Preserving Data Secrecy in Inter-organizational Process Mining

Valerio Goretti[0000−0001−9714−4278], Davide Basile[0000−1111−2222−3333], Luca Barbaro[0000−0002−2975−5330], and Claudio Di Ciccio[0000−0001−5570−0475]

Sapienza University of Rome, Italy
name.surname@uniroma1.it

Abstract. Inter-organizational business processes involve multiple independent organizations collaborating to achieve mutual interests. Process mining techniques have the potential to allow these organizations to enhance operational efficiency, improve performance, and deepen the understanding of their business based on the recorded process event data. However, inter-organizational process mining faces substantial challenges, including topical secrecy concerns: The involved organizations may not be willing to expose their own data to run mining algorithms jointly with their counterparts or third parties. In this paper, we introduce a novel framework, entitled CONFINE, that unlocks process mining on multiple actors' process event data while safeguarding the secrecy and integrity of the original records in an inter-organizational business setting. To ensure that the phases of the presented interaction protocol are secure and that the processed information is hidden from involved and external actors alike, our approach resorts to a decentralized architecture comprised of trusted applications running in Trusted Execution Environments (TEEs). We show the feasibility of our solution by showcasing its application to a healthcare scenario and evaluating our implementation in terms of convergence and memory usage.

Keywords: Collaborative Business Processes · Process Mining · TEEs

1 Introduction

In today's business landscape, organizations constantly seek ways to enhance operational efficiency, increase performance, and gain valuable insights to improve their processes. Process mining offers techniques to discover, monitor, and improve business processes by extracting knowledge from chronological records known as *event logs* [12]. Organizations record in these ledgers events referring to activities and interactions occurring within a business process. The vast majority of process mining contributions consider *intra-organizational* settings, in which business processes are executed inside individual organizations. However, organizations increasingly recognize the value of collaboration and synergy in achieving operational excellence. *Inter-organizational* business processes involve several independent organizations cooperating to achieve a shared objective [1]. Despite

the advantages of transparency, performance optimization, and benchmarking that companies can gain from such practices, inter-organizational process mining raises challenges that make it still hardly applicable. The major issue concerns confidentiality. Companies are reluctant to outsource to their partners inside information that is required to execute process mining algorithms. Indeed, the sharing of sensitive operational data across organizational boundaries introduces concerns about data privacy, security, and compliance with regulations. *Trusted Execution Environments* (TEEs) can serve as fundamental enablers to balance the need for insights with the imperative to protect sensitive information in inter-organizational settings. TEEs offer secure contexts that guarantee code integrity and data confidentiality in external devices. *Trusted applications* are tamper-proof software objects running in these environments.

In this paper, we propose the CONFINE framework for inter-organizational process mining. It resorts to trusted applications to preserve the secrecy and integrity of shared data. To pursue this aim, we design a decentralized software architecture for a four-staged protocol: (i) The initial exchange of preliminary metadata, (ii) The attestation of the miner entity, (iii) the secure transmission of encrypted data amid multiple parties, (iiii) The privacy-preserving merge of the shared information segments followed by the isolated and verifiable computation of process discovery algorithms on joined data. We evaluate our proof-of-concept implementation against synthetic and real-world-based data with a convergence test and memory effectiveness assessment.

The remainder of the paper is structured as follows: Sect. 2 provides an overview of related work inherent to the theme of inter-organizational process mining. In Sect. 3, we introduce a use case example that considers a healthcare scenario. The CONFINE architecture is presented in Sect. 4. Following on from this, we instantiate the addressed design principles in Sect. 5, focusing on the employed technologies, communication protocol, and implementation. In Sect. 6, we discuss our solution. Finally, we conclude and present directions for future work in Sect. 7.

2 Related Work

While inter-organizational process mining remains a consistent challenge, the academic literature has introduced a limited set of solutions. In the subsequent section, we enumerate these contributions, highlighting both their commonalities and distinctions in comparison to our work. The work of Müller et al. [11] pays attention to data privacy and security within third-party systems that mine data generated from external providers on demand. To safeguard the integrity of data earmarked for mining purposes, their research introduces a conceptual architecture that entails the execution of process mining algorithms within a cloud service environment, fortified with trusted execution environments. Drawing inspiration from this foundational contribution, our research work endeavors to design a decentralized approach characterized by organizational autonomy in the execution of process mining algorithms, devoid of synchronization mechanisms

involvement taking place between the involved parties. A notable departure from the Müller et al. framework lies in the fact that, in our architectural design, each participating organization retains the discretion to choose when and how mining operations are conducted. Moreover, we bypass the idea of fixed roles, engineering a peer-to-peer scenario in which organizations can simultaneously be data provisioners or miners. Elkoumy et al. [8,7] present a framework called Shareprom, which, like our work, offers a means for independent entities to execute process mining algorithms in inter-organizational settings while safeguarding their proprietary input data from exposure to external parties operating within the same context. Shareprom's functionality is confined to the execution of operations involving event log abstractions [2] represented as directed acyclic graphs, which the parties employ as intermediate pre-elaboration to be fed into secure multiparty computation (SMPC) [5] sessions. In contrast to our approach, where the exchanged data consists of encrypted source logs, the reliance of Shareprom on this specific graph representation imposes constraints that may prove limiting in various process mining scenarios, as stated by the authors. Given that process mining encompasses a wide array of data types and representations, we acknowledge the potential need for alternative data structures in diverse process mining contexts. Moreover, SMPC-based solutions require computationally intensive operations and synchronous cooperation among multiple parties, which make these protocols challenging to manage as the number of participants scales up [13]. In our research work, the secure computation is contained within single elaborators and does not require constant communication with external parties once the input data is exchanged. In the course of our research endeavor, we are confronted with the imperative task of integrating event logs originating from different data sources and constructing coherent traces that describe collaborative process instances. Consequently, we engage in a comprehensive examination of various methodologies delineated within the literature, each of which offers insights into the merging of event logs within inter-organizational settings. Among the array of potential solutions in this domain, the work of Claes et al. [4] holds particular significance for our research efforts. This seminal study introduces a two-step mechanism operating at the structured data level, contingent upon the configuration and subsequent application of merging rules. Each such rule delineates the criteria, namely the relations between attributes of the traces and/or the activities, that two distinct traces must satisfy in order to be combined. In contrast, the research by Hernandez et al. [9] posits a methodology functioning at the raw data level. This approach represents traces and activities as *bags-of-words* vectors, subject to cosine similarity measurements to discern links and relationships between the traces earmarked for combination. An appealing aspect of this approach lies in its capacity to generalize the challenge of merging without necessitating a priori knowledge of the underlying semantics inherent to the logs under consideration. However, we have diverged from adopting this particular approach due to considerations inherent to computational overhead. This substantial computational load carries the potential to impact both the scalability and performance of our solution.

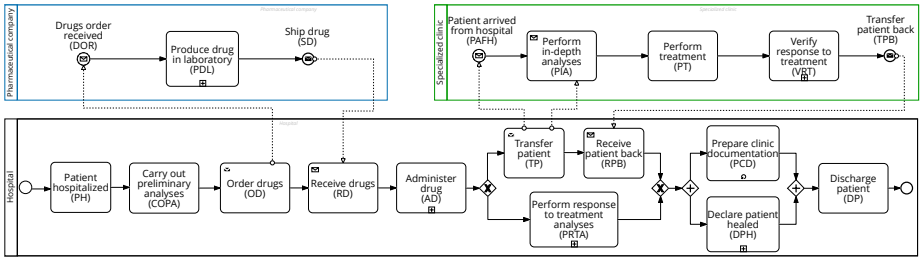


Fig. 1: A BPMN collaboration diagram of the healthcare scenario.

Table 1: Cases 312 and 711 recorded in the event logs of the Hospital, the Specialized clinic, and the Pharmaceutical company.

Hospital					
Case	Timestamp	Activity	Case	Timestamp	Activity
312	2022-07-14T10:36	PH	312	2022-07-15T22:06	TP
312	2022-07-14T16:36	COPA	711	2022-07-16T00:55	PRTA
711	2022-07-14T17:21	PH	711	2022-07-16T00:55	PCD
312	2022-07-14T17:36	OD	711	2022-07-16T02:55	DPH
711	2022-07-14T23:21	COPA	711	2022-07-16T04:55	DP
711	2022-07-15T00:21	OD	312	2022-07-16T07:06	RPB
711	2022-07-15T18:55	RD	312	2022-07-16T09:06	DPH
312	2022-07-15T19:06	RD	312	2022-07-16T09:06	PCD
711	2022-07-15T20:55	AD	312	2022-07-16T11:06	DP
312	2022-07-15T21:06	AD			

Pharmaceutical company		
Case	Timestamp	Activity
312	2022-07-15T09:06	DOR
711	2022-07-15T09:30	DOR
312	2022-07-15T11:06	PDL
711	2022-07-15T11:30	PDL
312	2022-07-15T13:06	SD
711	2022-07-15T13:30	SD

Specialized clinic		
Case	Timestamp	Activity
312	2022-07-16T00:06	PAFH
312	2022-07-16T01:06	PIA
312	2022-07-16T03:06	PT
312	2022-07-16T04:06	VRT
312	2022-07-16T05:06	TPB

T_{312} = ⟨ PH, COPA, OD, DOR, PDL, SD, RD, AD, TP, PAFH, PIA, PT, VRT, TPB, RPB, DPH, PCD, DP ⟩
T_{711} = ⟨ PH, COPA, OD, DOR, PDL, SD, RD, AD, TP, PAFH, DPH, PCD, DP ⟩

3 Motivating Scenario

For our motivating scenario, we focus on a simplified hospitalization process for the treatment of rare diseases that involves the cooperation of three parties: the Hospital, the Pharmaceutical organization, and the Specialized clinic. The process scheme is depicted in the BPMN diagram shown in Fig. 1. For the sake of simplicity, we describe the process through two cases. Alice's journey (case 312) begins when she enters the hospital for the preliminary examinations (the *patient hospitalized* event, PH). The Hospital then places to the Pharmaceutical company an order for the drugs (OD) needed to treat Alice's specific condition. Afterwards, the Pharmaceutical company acknowledges that the drugs order is received (DOR), proceeds to produce the drugs in the laboratory (PDL), and ships the drugs (SD) back to the Hospital. Upon receiving the medications, the Hospital administer the drug (AD), and conducts an assessment to determine if Alice can be treated internally. If specialized care is required, Alice is moved from the Hospital to the Specialized clinic (PAFH). When the patient arrives from the Hospital (PAFH), the Specialized clinic performs in-depth analyses (PIA) and proceeds with the treatment (PT). Once the Specialized clinic had completed the evaluations and verified the response to the alternative treatment (VRT), it transfers the patient back TPB. The Hospital receive the Alice patient back (RPB) and prepares the necessary clinic documentation (PCD). If Alice has successfully recovered, declares her as healed (DPH). When Alice's treatment is complete, the Hospital discharges the patient (DP). Bob enters the Hospital a few hours

later than Alice. His hospitalization process is similar to Alice's. However, he does not need specialized care, and his case (711) is only treated by the Hospital. Therefore, the Hospital perform the response to treatment analyses (PRTA) instead of transferring him to the Specialized clinic. Both the National Institute of Statistics of the country in which the three organizations reside, together with the University that hosts/manages the hospital, wish to uncover information on this inter-organizational process for reporting and auditing purposes [**?**] via process analytics. The involved organizations share the urge for such an analysis, and wish to be able to repeat the mining task also in-house. The Hospital, the Specialized clinic, and the Pharmaceutical company have a partial view of the overall unfolding of the inter-organizational process as they record the events stemming from the parts of their pertinence. In Table 1, e.g., we show the traces 312 and 711 recorded by the Hospital (i.e., T^H_{312} and T^H_{711}), the Specialized clinic (i.e., T^S_{312} and T^S_{711}), and the Pharmaceutical company (i.e., T^C_{312} and T^C_{711}). Those traces are projections of the two combined ones for the whole inter-organizational process: T_{312} =⟨PH, COPA, OD, DOR, PDL, SD, RD, AD, TP, PAFH, PIA, PT, VRT, TPB, RPB, DPH, PCD, DP⟩ and T_{711} =⟨PH, COPA, OD, DOR, PDL, SD, RD, AD, TP, PAFH, DPH, PCD, DP⟩. Results stemming from the analysis of the local traces would not provide a full picture. Data should be merged. However, to preserve the privacy of the people involved and safeguard the confidentiality of the information, the involved parties cannot give open access to their traces to other organizations. The diverging interests (being able to conduct process mining on data from multiple sources without giving away the local event logs in-clear) motivate our research. In the following, we describe the design of our solution.

4 Design

In this section, we present the high-level architecture of the CONFINE framework. We consider the main functionalities of each component, avoiding details on the employed technologies discussed in the next sections. After introducing the architecture, we focus on the `Secure Miner`, a core component of our contribution.

4.1 CONFINE architecture at large

Our architecture involves different organizational ecosystems characterized by one or more machines. An organization may take at least one of the following roles: **provisioning** if it delivers local event logs to be collaboratively mined; **mining** if it applies process mining algorithms using event logs retrieved from provisioners. In Fig. 2, we propose the high-level schematization of the CONFINE framework. In our solution, every organization hosts one or more `Node`s. Depending on the played role, `Node`s come endowed with a `Provisioner` or a `Secure Miner` component, or both. The `Provisioner` component consists of the following two main sub-components. The `Log Recorder` registers the events taking place in the

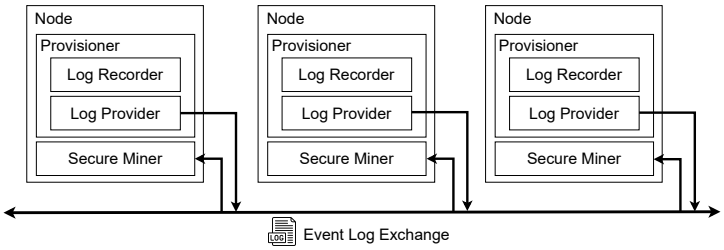


Fig. 2: CONFINE high-level architecture.

organizations' systems. The `Log Provider` delivers on-demand data to mining players. The Hospital (as well the other parties in our running example) records Alice and Bob's traces using the `Log Recorder`. The `Log Recorder` is queried by the `Log Provider` for event logs to be made available for mining. The latter controls access to local event logs by authenticating data requests by miners and rejecting those that come from unauthorized parties. In our motivating scenario, the Specialized clinic, Pharmaceutical company, and the Hospital leverage `Log Providers` to authenticate the miner party before sending their logs. The `Secure Miner` component shelters external event logs inside a protected environment to preserve data confidentiality and integrity. Notice that `Log Providers` accept requests issued solely by `Secure Miners`. Next, we provide an in-depth focus on the latter.

4.2 Secure Miner

The primary objective of the `Secure Miner` is to allow miners to securely execute process mining algorithms using event logs retrieved from provisioners such as the Specialized clinic, Pharmaceutical company, and the Hospital of our running example. `Secure Miners` are isolated components that guarantee data inalterability and confidentiality. In Fig. 3, we show a schematization of the `Secure Miner`, which consists of four sub-components: *(i)* The `Log Requester`; *(ii)* The `Log Receiver`; *(iii)* The `Log Manager`; *(iv)* The `Log Elaborator`. The `Log Requester` and the `Log Receiver` are the sub-components that we employ during the event log retrieval. `Log Requesters` send authenticable data requests to the `Log Provider` component of provisioners. The `Log Receiver` collects event logs sent by `Log Providers` and entrusts them to the `Log Manager`, securing them from accesses that are external to the `Secure Miner`. Miners of our motivating scenario, such as the University and the National Institute of Statistics, employ these three components to retrieve and store Alice and Bob's data. The `Log Elaborator` merges the event data locked in the `Secure Miner` to have a global view of

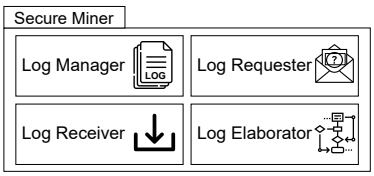


Fig. 3: Sub-components of the Secure Miner.

the inter-organizational process comprehensive of activities executed by each involved party. Thereupon, it executes process mining algorithms in a protected environment, inaccessible from the outside computation environment. In our motivating scenario, the `Log Elaborator` combines the traces of Alice (i.e., T^H_{312}, T^S_{312}, and T^C_{312}) and Bob (i.e, T^H_{711}, T^S_{711}, and T^C_{711}), generates the chronologically sorted traces T_{312} and T_{711}, and feeds them into the mining algorithms (see the bottom-right quadrant of Table 1).

5 Realization

In this section, we outline the technical aspects concerning the realization of our solution. Therefore, we first present the enabler technologies through which we instantiate the design principles presented in Sect. 4. After that, we discuss the CONFINE interaction protocol. Finally, we show the implementation details.

5.1 Deployment

As follows, we bridge the gap between the CONFINE high-level architecture and its practical realization. Fig. 4 depicts a *UML deployment diagram* [10] that aims to help with understanding the instantiated technologies. We differentiate between the technologies designated for mining, denoted as `Miner Node`s, and those specifically associated with provisioners, identified as `Provisioner Node`s. To enhance clarity, we maintain the separation of these *devices* in the accompanying diagram. However, organizations have the flexibility to opt for integrated technologies that incorporate both mining and provisioning functionalities. Using our motivating scenario as an example, the Hospital can be equipped with machines aimed for both mining and provisioning, while the Specialized clinic can make use of separate devices. We included the `Log Recorder`, the `Log Provider`, and `Secure Miner` (already discussed in Sect. 4) as abstract *components* of the diagram, whose manifestations are described as follows.

`Provisioner Node`s host `Provisioner`s components, encompassing the `Log Recorder` and `Log Provider`. We manifest the `Log Recorder` in the Process-Aware Information System (`PAIS`), which plays a crucial role in managing various business processes, including accounting and resource management [6]. In our motivating scenario, the Hospital and the other provisioners generate Alice and Bob's traces through this class of systems. The `PAIS` grants access to the `Log Server`, enabling it to retrieve event logs. The `Log Server`, on the other hand, embodies the functionalities of the `Log Provider`, implementing web services aimed at handling remote data requests and providing event log data to miners. The Hospital, the Specialized Clinic, and the Pharmaceutical Company of our running example employes `Log Servers` adhering to established web standards such as HTTP[1], FTP[2], and Goopher[3]. The `PAIS` and `Log Server` run on the top of the `Operating System` of the `Provisioner Node`.

[1] w3.org/Protocols/rfc2616/rfc2616.html. Accessed: 07/11/2023.
[2] w3.org/Protocols/rfc959/. Accessed: 07/11/2023.
[3] datatracker.ietf.org/doc/html/rfc1436. Accessed: 07/11/2023.

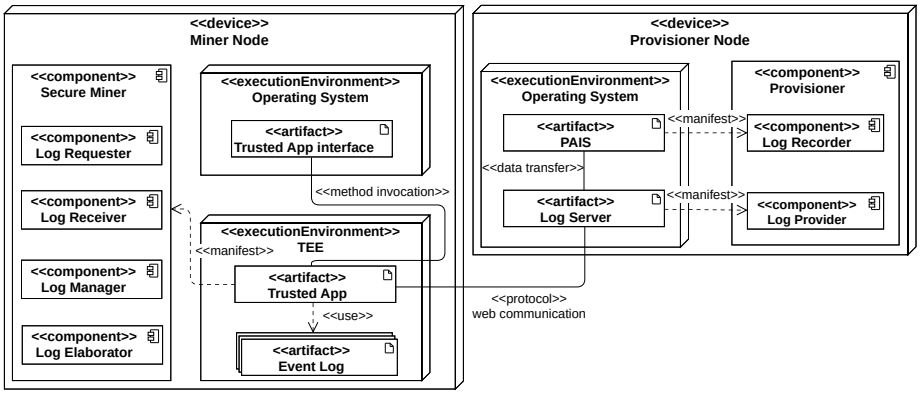


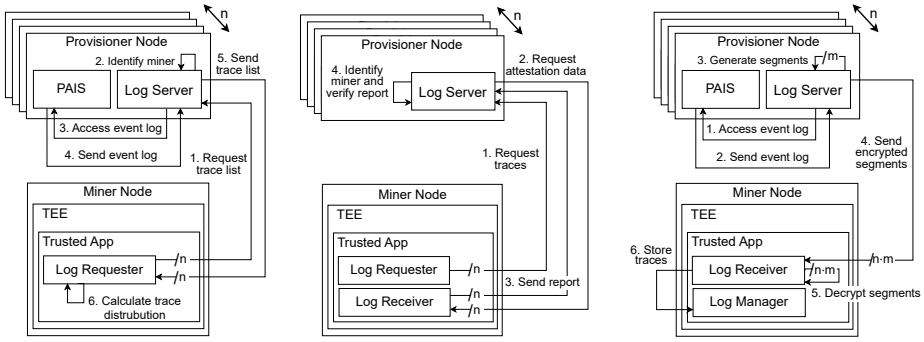
Fig. 4: UML deployment diagram.

The `Miner Node` is characterized by two distinct *execution environments*: the `Operating System` and the Trusted Execution Environment (`TEE`). `TEE`s establish isolated contexts separate from the normal `Operating System`, safeguarding code and data through hardware-based encryption mechanisms. This technology relies on specialized components of the `Miner Node`'s CPU capable of handling encrypted data within a reserved section of RAM [**?**]. We leverage the security guarantees provided by `TEE`s to protect a `Trusted App` responsible for fulfilling the functions of the `Secure Miner` and its associated subcomponents. The `TEE` ensures the integrity of the `Trusted App` code, protecting it against malicious manipulations and unauthorized access by entities running within the `Operating System`. Additionally, we utilize the isolated environment of `TEE`s to securely store event log data (e.g., Alice and Bob's traces of our example) from provisioner organizations within the `Miner Node`. The `TEE` safeguards this sensitive information alongside a unique public and private key couple used for attestation purposes, preventing exposure to the `Operating System`. Access to data located in the `TEE` is restricted solely to the `Trusted App`. Users interact with the `Trusted App` through the `Trusted App Interface`, which serves as the exclusive communication channel. The `Trusted App` offers secure methods, invoked by the `Trusted App Interface`, for safely receiving information from the `Operating System` and outsourcing the results of computations, maintaining a high level of data security.

5.2 CONFINE protocol

In Fig. 5, please rename components as per their concept-diagram name. Hence, PAIS → *PAIS* \n : Log Recorder, Log Server → *Log Server* \n : Log Provider; Trusted App → *Trusted App*: Secure Miner. Make the TEE boundary box thicker and dark-grey (or semi-transparent black). On overlay, make sure the boundary box is below the arrows. The same holds for Fig. 5, with OS with a thick grey boundary box.

We separate the protocol into subsequent stages, namely *initialization*, *remote attestation*, *data transmission*, and *computation*. This sequence of phases is



(a) Initialization (b) Remote Attestation (c) Data Transmission

Fig. 5: Initialization, remote attestation and data transmission phases of the CONFINE protocol.

enacted by two principal entities: a `Miner Node` and a variable number, denoted as n, of `Provisioner Nodes`. We depict each stage in Fig. 5 and Fig. 6.

Initialization. The objective of the initialization stage is to inform the miner about the distribution of traces related to a business process among the `Provisioner Nodes`. At the onset of this stage, the `Log Requester` component within the `Trusted App` issues n requests to the `Log Server` components of the provisioners for the list of owned traces (step 1, in Fig. 5(a)). Following sender authentication (2), each `Log Server` retrieves the local event log from the `PAIS` (3, 4) and subsequently responds to the `Log Requester` by providing a list of its associated traces (5). After collecting these n responses, the `Log Requester` delineates the distribution of traces. In the context of our motivating scenario, by the conclusion of the initialization phase, the miner gains knowledge that traces belonging to Bob's process, specifically T^H_{711} and T^C_{711}, are exclusively retained by the Hospital and the Specialized Clinic. In contrast, traces recording the Alice's process, denoted as T^H_{312}, T^C_{312}, and T^S_{312}, are scattered across all three organizations.

No special formatting needed for step numbers.

Remote Attestation. The remote attestation serves the purpose of establishing trust between miners and provisioners in the context of fulfilling data requests. This phase adheres to the overarching principles outlined in the RATS RFC standard [3] serving as the foundation for several attestation schemes (e.g., Intel EPID,[4] and AMD SEV-SNP[5]). Remote attestation has a dual objective: (i) to furnish provisioners with compelling evidence that the data request for an event log originates from a `Trusted App` executing within a `TEE`, and (ii) to confirm the precise nature of the `Trusted App` as an authentic `Secure Miner` software entity. Upon the initiation of a new log request by the `Log Requester` (1, in Fig. 5(b)), each of the n `Log Servers` commences the verification process

[4] sgx101.gitbook.io/sgx101/sgx-bootstrap/attestation. Accessed: 07/11/2023.
[5] amd.com/en/processors/amd-secure-encrypted-virtualization. Accessed: 07/11/2023.

by requesting the necessary information from the `Log Receiver` to conduct the attestation (2). Subsequently, the `Log Receiver` generates the attestation report containing the measurement of the `Trusted App`, which is defined as the hash value of the combination of the code and initial data of the `Secure Miner`. Once this report is signed using the attestation private key associated with the hardware of the `Miner Node`, it is transmitted by the `Log Receiver` to the `Log Servers` alongside the attestation public key of the `Miner Node` (3). The `Log Servers` authenticate the miner using the public key and decrypt the report (4). In this last step, the `Log Servers` undertake a comparison procedure in which they juxtapose the measurement found within the decrypted report against a predefined reference value associated with the source code of the `Secure Miner`. If the decrypted measurement matches the predefined value, the `Miner Node` gains trust from the provisioned.

Data Transmission. Once verified the trusted nature of the `Trusted App`, the `Log Servers` proceed with the transmission of their traces. To accomplish this, each `Log Server` retrieves the event log from the `PAIS` (1 and 2, in Fig. 5(c)), and divides it into segments, resulting in m segments (3). Each of these segments contains a variable number of complete traces, with the cumulative size remaining within the threshold specified by the miner as a parameter of the initial request. As an illustrative example from our motivating scenario, the `Log Server` of the Hospital may structure the segmentation such that T^H_{312} and T^H_{711} reside within the same segment, whereas the Specialized clinic might have T^S_{312} and T^S_{711} in separate segments. Subsequently, the n `Log Servers` transmit their m encrypted segments to the `Log Receiver` of the `Trusted App` (4). The `Log Receiver`, in turn, decrypts each of the $n \times m$ responses (5) and securely stores the traces within the `TEE` through the `Log Manager` (6).

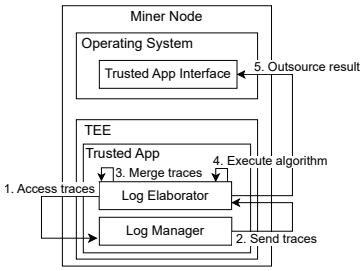


Fig. 6: Computation phase of the CONFINE protocol.

Computation. The `Trusted App` requires all the provisioners to have delivered traces referring to the same process instances. For example, when T^H_{312}, T^S_{312} and T^C_{312} have all been received by the `Trusted App`, the process instance associated with Alice becomes eligible for computation. The computation procedure may be activated either manually by the user operating the `Miner Node` or automatically upon the fulfillment of predetermined conditions (e.g., as soon as all the traces are collected). Upon meeting these conditions, the `Log Elaborator` requests the traces earmarked for computation from the `Log Manager` (1 and 2, in Fig. 6). To reconstruct a complete process instance traces belonging to the same process instance must be merged by the `Log Elaborator` in a single trace (e.g., T_{312} for Alice case) comprehensive of all the events in the partial traces (e.g., T^H_{312}, T^S_{312} and T^C_{312} for Alice case). To accomplish this, the `Log Elaborator` applies a specific *merging schema* as stated in [4]. According to this research work, a merging schema is defined as a rule specifying the attributes that link two

traces during the merging process. In our illustrative scenario, the traces of Alice and Bob are related by the same case id (i.e., 312 for Alice case, and 711 for Bob case). Therefore, the merging schema to combine their traces is contingent upon the linkage established by the `concept:name` attribute, as exemplified below:

```
merge all traces where
  "concept:name" of trace in log A
        equals
  "concept:name" of trace in log B
```

We underline that our solution allows for various merging schemas involving different trace attributes. After the merging operation (3), the `Log Elaborator` proceeds to input the merged traces into the process mining algorithm (4). Ultimately, the outcome of the computation is relayed by the `Log Elaborator` from the `TEE` to the `Trusted App Interface` running atop the `Operating System` of the `Miner Node` (5).

5.3 Implementation

We implemented the `Secure Miner` component as an Intel SGX[6] trusted application, encoded in Go through the EGo framework.[7] To demonstrate the effectiveness of our framework, we integrated the *HeuristicsMiner* discovery algorithm within the `Trusted Application`. We established the communication between miners and provisioners using the HTTP web protocol. Upon successful execution, our *HeuristicsMiner* implementation generates workflow nets corresponding to the analyzed processes. For additional details, our implementation can be accessed publicly at the following url: github.com/dave0909/TEExProcessMining/.

6 Discussion

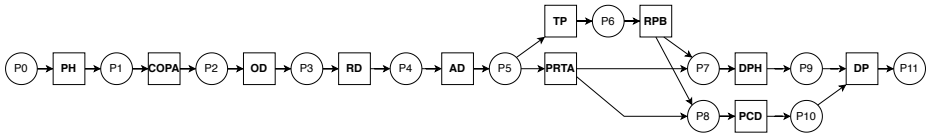
In this section, we evaluate the proposed approach. In the first paragraph of the section we delve into a convergence analysis by evaluating the correctness of the collaborative data exchange process. Subsequently, we took into assessment the memory usage by measuring the RAM usage associated with the execution of the `Secure Miner` components, using diverse parameter configurations.

Convergence of the outputs. We assess the convergence of the *HeuristicsMiner* outcomes as a means of validating the correct functioning of the event log transmission mechanism. Specifically, we generated workflow net outputs by applying intra-organizational mining procedures, wherein each organization independently mined its event log. Subsequently, we employed our approach with a miner actor conducting the *HeuristicsMiner* algorithm using an inter-organizational event log obtained as a result of the log exchange and merging mechanisms. Finally, we compared the results obtained through our approach with the single outputs mined using the partial event log in intra-organizational setting. For the purpose of this test, we utilize the synthetic event log based

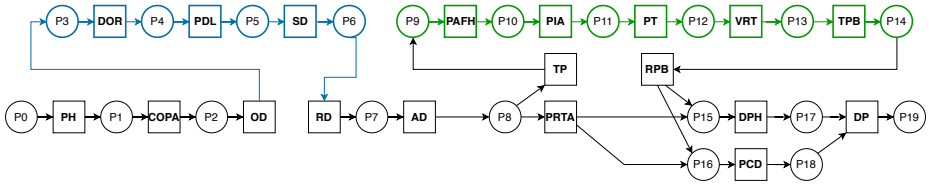
[6] sgx101.gitbook.io/sgx101/. Accessed: 07/11/2023.
[7] docs.edgeless.systems/ego. Accessed: 07/11/2023.

(a) Workflow net mined from the Pharmaceutical company log.

(b) Workflow net mined from the Specialized clinic log.



(c) Workflow net mined from the Hospital log.



(d) Workflow net mined from the merged log in our inter-organizational setting.

Fig. 7: Outputs of the HeuristicsMiner used for the convergence test.

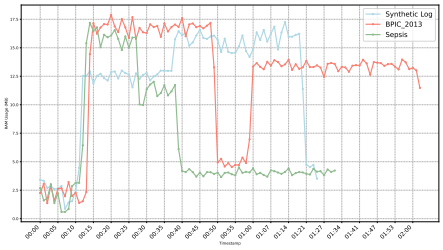
Table 2: Tested event logs.

Name	**Type**	**Activities**	**Cases**	**Max** tr. len.	**Min** tr. len.	**Avg**. tr. len.	**Organization** $\mapsto$ **Activities**
Motivating Scenario	Synthetic	19	1000				$\mathcal{O}^P \mapsto 3$, $\mathcal{O}^C \mapsto 5$, $\mathcal{O}^H \mapsto 14$
Sepsis	Real	19	4000				1
BPIC2013	Real	35	1432				3

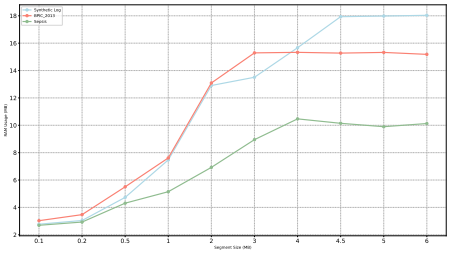
on our motivating scenario, which features a corresponding BPMN diagram (as depicted in Fig. 1). Upon careful examination of Fig. 9, we show that the workflow net generated by the *HeuristicsMiner* in our approach, as displayed in Fig. 7(d), encapsulates the structure and behavior observed in the workflow nets produced by the intra-organizational *HeuristicsMiner* execution. In specific terms, the blue-colored part in Fig. 7(d) reflects the behavior of the Pharmaceutical company, depicted in Fig. 9(a), commencing from the moment the drug order is received to its fulfillment. Furthermore, the green-colored part delineates the process of the Specialized clinic, starting from the patient's arrival from the `Hospital` to their transfer described in Fig. 9(b). Finally, the black-colored portion of the workflow net is consistent with the output resulting from the partial log of the Hospital showed in Fig. 7(c).

Memory Usage.

Scalability.

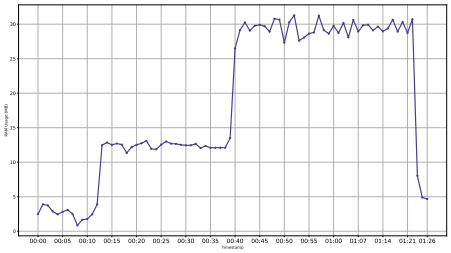


(a) Memory usage at runtime.

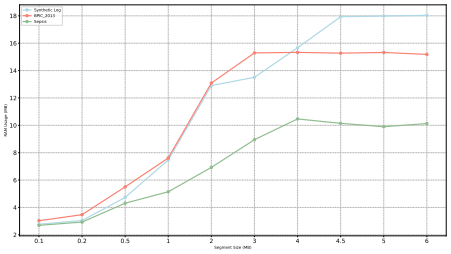


(b) Memory usage as segment size increases.

Fig. 8: Memory usage tests.



(a) FIGURE TO BE REPLACED WITH LINEPLOT (THROUGHPUT;n°organizaitons)



(b) FIGURE TO BE REPLACED WITH LINEPLOT(RAM;n°organizations)

Fig. 9: FIGURE FOR SCALABILITY TEST TO BE REPLACED

7 Conclusion and Future Work

Confidentiality is of paramount importance in inter-organizational process mining due to the transmission of sensitive data across organizational boundaries. Our research investigates a secrecy-preserving approach that enables organizations to employ process mining techniques with event logs from multiple organizations while ensuring the protection of privacy and confidentiality. Our solution still has room for improvement. We operate under the assumption of fair conduct by data provisioners and do not account for the presence of injected or maliciously manipulated event logs. In addition, we do not handle TEE crashes and suppose that miners and providers exchange messages in perfect communication channels where no loss, no snap, and no bit corruption occours. Additionally, our approach relies on certain assumptions about event log data, including the existence of a universal clock for event timestamps, which may not be realistic in situations where organizations are not perfectly synchronized. To address this challenge, we intend to explore a solution based on Network Time Protocols (NTP). Our future work encompasses the development of a formalized interaction protocol governing the communication between data provisioners and miners. The presented solution

embraces model process mining techniques in a general way. However, we believe that the presented approach is particularly compatible with declarative model representations. Therefore, trusted applications could compute and store the entire set of rules representing a business process, and users may interact with them via trusted queries. Finally, in our implementation, we have focused on process discovery tasks. However, our approach has the potential to seamlessly cover a wider array of process mining functionalities such as *conformance checking*, and *performance analysis* techniques. Implementing them and show their integrability with our approach paves the path for future work.

References

1. van der Aalst, W.M.P.: Intra-and inter-organizational process mining: Discovering processes within and between organizations. In: IFIP WG 8.1 (2011)
2. van der Aalst, W.M.: Federated process mining: Exploiting event data across organizational boundaries. In: SMDS 2021 (2021)
3. Birkholz, H., Thaler, D., Richardson, M., Smith, N., Pan, W.: Remote ATtestation procedureS (RATS) Architecture (2023)
4. Claes, J., Poels, G.: Merging event logs for process mining: A rule based merging method and rule suggestion algorithm. Expert Syst. Appl. **41**(16) (2014)
5. Cramer, R., Damgård, I.B., et al.: Secure multiparty computation. Cambridge University Press (2015)
6. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, Second Edition (2018)
7. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Secure multi-party computation for inter-organizational process mining. In: BPMDS/EMMSAD CAiSE (2020)
8. Elkoumy, G., Fahrenkrog-Petersen, S.A., Dumas, M., Laud, P., Pankova, A., Weidlich, M.: Shareprom: A tool for privacy-preserving inter-organizational process mining. BPM (PhD/Demos) **2673** (2020)
9. Hernandez-Resendiz, J.D., Tello-Leal, E., Marin-Castro, H.M., Ramirez-Alcocer, U.M., Mata-Torres, J.A.: Merging event logs for inter-organizational process mining. In: New Perspectives on Enterprise Decision-Making Applying Artificial Intelligence Techniques (2021)
10. Koch, N., Kraus, A.: The expressive power of uml-based web engineering. In: IWWOST02. vol. 16 (2002)
11. Müller, M., Simonet-Boulogne, A., Sengupta, S., Beige, O.: Process mining in trusted execution environments: Towards hardware guarantees for trust-aware inter-organizational process analysis. In: ICPM (2021)
12. Van Der Aalst, W., Adriansyah, A., De Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., Van Den Brand, P., Brandtjen, R., Buijs, J., et al.: Process mining manifesto (2012)
13. Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.Z., Li, H., Tan, Y.a.: Secure multi-party computation: theory, practice and applications. Inf. Sci. **476** (2019)