Amun: A tool for Differentially Private Release of Event Logs for Process Mining

Gamal Elkoumy
University of Tartu, Tartu, Estonia
gamal.elkoumy@ut.ee

Alisa Pankova
Cybernetica, Tartu, Estonia
alisa.pankova@cyber.ee

Marlon Dumas
University of Tartu, Tartu, Estonia
marlon.dumas@ut.ee

***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.**

***Index Terms*—Process Mining, Event Log, Differential Privacy**

I. Introduction

Process mining is a family of techniques that analyze the performance, quality, and conformance of business processes inside organizations [2]. 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 [4] stands to ensure that privacy regulations are met by regulation-compliant guarantees, such as differential privacy [3]. Several approaches in literature have addressed the problem of releasing event logs for process mining [4]. 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 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. II describes Amun's functionality and components. Sect. III discusses the availability and maturity of the tool. Sect. IV presents the conclusions.

II. 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 [5] and [6].

a) 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.

b) 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) [1]. Next, Amun annotates each event log with its DAFSA transition, as explained in [6]. 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 [6].

c) 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. The first approach is oversampling [5]. Oversampling requires that the set of trace variants in the input log and the anonymized log are the same. To this aim, Amun applies the approach presented by Elkoumy et al. [5]. The second approach is Sampling. The sampling approach

Work funded by the European Research Council (PIX project)

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

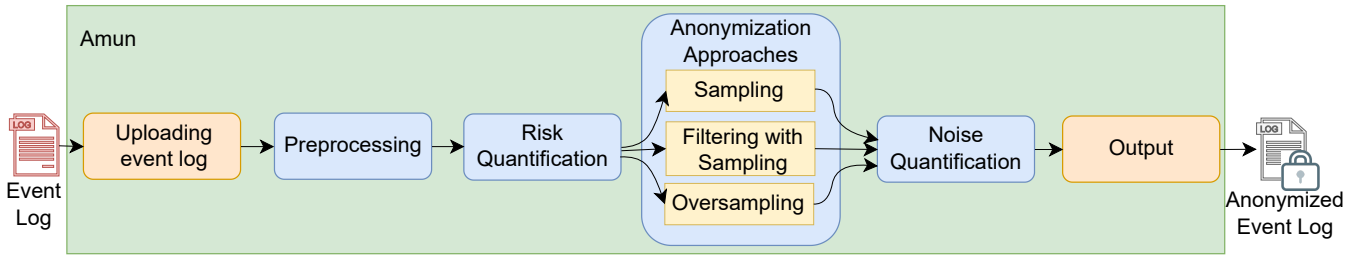

Fig. 1. Overview of Amun

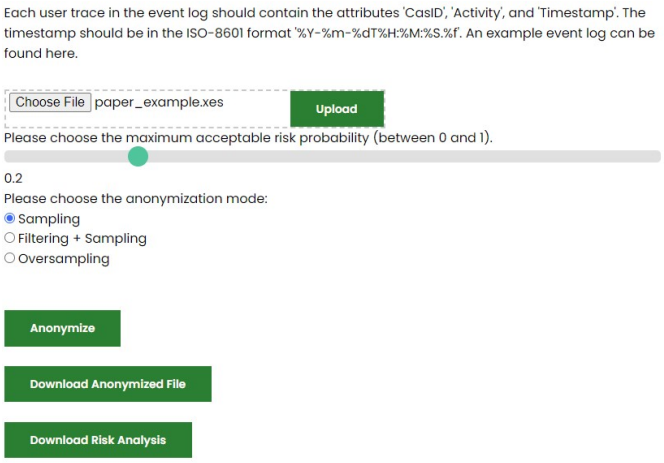

Fig. 2. Upload an event log and anonymize it using a selected approach

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 [6]. Some event logs may contain very unique user traces, resulting in large noise injection to achieve differential privacy guarantees. Therefore, Amun applies the filtering approach presented by Elkoumy et al. [6].

d) 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.

e) 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.

III. Maturity and Availability

Amun has been empirically evaluated with real-life event logs as reported in [5], [6]. 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. I.

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://amun.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.

IV. 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.

References

[1] Daciuk, J., Mihov, S., Watson, B.W., Watson, R.E.: Incremental construction of minimal acyclic finite-state automata. Comput. Linguistics **26**(1), 3–16 (2000)
[2] Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A., et al.: Fundamentals of business process management, vol. 1. Springer (2013)
[3] Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3-4), 211–407 (2014)
[4] 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)
[5] 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)
[6] Elkoumy, G., Pankova, A., Dumas, M.: Differentially private release of event logs for process mining. CoRR **abs/2201.03010** (2022)