Privacy-Preserving Process Mining with PM4Py

Henrik Kirchmann[†], Stephan A. Fahrenkrog-Petersen[†], Martin Kabierski[†], Han van der Aa[♯], Matthias Weidlich[†]

[†]*Humboldt-Universität zu Berlin,* [♯]*University of Mannheim*

{henrik.kirchmann,stephan.fahrenkrog-petersen,martin.kabierski,matthias.weidlich}@hu-berlin.de
han.van.der.aa@uni-mannheim.de

***Abstract*—Process Mining allows for the data-driven analysis of business processes based on logs that contain fine-granular data from the process' execution. However, such logs can potentially be exploited to extract sensitive information about process participants. To mitigate this risk, techniques that anonymize event logs to guarantee the privacy of process participants have recently been proposed. In this paper, we report on the integration of anonymization techniques for event logs into *PM4Py*, one of the leading process mining tools. Specifically, we incorporated several state-of-the-art solutions for differential privacy-based protection. By presenting the first integration of anonymization techniques into a general process mining toolkit, we make the respective algorithms accessible to the wider community of process mining experts and data scientists.**

***Index Terms*—process mining, privacy-preserving data publishing, differential privacy, event logs**

I. Introduction

Process mining is a family of techniques to analyze the data recorded in information systems during the execution of business processes. The data is stored in so-called event logs that may include sensitive information, e.g., if they represent the clinical workflow of patients in a hospital. Privacy regulations such as the *GDPR* and the *CCPA* enforce the protection of such information [1]. Since it was shown that individuals can be re-identified within such datasets [2], [3], anonymization of event logs is needed to mitigate privacy risks.

Recently, the development of anonymization techniques for event logs gained a lot of attention [4]–[6]. Nonetheless, the adoption and uptake of these techniques has been limited. One reason being the lack of an easy-to-use integration of anonymization techniques into existing process mining toolkits [7], [8]. Specifically, many of the techniques for privacy-preserving process mining have been published in stand-alone tools [9]–[11], and they have, so far, not been accessible as part of the toolkits commonly used to realize process mining projects.

In this demo, we address this gap with the first integration of anonymization techniques for event logs in a leading process mining toolkit, i.e., *PM4Py* [7]. Particularly, we incorporate techniques that protect event logs with differential privacy, which is considered the state-of-the-art privacy guarantee, as also adopted by *SAP* [12], and the *US Census Bureau* [13].

Below, we first review the features that have been added to the PM4Py library in Section II. Then, we provide information on the usage of our tool and its maturity (Section III), before we conclude (Section IV).

II. Feature Overview

We chose to integrate our anonymization techniques into *PM4Py* [7] due to the rich ecosystem provided by the toolkit. This includes, for instance, the ability to handle event logs in different file formats, such as *IEEE XES* and *CSV* files.

Our tool facilitates two anonymization steps: Control-flow anonymization and the anonymization of contextual information. While the control-flow anonymization can be performed independently, the anonymization of contextual information requires the control-flow anonymization as a first step. In any case, we protect the privatized data with differential privacy through the insertion of noise into the event logs.

A. Control-Flow Anonymization

Our tool offers control-flow anonymization through different algorithms that implement so-called *trace variant queries*, such as the Laplacian mechanism [14] and *SaCoFa* [15]. Both algorithms insert noise into a trace-variant count, through the step-wise construction of a prefix tree.

Given an event log, the algorithms are configured with the following parameters:

- $\boldsymbol{\epsilon}$: The strength of the differential privacy guarantee. The smaller the value of ϵ, the stronger the privacy guarantee that is provided.
- $\boldsymbol{k}$: The maximal length of considered traces in the prefix tree. We note that this parameter governs the runtime complexity of both algorithms, which is $\mathcal{O}(|A|^k)$ with A being the set of activities for which events have been recorded in the log. We recommend setting k, so that roughly 80% of all traces from the original event log are covered. However, setting k to the same length as the maximum prefix-length in the original log might lead to an overfitting towards long traces.
- $\boldsymbol{p}$: The pruning parameter, which denotes the minimum count a prefix has to have in order to not be discarded. The k dependent exponential runtime of the algorithms is mitigated by the pruning parameter.

B. Anonymization of Contextual Information

In many application scenarios, an analyst might not only study control-flow information, but also incorporate contextual information, such as timestamps and resources. If that is the case, a solution that solely anonymizes the control-flow is not

```python
import pm4py
from pm4py.algo.anonymization.trace_variant_query import algorithm as trace_variant_query
from pm4py.algo.anonymization.pripel import algorithm as pripel

log = pm4py.read_xes("logName.xes")
epsilon = 0.01

sacofa_result = trace_variant_query.apply(log=log, variant=trace_variant_query.Variants.SACOFA,
    parameters={"epsilon": epsilon, "k": 15, "p": 20})

anonymized_log = pripel.apply(log=log, trace_variant_query=sacofa_result, epsilon=epsilon)
```

Algorithm 1: An example how to anonymize a given log with a *SaCoFa*-based trace variant query and *PRIPEL*

sufficient. Our tool allows to handle these scenarios by the application of *PRIPEL* [16], an algorithm that enriches a control-flow anonymized event log with contextual information, still achieving differential privacy. In our tool, *PRIPEL* can be combined with both aforementioned control-flow anonymization techniques. For this reason, the implementation of *PRIPEL* requires the original event log and the corresponding result of the control-flow anonymization as input. The approach is fine-tuned by setting the following parameters:

- ϵ: The strength of the differential privacy guarantee. The ϵ value for *PRIPEL* and the ϵ value for the adopted control-flow anonymization should be the same.
- **Blocklist**: Some event logs contain attributes that are equivalent to a case ID. For privacy reasons, such attributes must be deleted from the anonymized log. We handle such attributes with this list. As an example, in a hospital, the case ID could be based on a patient visit. However, the patient ID could be equivalently serving as a case ID and should therefore be omitted.

III. Availability and Usage of the Tool

Our tool is publicly available on GitHub[1]. Algorithm 1 illustrates the application of it to anonymize an event log. First, the listing shows how a trace variant query is applied to anonymize the control flow of the event log. Our tool adopts a factory design pattern, which enables later extensions with novel types of trace variant queries. Afterwards, *PRIPEL* is also executed to anonymize the log's contextual information. We showcase the usage in more detail in a screencast[2].

Turning to the maturity of the tool, we note that it is based on algorithms that have been published in peer-reviewed venues. Moreover, we are currently in the process of publishing our tool as part of the official release of PM4Py.

IV. Conclusion

In this paper, we presented an enhancement for a leading process mining toolkit, *PM4Py*, which enables the anonymization of event logs. As such, we make the state of the art in privacy-preserving process mining more accessible for researchers and practitioners. In future work, we want to keep expanding the list of algorithms covered by our tool.

[1]https://github.com/samadeusfp/pm4py-core-anonymization/tree/Demo-Track

[2]https://youtu.be/BRLMG_Bvdbs

References

[1] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. F. Sani, A. Koschmider, F. Mannhardt, S. N. Von Voigt, M. Rafiei, and L. V. Waldthausen, "Privacy and confidentiality in process mining: threats and research challenges," *ACM TMIS*, vol. 13, no. 1, pp. 1–17, 2021.

[2] S. Nuñez von Voigt, S. A. Fahrenkrog-Petersen, D. Janssen, A. Koschmider, F. Tschorsch, F. Mannhardt, O. Landsiedel, and M. Weidlich, "Quantifying the re-identification risk of event logs for process mining," in *International Conference on Advanced Information Systems Engineering*. Springer, 2020, pp. 252–267.

[3] K. Maatouk and F. Mannhardt, "Quantifying the re-identification risk in published process models," in *ICPM Workshops*. Springer, 2021, pp. 382–394.

[4] G. Elkoumy, A. Pankova, and M. Dumas, "Mine me but don't single me out: Differentially private event logs for process mining," in *3rd International Conference on Process Mining, ICPM 2021, Eindhoven, The Netherlands, October 31 - Nov. 4, 2021*, C. D. Ciccio, C. D. Francescomarino, and P. Soffer, Eds. IEEE, 2021, pp. 80–87.

[5] M. Rafiei and W. M. P. van der Aalst, "Group-based privacy preservation techniques for process mining," *Data Knowl. Eng.*, vol. 134, p. 101908, 2021.

[6] E. Batista and A. Solanas, "A uniformization-based approach to preserve individuals' privacy during process mining analyses," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 3, pp. 1500–1519, 2021.

[7] Alessandro Berti, Sebastiaan J. van Zelst, and Wil M. P. van der Aalst, "Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science," *CoRR*, vol. abs/1905.06169, 2019.

[8] G. Janssenswillen, B. Depaire, M. Swennen, M. Jans, and K. Vanhoof, "bupar: Enabling reproducible business process analysis," *Knowledge-Based Systems*, vol. 163, pp. 927–930, 2019.

[9] Martin Bauer, Stephan A. Fahrenkrog-Petersen, Agnes Koschmider, Felix Mannhardt, Han van der Aa, and Matthias Weidlich, "ELPaaS: Event Log Privacy as a Service," in *BPM Demos 2019*, ser. CEUR Workshop Proceedings, vol. 2420. CEUR-WS.org, 2019, pp. 159–163.

[10] M. Rafiei, A. Schnitzler, and W. M. P. van der Aalst, "PC4PM: A Tool for Privacy/Confidentiality Preservation in Process Mining," 2021.

[11] Gamal Elkoumy, Stephan A. Fahrenkrog-Petersen, Marlon Dumas, Peeter Laud, Alisa Pankova, and Matthias Weidlich, "Shareprom: A Tool for Privacy-Preserving Inter-Organizational Process Mining," in *BPM Demo 2020*, ser. CEUR Workshop Proceedings, vol. 2673. CEUR-WS.org, 2020, pp. 72–76.

[12] S. Kessler, J. Hoff, and J.-C. Freytag, "Sap hana goes private: from privacy research to privacy aware enterprise analytics," *Proceedings of the VLDB Endowment*, vol. 12, no. 12, pp. 1998–2009, 2019.

[13] J. M. Abowd, "The us census bureau adopts differential privacy," in *KDD*, 2018, pp. 2867–2867.

[14] F. Mannhardt, A. Koschmider, N. Baracaldo, M. Weidlich, and J. Michael, "Privacy-Preserving Process Mining," *Business & Information Systems Engineering*, vol. 61, no. 5, pp. 595–614, 2019.

[15] S. A. Fahrenkog-Petersen, M. Kabierski, F. Rosel, H. van der Aa, and M. Weidlich, "SaCoFa: Semantics-aware Control-flow Anonymization for Process Mining," in *ICPM 2021*, pp. 72–79.

[16] S. A. Fahrenkrog-Petersen, H. van der Aa, and M. Weidlich, "PRIPEL: Privacy-Preserving Event Log Publishing Including Contextual Information," *BPM 2020*.