Amun: A tool for Differentially Private Release of Event Logs for Process Mining (Extended Abstract)

Gamal Elkoumy[1,*], Alisa Pankova[2] and Marlon Dumas[1]

[1]University of Tartu, 18 Narva mnt, Tartu, 51009, Estonia
[2]Cybernetica, 20 Narva mnt, Tartu, 51009, Estonia

Abstract

Event logs capture the execution of business processes inside organizations. Event logs may contain private information about individuals, such as customers in customer-facing business processes, which can be a roadblock to analyzing the logs due to data regulations. To circumvent that, this paper introduces Amun: A web-based application for releasing event logs using differential privacy. The tool enables the users to get a differentially private event log that minimizes the risk to the maximum acceptable threshold given by the user. Therefore, the customer's privacy is guaranteed, and the organization could release their logs to be analyzed.

Keywords

Process Mining, Event Log, Differential Privacy

Process mining is a family of techniques that analyze the performance, quality, and conformance of business processes inside organizations [1]. The input to most process mining techniques is an event log that captures an organization's process execution. An event log may contain sensitive information about the customers being served in a customer-facing business process. Thus, organizations find the analysis of such logs subject to data privacy regulations such as GDPR[1].

Privacy-preserving process mining [2] stands to ensure that privacy regulations are met by regulation-compliant guarantees, such as k-anonymity and differential privacy [3]. Some tools enable the user to apply k-anonymity mechanisms to the event logs such as ELPaaS [4] and PC4PM [5]. Other tools enable privacy-preserving process mining across distributed event logs [6].

Among privacy-enhancing technologies, differential privacy stands out due to its proven privacy guarantees and composability. Several approaches in literature have addressed the problem of releasing differentially-private event logs for process mining [2]. However, most of these approaches have stayed in academia and have not been widely adopted in real-world scenarios where organizations need to release their event logs for process mining analysts to find enhancement opportunities.

This paper presents Amun, an open-source differentially private event log-releasing tool. The tool anonymizes the user traces in the log so that an individual cannot be singled out using

ICPM 2022 Doctoral Consortium and Tool Demonstration Track, October 23–28, 2022, Bolzano, Italy

*Corresponding author.

gamal.elkoumy@ut.ee (G. Elkoumy); alisa.pankova@cyber.ee (A. Pankova); marlon.dumas@ut.ee (M. Dumas)

[1] http://data.europa.eu/eli/reg/2016/679/oj

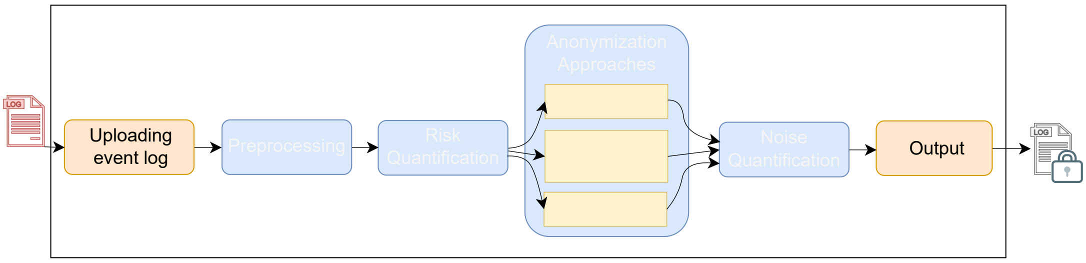


Figure 1: Overview of Amun

a sub-trace. Furthermore, Amun anonymizes the execution timestamps and masks the case IDs. As a nonfunctional requirement, Amun can process large event logs with hundreds of thousands of events. Moreover, the tool lets the users know each event's re-identification risk in the original log.

The rest of the paper is structured as follows. Sect. 1 describes Amun's functionality and components. Sect. 2 discusses the availability and maturity of the tool. Sect. 3 presents the conclusions.

1. Functionality

Figure 1 gives an overview of Amun's components. Below, we summarize the functionality of each component of Amun. Amun's detailed explanation and evaluation are presented in [7] and [8].

Input Figure 2 presents the upload page of the web application. The event log publisher uploads their event log to Amun as either an XES (eXtensible Event Stream) or CSV (Comma Separated Value) file. Amun requires the event log to have at least a column representing the case ID, a column representing the activity instance, and a column that records the timestamp executing each activity. Then, the user sets the maximum acceptable risk probability (δ) using the slider, selects the anonymization method (sampling, oversampling, or filtering), and clicks Anonymize.

The maximum acceptable risk probability (δ) represents the increase in the probability of singling out an individual after releasing the log. For example, suppose the attacker has prior information about an individual that makes the presence probability of that individual 20%. In that case, δ is the increase of that presence probability after releasing the log.

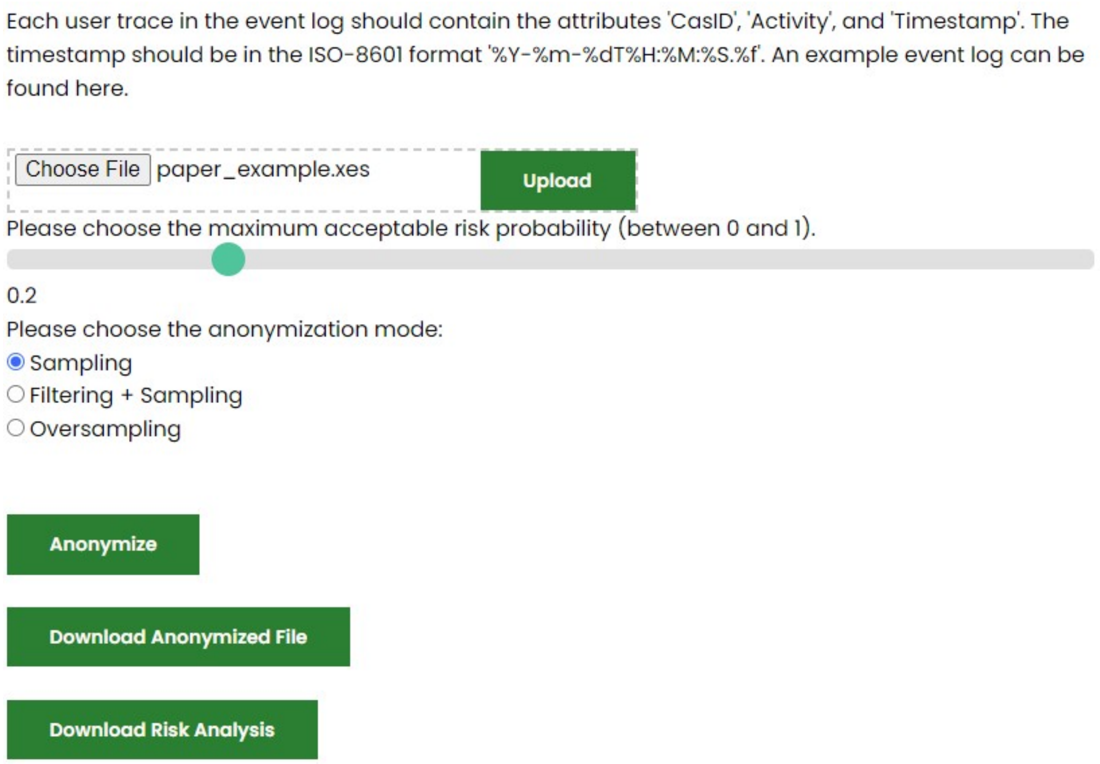
Preprocessing and risk quantification Once the user clicks Anonymize, Amun starts processing the file. The first step is to establish a representation that helps to quantify the re-identification risk attached to releasing each event in the log. To this end, Amun represents the input event log as a lossless representation, namely a Deterministic Acyclic Finite State Automata (DAFSA) [9]. Next, Amun annotates each event log with its DAFSA transition, as explained in [8]. Then, for each event, Amun estimates the prior knowledge P_k, which represents the re-identification risk before publishing the log, and the posterior knowledge P'_k, which means the re-identification risk after publishing the log. A detailed explanation of this risk quantification is presented in [8].

Anonymization Methods Amun offers the user three different anonymization approaches. All the approaches guarantee that the customers in the anonymized log will not be singled out using a subset of their trace variants or the timestamp of executing their activities. All the approaches provide differential privacy guarantees [3] by injecting noise, quantified by the differential privacy parameter ϵ, from the control flow perspective, representing user traces in the log and the timestamp perspective. Amun offers the following anonymization approaches:

- **Oversampling** [7]. In some settings, the user requires to have the same set of trace variants in the anonymized event log as in the original log. Therefore, the oversampling approach preserves the same set of trace variants while preventing singling out traces in the log. To this aim, Amun applies the approach presented by Elkoumy et al. [7]. This approach fits structured event logs where the cases of the log share trace variants.
- **Sampling**. In some settings, the user may accept the deletion of some trace variants in order to release an anonymized event log that is close to the original log. To this end, the sampling approach anonymizes the event log so that the anonymization does not add new trace variants in the log, and the difference between the real and the anonymized timestamp is minimal. Amun applies the sampling approach presented in [8]. This approach works with semi-structured event logs.
- **Filtering with Sampling**. Some event logs may contain very unique user traces, resulting in large noise injection to achieve differential privacy guarantees. Therefore, Amun applies the filtering with sampling approach presented by Elkoumy et al. [8] to enable the anonymization of unstructured event logs, i.e., event logs with unique traces. The filtering approach filters out very risky traces that requires large noise injection. Thus, the anonymized logs preserve more utility.

Noise Quantification and Injection At this step, given the estimated re-identification risk per event, Amun estimates the suitable ϵ value. We draw noise from Laplacian distribution and inject noise for both the control flow and time perspectives. This step is performed for each event independently.

Output Once the event log anonymization is finished, the anonymized event log will be available for download. Amun downloads the anonymized log in the same format as the original log. Amun offers to download the risk quantification of each activity instance in the log as a CSV file. The risk quantification per each activity instance is a column called original risk, which represents the re-identification risk of releasing the event log before the anonymization. Amun anonymizes only the three columns: case ID, activity label, and timestamp. Amun drops the other attributes from the anonymized log.



Each user trace in the event log should contain the attributes 'CasID', 'Activity', and 'Timestamp'. The timestamp should be in the ISO-8601 format '%Y-%m-%dT%H:%M:%S.%f'. An example event log can be found here.

paper_example.xes

Please choose the maximum acceptable risk probability (between 0 and 1).

0.2

Please choose the anonymization mode:

● Sampling
○ Filtering + Sampling
○ Oversampling

Figure 2: Upload an event log and anonymize it using a selected approach

2. Maturity and Availability

Amun has been empirically evaluated with real-life event logs as reported in [7, 8]. The empirical evaluation shows that Amun overcomes the state-of-the-art in terms of Jaccard distance and earth movers' distance. Also, the empirical evaluation validates the non-functional requirements, as presented in Sect. 2.

Amun is developed as a React web application and an API for ease of use. To enable quick trials by the users, Amun is available as a cloud service that can be found at http://a-mun.cloud.ut.ee. The current server deployment accepts event logs with sizes up to 5 MB. Amun is available as a docker image. The image and its installation steps can be found at https://github.com/Elkoumy/amun/tree/amun-flask-app. Also, Amun is available as a python package and can be integrated into other process mining tools. The source code and the installation steps can be found at https://github.com/Elkoumy/amun. A screencast that describes the tool is available on YouTube at https://youtu.be/1dxaCNE9WHk.

3. Conclusion

In this paper, we introduced Amun, a tool that provides differential privacy guarantees to release event logs for process mining. Amun offers approaches for event logs anonymization, which are suitable for different requirements of event logs publishers. The tool also quantifies the re-identification risk of releasing every activity instance in the log.

Acknowledgments

Work funded by European Research Council (PIX project) and by EU H2020-SU-ICT-03-2018 Project No.830929 CyberSec4Europe.

References

[1] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers, et al., Fundamentals of business process management, volume 1, Springer, 2013.

[2] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. F. Sani, A. Koschmider, F. Mannhardt, S. N. von Voigt, M. Rafiei, L. von Waldthausen, Privacy and confidentiality in process mining: Threats and research challenges, ACM Trans. Manag. Inf. Syst. 13 (2022) 11:1–11:17.

[3] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy., Found. Trends Theor. Comput. Sci. 9 (2014) 211–407.

[4] M. Bauer, S. A. Fahrenkrog-Petersen, A. Koschmider, F. Mannhardt, H. van der Aa, M. Weidlich, ELPaaS: Event log privacy as a service, in: BPM (PhD/Demos), volume 2420 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 159–163.

[5] M. Rafiei, A. Schnitzler, W. M. P. van der Aalst, PC4PM: A tool for privacy/confidentiality preservation in process mining, in: BPM (PhD/Demos), volume 2973 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 106–110.

[6] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. Dumas, P. Laud, A. Pankova, M. Weidlich, Shareprom: A tool for privacy-preserving inter-organizational process mining, in: BPM (PhD/Demos), volume 2673 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 72–76.

[7] G. Elkoumy, A. Pankova, M. Dumas, Mine me but don't single me out: Differentially private event logs for process mining, in: ICPM, IEEE, 2021, pp. 80–87.

[8] G. Elkoumy, A. Pankova, M. Dumas, Differentially private release of event logs for process mining, CoRR abs/2201.03010 (2022).

[9] J. Daciuk, S. Mihov, B. W. Watson, R. E. Watson, Incremental construction of minimal acyclic finite-state automata, Comput. Linguistics 26 (2000) 3–16.