tilde{L}_c^{i+1} \uplus \sigma'^{i+1}_c)}_{(a)} \underbrace{Pr(\sigma'^{i+1}_c|\sigma'^i_c)}_{(b)}}$$

$$+ \sup_{\tilde{L}'^i, \sigma_c^i, \sigma_c'^i} log \frac{Pr(\tilde{L}'^i \mid \tilde{L}_c^i \uplus \sigma_c^i)}{\underbrace{Pr(\tilde{L}'^i \mid \tilde{L}_c^i \uplus \sigma'^i_c)}_{(c)}}$$

$$(12)$$

In Eq. (12), the part annotated with (a) refers to FPL at release point $i+1$, (b) refers to the forward conditional probabilities for the case c given its state at point i, and (c) is the privacy leakage of single release at point i without

considering temporal correlations. Similar to Eq. (10), in Eq. (12), if $i = 1$, then $FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^1) = PL(Ad^{L_c^1}, \mathcal{M}^1)$, and if $i > 1$, $FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^i)$ is as follows, where $AL_F(.)$ is a function to calculate the accumulated forward privacy leakage.

$$FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^i) = AL_F(FPL(Ad_F^{L_c^{1..\lambda}}, \mathcal{M}^{i+1})) + PL(Ad^{L_c^i}, \mathcal{M}^i) \qquad (13)$$

Equation (13) shows that FPL can also be recursively calculated and may accumulate over time. Since event data publishing is incremental, the complete traces remain the same in all the next releases. Thus, FPL in CEDP can be on an extreme side whenever there exist complete traces in the previous releases. Moreover, based on Eq. (12), we assume FPL in CEDP depends on the variation of traces in an event log. For instance, considering the certain scenario, if an event log only contains one trace variant, FPL can be on an extreme side because the next state of a new case is certainly known based on the previously recorded states. Hence, FPL is expected to linearly increase. We investigate the effect of the *trace uniqueness* ratio on FPL in our experiments.

Calculating Accumulative Privacy Leakage Cao et al. [2] show that the accumulative privacy leakages can be formulated as an optimization problem where the objective function is a ratio of two linear functions and the constraints are linear equations. Since we rely on transition systems to obtain temporal correlations, the knowledge of temporal correlations is bounded to the traces in the state space of the transition system. For the traces that are not included in the state space, we consider the worst-case w.r.t. the knowledge of correlations, i.e., no correlation. We assume that adapting the optimization problem from [2], is a straightforward process. Thus, we avoid including it here. Nevertheless, we provided the adapted optimization problem and a short explanation regarding the *computational complexity* of our approach as supplementary material in our GitLab repository.[3]

5   Experiments

The aims of the experiments are as follows: (1) Investigating the effect of temporal correlations among event logs on the provided privacy guarantees, (2) Exploring the effect of different CEDP scenarios on temporal correlations and privacy leakages, and (3) Exploring the impact of trace uniqueness in event logs on temporal privacy leakages. We have implemented a Python script to conduct the experiments. The source code is available on GitLab[4] and as a Python package[5] that can be installed using *pip* commands. Table 1 shows the general statistics of the real-life public event logs that we employed for our experiments. The *trace uniqueness* shows the rate of unique traces, i.e., $\#Variants/\#Traces$. These event logs cover a wide range w.r.t. the trace uniqueness.

---

[3] https://github.com/m4jidRafiei/QDP_CEDP/tree/main/supplementary
[4] https://github.com/m4jidRafiei/QDP_CEDP
[5] https://pypi.org/project/pm-cedp-qdp/

Table 1: General statistics of the event logs used in the experiments.

Event Log	#Events	#Unique Activities	#Traces	#Variants	Trace Uniqueness
Sepsis	15214	16	1050	846	80%
BPIC-2013	65533	4	7554	1511	20%
BPIC-2012-App	60849	10	13087	17	0.12%

To simulate CEDP, we need to specify the initial release and a sequence of event logs that are considered to be continuously published. Thus, we need a *split-point* that splits an event log into two parts; *initial* and *continuous*. One can partition an event log into initial and continuous parts in a variety of ways, e.g., having an initial log that contains all the cases, or having an initial log that contains $x\%$ of cases or events, and so on. We consider the percentage of events included in the initial part as a criteria for splitting an event log. We split Sepsis and BPIC-2012-App into two parts such that the initial part contains roughly 50% of events so that there is enough data to obtain reliable knowledge regarding the correlations. However, BPIC-2013 is partitioned in such a way that the initial part contains roughly 35% of events so that there exist no complete trace, yet, at the same time, there is enough data to discover a transition system and obtain the probabilities.[6] Table 2 shows general statistics of the event logs partitions after being partitioned. Note that incomplete (partial) traces are the same in both partitions.

The initial part is published as the first release. Then, each future release is generated w.r.t. the scenarios S1 and S2 (see Subsection 4.1). In both scenarios, the window size, i.e., the number of new events per trace in a future release, varies from 1 to 4. Note that to simulate scenario S2, a random integer within the window size is generated to determine the number of new events. For each scenario, we continue the publishing process for up to 5 releases or until there are no incomplete traces to publish.

Figure 4 and 5 show the privacy leakages for different releases of the event logs based on the CEDP scenarios S1 and S2, respectively. We consider $\epsilon = 0.01$ as the privacy budget of a differential privacy mechanism $\mathcal{M}$ that is applied to each release. Thus, for the first release $FPL=BPL=TPL=0.01$. Recall that $TPL=FPL+BPL-\epsilon$. Note that the implementation details of such a mechanism that does not consider correlations among different releases will not impact our experiments. In the following, we explain the results for each scenario.

Table 2: General statistics of the initial and continuous parts of event logs used in the experiments.

Event Log	Parts	#Events	#Complete Traces	#Incomplete Traces
Sepsis	Initial	7290	442	84
	Continuous	7924	524	84
BPIC-2013	Initial	21705	0	2271
	Continuous	43828	5283	2271
BPIC-2012-App	Initial	29227	5849	690
	Continuous	31622	6548	690

---

[6] Note that experiments can be extended considering different partitioning scenarios and focusing on different log characteristics.
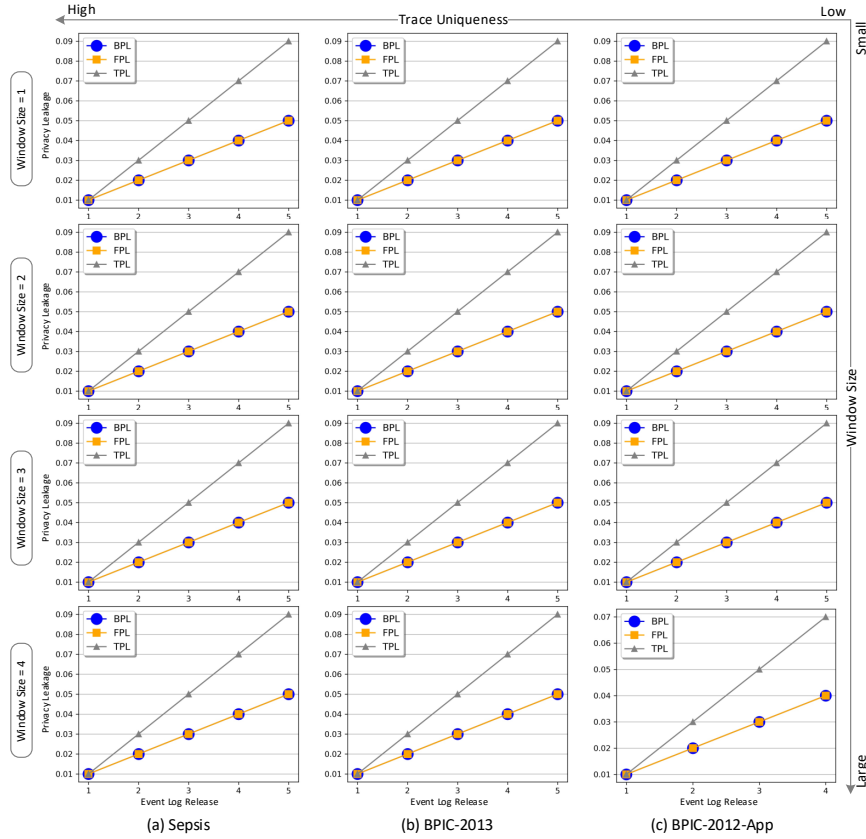
Fig. 4: FPL, BPL, and TPL for different releases of the event logs, when the CEDP scenario is S1, window size varies from 1 to 4, and $\epsilon = 0.01$. The window size indicates the number of new events per trace in a future release.

Scenario S1: The first observation is that the results are the same for all the event logs. The only different plot is at the bottom right with less number of releases because there exists no incomplete trace for BPIC-2012-App after the 4th release. Since the previous states are certain, for each state there is one state with $BTC = 1$. Thus, the correlations are strong, and BPL linearly increases for all the event logs. The same results can be seen for FPL due to different reasons. In Sepsis and BPIC-2012-App, FPL linearly increases because initial releases of these event logs contain complete traces that remain unchanged in all the next releases. Thus, there are strong correlations among those traces in all the releases. Moreover, for the most of the incomplete states (traces) there exist certain states in future releases. For example, in the second release of Sepsis, almost 78% of the incomplete states have a certain state when window size is 2. We see the same trend in BPIC-2013 although there exist no complete trace in its initial release. This is because of two reasons: (1) there are complete traces that
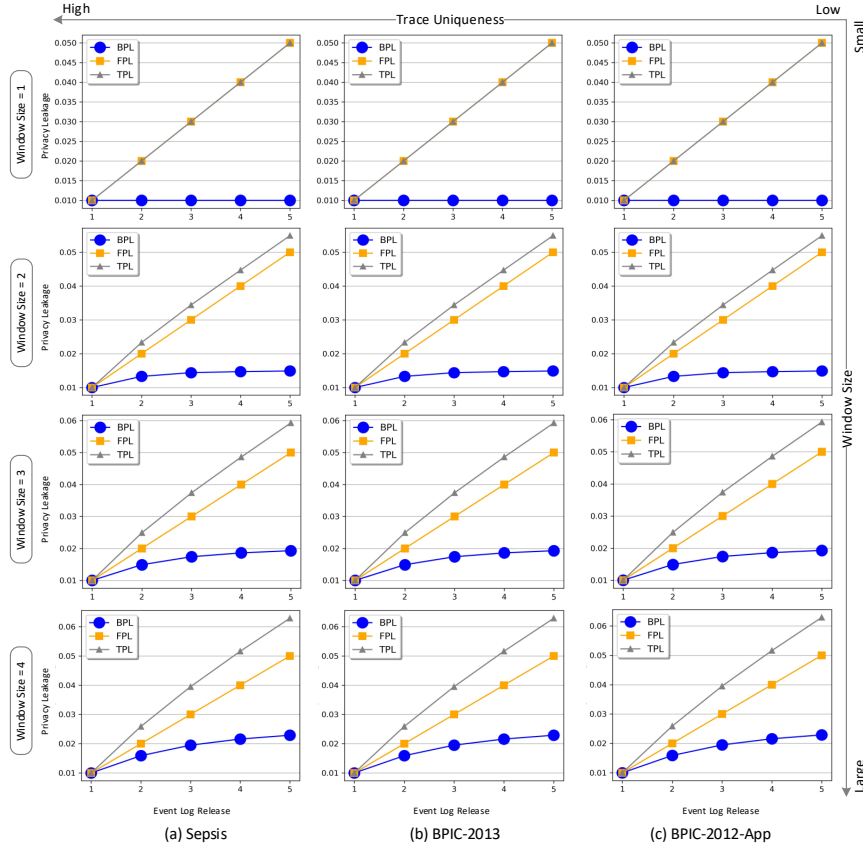
Fig. 5: FPL, BPL, and TPL for different releases of the event logs, when the CEDP scenario is S2, window size varies from 1 to 4, and $\epsilon = 0.01$. The window size indicates the maximum number of new events per trace in a future release.

appear in the second release, which is used to discover the updated transition system, and (2) BPIC-2013 contains a few distinct activities that leads to strong correlations between states. In BPIC-2013, there exist only 4 unique activities, and 86% of variants contain only two activities "Accepted" and "Queued". That leads to a situation where for many states in the corresponding transition system there exists only one possible next state that results in strong correlations.

When the window size is increased, one may expect to see lower forward correlations between states. Particularly, for the event logs with a high trace uniqueness. However, due to more complete traces that appear by increasing the window size, FPL does not decrease. We continued releasing Sepsis event logs considering 4 and 8 as window sizes until there was no more incomplete traces. According to the results, FPL never decreased.[7] Moreover, the trace uniqueness

---

[7] https://github.com/m4jidRafiei/QDP_CEDP/tree/main/more_exp

that may affect FPL does not show any impact because of the existence of strong correlations between states in all the event logs.

Scenario S2: The first observation is that all the event logs follow the same trend based on the window size. One can see a logarithmic increase for BPL based on the window size that corresponds to the so-called moderate type of correlations. That is because for the larger window sizes more states are explored on the corresponding transition system. Thus, more knowledge is gained regarding correlations. However, at the same time, more uncertainty is imposed because the previous state can be any state within the window size distance (see Definition 10). FPL still linearly increases, similar to scenario S1, which is mainly due to the complete traces leading to strong correlations. Also, the results do not change based on the trace uniqueness because of the existence of the strong correlations.

Scenario S1 vs Scenario S2: By comparing the two CEDP scenarios, one can see that scenario S1, as a certain scenario, leads to higher privacy leakages, as expected. That is because certain scenarios result in stronger correlations. This observation shows that not revealing exact data publishing scenarios in CEDP can mitigate temporal privacy leakages to some extent.

6   Conclusion and Discussion

In this paper, we quantified the privacy leakage of differential privacy mechanisms in the context of continuous event data publishing under temporal correlations. We utilized transition systems to model and quantify the correlations. We did experiments on real-life public events logs considering different CEDP scenarios. Our experiments showed that privacy leakage of a differential privacy mechanism may increase over time. In the following, we discuss some design choices, possible next steps, and limitations that need to be taken into account.

The concept of state, which is defined based on a state representation function, provides a general way to quantify correlations w.r.t. sensitive data. For instance, if one considers the set of activities in a trace as sensitive data rather than the sequence, and the corresponding differential privacy mechanism aims to protect the set of activities in traces. Then, our approach can be adapted to quantify the corresponding temporal privacy leakage by changing the state representation function, s.t., each state represents the set of activities in a trace.

The incrementally updated transition system based on the last collected event log may not be reliable for calculating forward temporal correlations if it contains only a few states. To gain more reliable knowledge regarding the correlations, one can consider a minimum number of cases reflecting a specific correlation. One can also apply more conditions, such as only considering the correlations obtained based on complete traces.

We only considered the control-flow aspect of event logs, while in reality the events recorded by information systems often contain more attributes. Each event attribute in a trace can be used to create a new correlation model or to alter an existing one. Depending on the attributes present in published event

logs, one may need to analyze the corresponding correlations to examine possible privacy leakages. Overall, this work highlights the necessity of designing differential privacy mechanisms that consider temporal correlations when event data are continuously published.

References

1. GDPR, `http://data.europa.eu/eli/reg/2016/679/oj`, Accessed: 2021-05-15
2. Cao, Y., Yoshikawa, M., Xiao, Y., Xiong, L.: Quantifying differential privacy in continuous data release under temporal correlations. IEEE Trans. Knowl. Data Eng. **31**(7), 1281–1295 (2019)
3. Chen, R., Fung, B.C., Philip, S.Y., Desai, B.C.: Correlated network data publication via differential privacy. The VLDB Journal **23**(4), 653–676 (2014)
4. Dwork, C.: Differential privacy. In: ICALP (2). Lecture Notes in Computer Science, vol. 4052, pp. 1–12. Springer (2006)
5. Dwork, C.: Differential privacy: A survey of results. In: Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008 (2008)
6. Dwork, C.: Differential privacy in new settings. In: SODA. SIAM (2010)
7. Dwork, C., Naor, M., Pitassi, T., Rothblum, G.N.: Differential privacy under continual observation. In: STOC. pp. 715–724. ACM (2010)
8. Elkoumy, G., Fahrenkrog-Petersen, S.A., Sani, M.F., Koschmider, A., Mannhardt, F., von Voigt, S.N., Rafiei, M., von Waldthausen, L.: Privacy and confidentiality in process mining: Threats and research challenges. ACM Trans. Manag. Inf. Syst. **13**(1), 11:1–11:17 (2022)
9. Elkoumy, G., Pankova, A., Dumas, M.: Mine me but don't single me out: Differentially private event logs for process mining. In: ICPM. pp. 80–87. IEEE (2021)
10. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: PRIPEL: privacy-preserving event log publishing including contextual information. In: BPM. Lecture Notes in Computer Science, vol. 12168, pp. 111–128. Springer (2020)
11. Fahrenkrog-Petersen, S.A., Kabierski, M., Rösel, F., van der Aa, H., Weidlich, M.: SaCoFa: Semantics-aware control-flow anonymization for process mining. In: ICPM. pp. 72–79. IEEE (2021)
12. Fan, L., Xiong, L., Sunderam, V.S.: FAST: differentially private real-time aggregate monitor with filtering and adaptive sampling. In: SIGMOD Conference. pp. 1065–1068. ACM (2013)
13. 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)
14. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. Proc. VLDB Endow. **7**(12), 1155–1166 (2014)
15. Mannhardt, F., Koschmider, A., Baracaldo, N., Weidlich, M., Michael, J.: Privacy-preserving process mining - differential privacy for event logs. Bus. Inf. Syst. Eng. **61**(5), 595–614 (2019)
16. Rafiei, M., van der Aalst, W.M.P.: Towards quantifying privacy in process mining. In: ICPM Workshops. Lecture Notes in Business Information Processing, vol. 406, pp. 385–397. Springer (2020)
17. Rafiei, M., van der Aalst, W.M.P.: Group-based privacy preservation techniques for process mining. Data Knowl. Eng. **134**, 101908 (2021)
18. Rafiei, M., van der Aalst, W.M.P.: Privacy-preserving continuous event data publishing. In: BPM (Forum). Lecture Notes in Business Information Processing, vol. 427, pp. 178–194. Springer (2021)