IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 February 2023, accepted 21 March 2023, date of publication 31 March 2023, date of current version 6 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3263673

RESEARCH ARTICLE

An Abstraction-Based Approach for Privacy-Aware Federated Process Mining

MAJID RAFIEI AND WIL M. P. VAN DER AALST, (Fellow, IEEE)
Chair of Process and Data Science, RWTH Aachen University, 52074 Aachen, Germany

Corresponding author: Majid Rafiei (majid.rafiei@pads.rwth-aachen.de)

Funded under the Excellence Strategy of the Federal Government and the Länder. We also thank the Alexander von Humboldt Stiftung for supporting our research.

ABSTRACT Process awareness is an essential success factor in any type of business. Process mining uses event data to discover and analyze actual business processes. Although process mining is growing fast and it has already become the basis for a plethora of commercial tools, research has not yet sufficiently addressed the privacy concerns in this discipline. Most of the contributions made to privacy-preserving process mining consider an intra-organizational setting, where a single organization wants to safely publish its event data so that process mining experts can analyze the data and provide insights. However, in real-life settings, organizations need to collaborate for performing their processes, e.g., a supply chain process may involve many organizations. Therefore, event data and processes are often distributed over several partner organizations, yet organizations hesitate to share their data due to privacy and confidentiality concerns. In this paper, we introduce an abstraction-based approach to support privacy-aware process mining in inter-organizational settings. We implement our approach and demonstrate its effectiveness using real-life event logs.

INDEX TERMS Confidentiality, event data, federated process mining, inter-organizational process mining, privacy preservation.

I. INTRODUCTION

Process mining provides a family of techniques to discover, analyze, and improve latent business processes [1]. It provides fact-based and actionable insights into the actual processes using event logs. Three basic types of process mining are (1) *process discovery*, where the goal is to learn real process models from event logs, (2) *conformance checking*, where the aim is to find commonalities and disconformities between a process model and an event log, and (3) *process enhancement*, where the aim is to extend or improve a process model using different aspects of the available data.

Events are the smallest units of process execution characterized by their attributes. Process mining requires that each event contains at least the following main attributes to enable the application of analysis techniques: *case id*, *activity*, and *timestamp*. The *case id* often refers to an individual to whom the event belongs, e.g., a patient or customer. The *activity* refers to the activity associated with the event, and the *timestamp* is the time when the activity was executed. A sequence of events having a fixed ordering based on their timestamps is called a *trace*, which is considered a crucial case attribute for process mining techniques.

Depending on the context of a process, the corresponding events may contain more attributes. For example, in the healthcare context, the *resource* attribute can be used to indicate the activity performer, e.g., a nurse, or an event attribute may show the *disease* of the corresponding patient. Table 1 shows a part of an event log recorded by an information system in a hospital. Some of the event attributes may refer to individuals such as *case id* and *resource*. For instance, in Table 1, the *case id* attribute refers to the patients whose data are recorded, and the *resource* attribute refers to the employees performing activities for the patients.

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

TABLE 1. **Sample event log (each row represents an event).**

Case Id	Activity	Timestamp	Resource	Disease
1	Registration (RE)	01.01.2019-08:30:00	Employee 4 (E4)	Corona
1	Visit (VI)	01.01.2019-08:45:00	Doctor 1 (D1)	Corona
1	Release (RL)	01.01.2019-08:58:00	Employee 6 (E6)	Corona
2	Registration (RE)	01.01.2019-09:00:00	Employee 1 (E1)	Cancer
3	Registration (RE)	01.01.2019-09:05:00	Employee 4 (E4)	Flu
2	Visit (VI)	01.01.2019-09:20:00	Doctor 3 (D3)	Cancer
2	Hospitalization (HO)	01.01.2019-09:55:00	Employee 6 (E6)	Cancer
2	Blood Test (BT)	01.01.2019-10:10:00	Nurse 2 (N2)	Cancer
3	Visit (VI)	01.01.2019-10:20:00	Doctor 3 (D3)	Flu
3	Blood Test (BT)	01.01.2019-10:40:00	Nurse 2 (N2)	Flu
3	Hospitalization (HO)	01.01.2019-12:20:00	Employee 2 (E2)	Flu
3	Release (RL)	01.01.2019-14:20:00	Employee 6 (E6)	Flu
2	Release (RL)	02.01.2019-16:00:00	Employee 2 (E2)	Cancer
4	Registration (RE)	02.01.2019-16:10:00	Employee 2 (E2)	HIV
4	Injection (IN)	02.01.2019-16:30:00	Nurse 2 (N2)	HIV
4	Release (RL)	02.01.2019-18:00:00	Employee 2 (E2)	HIV

Moreover, some attributes may be considered as sensitive attributes, e.g., the *disease* attribute in Table 1. Privacy issues are highlighted when person-specific information is included in an event log. For example, in Table 1, knowing that "Injection" was performed for a patient, the corresponding case, which is case 4, is re-identified. Consequently, the whole sequence of activities performed for the patient and also the disease are disclosed.

The terms *inter-organizational process mining*, *cross-organizational process mining*, and *federated process mining* all refer to a sub-discipline of process mining where the goal is to jointly discover, monitor, analyze, and improve cross-organizational processes [2], [3], [4], [5]. Remaining in the healthcare context, consider a collection of clinics and hospitals involved in the treatment process of some patients. Federated process mining can be used to discover the overall treatment process that traverses several hospitals, find bottlenecks in the process, identify successful/failed treatment processes, etc. However, process mining is rarely applied in an inter-organizational setting mainly due to privacy/confidentiality concerns. Setting the right inter-organizational boundaries, regarding privacy issues, is an important element of advancing process mining.

Organizations, such as healthcare providers, are obviously unwilling to share their entire event logs containing highly sensitive information with other parties involved in a joint process. Moreover, they cannot afford to trust third parties. Thus, the main challenge of Privacy-Aware Federated Process Mining (PAFPM) is to get process mining insights regarding the entire process while considering privacy, and without a need for a trusted third party.

We consider two main levels for privacy concerns in federated process mining: the *individual level* and the *organizational level*. The former aims to protect private data belonging to individuals in organizations. The latter considers all the internal activities of an organization as sensitive private information that should not be revealed. We propose an approach that can address both levels of privacy concerns. Our approach for PAFPM focuses on the following challenges: (1) no need to bound the number of involved parties, (2) no need for a trusted third party, (3) no need for designing complex communication protocols among parties, e.g., *secure multi-party computation* protocols, and (4) possibility of ensuring that all the involved parties can share data while allowing for the necessary levels of data utility and data privacy.

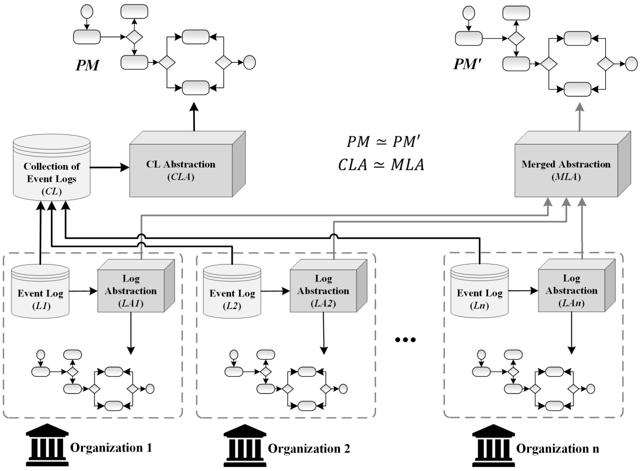


FIGURE 1. **The general overview of the abstraction-based approach for privacy-aware federated process mining. Inside the dashed squares is considered as the *trusted environment*, and outside these squares is considered as the *untrusted environment*.**

The proposed approach is based on the concept of *abstractions* in process mining [6]. Abstractions are intermediate results of process mining algorithms that relate event logs and final results. For example, a *directly follows graph*, representing directly follows relations between activities, is an abstraction of process discovery algorithms that relates an event log to a formal process model describing the observed behavior in the event log. Abstractions are generated by specific abstraction functions that reduce event logs, containing highly sensitive detailed information, to the minimal information required for obtaining specific results. Thus, sharing abstractions is of lower risk for organizations compared to sharing original event logs. However, using abstractions arises the following challenges that need to be addressed. First, given a single event log, the effectiveness of an abstraction function on both aspects of data utility and data privacy needs to be evaluated. For the specific type of abstraction used in this paper, we demonstrate the usefulness of the abstraction function. Second, abstractions need to be shared in such a way that the provided data utility and data privacy for a single event log do not degrade in an inter-organizational setting.

Figure 1 depicts the general overview of the abstraction-based approach for privacy-aware federated process mining. Gray arrows depict sharing abstractions where the same type of abstractions of event logs, i.e., obtained by applying the same abstraction function, are shared rather than sharing original sensitive event logs. The challenge w.r.t. the data utility is to merge abstractions in such a way that the merged abstraction is the same as the abstraction obtained from the collection of event logs by applying the same abstraction function. If so, the process mining results obtained by sharing abstractions are the same as the results obtained by sharing original event logs.

In this paper, we focus on the *control-flow* aspect of process mining that requires the basic three attributes (i.e., *case id*, *activity*, and *timestamp*) to perform two main activities of process mining, i.e., *process discovery* and *conformance checking*. After discovering a joint process model using the abstraction-based approach, we propose the so-called Risk-Aware Reveal Method (RARM) that can be used to answer more in-depth inquiries about the process while considering the privacy concerns of organizations.

The remainder is organized as follows. In Section II, the preliminaries are explained. In Section III, we discuss related work. In Section IV, we analyze the data utility and privacy constraints of abstractions, and the *risk-aware reveal method* is proposed to address the limitations. In Section V, we demonstrate our approach for privacy-aware federated process mining. In Section VI, we employ real-life event logs to show the effectiveness of the abstraction-based approach, and Section VII concludes the paper with a discussion regarding limitations and possible next steps.

II. BACKGROUND

In this section, we introduce some basic concepts and provide formal models that will be used in the remainder of the paper to describe the approach.

A. EVENT LOG

We first introduce some basic notations. 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 $\sigma : \{1, \ldots, n\} \rightarrow A$, represented as $\sigma = \langle a_1, a_2, \ldots, a_n \rangle$ where $a_i = \sigma(i)$ for any $1 \leq i \leq n$. $|\sigma|$ denotes the length of the sequence. For $\sigma \in A^*$, we write $a \in \sigma$, iff $\exists_{1 \leq i \leq |\sigma|} \sigma(i) = a$. Given A and B as two multisets, $A \uplus B$ is the sum over multisets, e.g., $[a^2, b^3] \uplus [b^2, c^2] = [a^2, b^5, c^2]$. For $\sigma_1, \sigma_2 \in A^*$, $\sigma_1 \sqsubseteq \sigma_2$ if σ_1 is a subsequence of σ_2, e.g., $\langle z, b, c, x \rangle \sqsubseteq \langle z, x, a, b, b, c, a, b, c, x \rangle$, and $\sigma_1 \oplus \sigma_2$ is the concatenation of two sequences, e.g., $\langle a, b, c \rangle \oplus \langle d, e \rangle = \langle a, b, c, d, e \rangle$. For $\sigma \in A^*$, $\{a \in \sigma\}$ is the set of elements in σ, and $[a \in \sigma]$ is the multiset of elements in σ, e.g., $[a \in \langle x, y, z, x, y \rangle] = [x^2, y^2, z]$.

Definition 1 (Event, Event Log): An **event** is a tuple $e = (c, a, t, r, d_1, \ldots, d_m) \in \mathcal{C} \times \mathcal{A} \times \mathcal{T} \times \mathcal{R} \times \mathcal{D}_1 \times \ldots \times \mathcal{D}_m$, where $c \in \mathcal{C}$ is the *case id*, $a \in \mathcal{A}$ is the activity associated with the event, $t \in \mathcal{T}$ is the event timestamp, $r \in \mathcal{R}$ is the *resource*, and $d_1, \ldots, d_m$ is a list of additional attributes values, where for any $1 \leq i \leq m$, $d_i \in \mathcal{D}_i$. $\mathcal{E} = \mathcal{C} \times \mathcal{A} \times \mathcal{T} \times \mathcal{R} \times \mathcal{D}_1 \times \ldots \times \mathcal{D}_m$ denotes the event universe.

For $e = (c, a, t, r, d_1, \ldots, d_m)$, $\pi_{case}(e) = c$, $\pi_{act}(e) = a$, $\pi_{time}(e) = t$, $\pi_{res}(e) = r$, and $\pi_{dom_i}(e) = d_i$, $1 \leq i \leq m$, are its projections. An **event log** is $L \subseteq \mathcal{E}$ where events are unique.

Definition 2 (Trace, Trace Variant): A **trace** $\sigma = \langle e_1, e_2, \ldots, e_n \rangle \in \mathcal{E}^*$ is a sequence of events, s.t., for each $e_i, e_j \in \sigma$: $\pi_{case}(e_i) = \pi_{case}(e_j)$, and $\pi_{time}(e_i) \leq \pi_{time}(e_j)$ for $1 \leq i < j \leq n$. A **trace variant** $\sigma \in \mathcal{A}^*$ is a trace where all the events are projected on the activity attributes.

Definition 3 (Simple Event Log): A simple event log is a multiset of trace variants, i.e., $L \in \mathcal{B}(\mathcal{A}^*)$. We assume that each trace in L belongs to an individual and $\sigma \neq \langle\rangle$ if $\sigma \in L$. $A_L = \{a \in \sigma \mid \sigma \in L\}$ is the set of activities in L, and $\tilde{L} = \{\sigma \in L\}$ denotes the set of trace variants in L.

Table 2 shows a simple event log derived from Table 1. In this paper, the term *event log* refers to a *simple event log* unless it is clearly mentioned that we mean a set of events.

Definition 4 (Entropy of Event Log): $ent : \mathcal{B}(\mathcal{A}^*) \rightarrow \mathbb{R}_{\geq 0}$ is a function which retrieves the entropy of traces in an event log, s.t., for $L \in \mathcal{B}(\mathcal{A}^*)$, $ent(L) = -\sum_{\sigma \in \tilde{L}} \frac{L(\sigma)}{|L|} log_2 \frac{L(\sigma)}{|L|}$. $max\_ent(L) = log_2 |L|$ is the maximal entropy that can be achieved when all the trace variants are unique.

Definition 5 (Directly Follows Relations (DFR)): Given $\sigma \in \mathcal{A}^*$, $DFR_\sigma = [(\sigma(i), \sigma(i+1)) \mid 1 \leq i < |\sigma|] \uplus [(\blacktriangleright, \sigma(1))] \uplus [(\sigma(|\sigma|), \blacksquare)]$ is the multiset of directly follows relations between activities in σ, where the first and last activities are tuples with $\blacktriangleright$ and $\blacksquare$ as the dummy start and end activities, respectively. The tuple including $\blacktriangleright$ is called the start relation, and the one including $\blacksquare$ is called the end relation. For $L \in \mathcal{B}(\mathcal{A}^*)$, $DFR_L = \biguplus_{\sigma \in L} DFR_\sigma$ is the multiset of directly follows relations between activities in the traces of L. Given $dfr = (a, b)$, $\pi_1(dfr) = a$ and $\pi_2(dfr) = b$ are the projections of dfr.

Definition 6 (Directly Follows Graph (DFG)): Let L be a simple event log, A_L be the set of activities in L, and DFR_L be the multiset of directly follows relations in L. $DFG_L = (A_L \cup \{\blacktriangleright, \blacksquare\}, DFR_L)$ is the directly follows graph of L.

Figure 2 shows the DFG of the simple event log shown in Table 2, where each node represents an activity, and the directed arcs represent the DFRs between activities. Note that $\blacktriangleright$ and $\blacksquare$ denote the dummy start and end activities, respectively. The numbers above arcs represent the frequency of the corresponding DFRs. A multiset of DFRs is a specific type of event log abstraction that can be used to generate a DFG. Definition 7 provides a generic definition for abstraction functions on simple event logs.

Definition 7 (Abstraction Function): Let $\mathcal{LA}$ be the universe of event log abstractions, e.g., DFRs, log statistics, footprints, etc. An abstraction function $abs : \mathcal{B}(\mathcal{A}^*) \rightarrow \mathcal{LA}$ is a function that maps a simple event log onto an abstraction. For $la \in \mathcal{LA}$, $abs^{-1}(la) = \{L \in \mathcal{B}(\mathcal{A}^*) \mid abs(L) = la\}$. For instance, $abs_{DFR} : \mathcal{B}(\mathcal{A}^*) \rightarrow \mathcal{B}(\mathcal{A} \cup \{\blacktriangleright\} \times \mathcal{A} \cup \{\blacksquare\})$ is an abstraction function that maps a given simple event log onto a multiset of DFRs.

Definition 7 shows that there could be more than one event log returning the same abstraction. For instance, consider $abs_{DFR} : \mathcal{B}(\mathcal{A}^*) \rightarrow \mathcal{B}(\mathcal{A} \cup \{\blacktriangleright\} \times \mathcal{A} \cup \{\blacksquare\})$. Given $L \in \mathcal{B}(\mathcal{A}^*)$, $abs_{DFR}(L) = DFR_L$, and $abs^{-1}_{DFR}(DFR_L) = \{L \in \mathcal{B}(\mathcal{A}^*) \mid abs_{DFR}(L) = DFR_L\}$. Considering L_1 as our example event log (Table 2) and the corresponding DFRs, $L_2 = [\langle RE, VI, RL \rangle, \langle RE, VI, HO, BT, HO, RL \rangle, \langle RE, VI, BT, RL \rangle, \langle RE, IN, RL \rangle] \in abs^{-1}(DFR_{L_1})$ is another event log with the same multiset of DFRs.

TABLE 2. The simple event log derived from Table 1.

Trace Variant
$\langle RE, VI, RL\rangle$
$\langle RE, VI, HO, BT, RL\rangle$
$\langle RE, VI, BT, HO, RL\rangle$
$\langle RE, IN, RL\rangle$

B. DISCLOSURE RISKS

The main process mining activities, i.e., *process discovery* and *conformance checking*, can be performed using simple event logs including only sequences of activities. Such event logs, which do not contain other attributes, seem to be safe. However, the trace itself, as a complete sequence of activities performed for a case, is considered a sensitive attribute [7]. In this subsection, we demonstrate two types of disclosure risks associated with publishing simple event logs.

Consider the event log shown in Table 2. Assuming that an adversary knows that a patient's data are in the event log, little information about the activities performed for the patient could result in a successful re-identification. For example, if the adversary knows that an injection was performed for a victim patient, the only matching case is 4. We assume that the adversary's Background Knowledge (BK) is a subsequence of activities performed for a victim case which can be considered as the strongest assumable knowledge w.r.t. the available information in simple event logs. Thus, the attack model is defined as follows.

Definition 8 (Attack Model): Let L be a simple event log and A_L be the set of activities in L. We formalize the attack model as a function $match_L$:$A_L^* \rightarrow 2^L$. For $bk \in A_L^*$, $match_L(bk) = [\sigma \in L \mid bk \sqsubseteq \sigma]$.

For example, if the adversary knows that $bk = \langle HO, BT\rangle$ is a subsequence of activities performed for a case, case 2 is the only matching case. Once a case is re-identified *a complete sequence of activities performed for the case* is disclosed which is considered sensitive information. The strength of such an attack highly depends on the strength of the background knowledge that can be quantified based on the length of the sequence, so-called the size of BK [7].

Definition 9 (Background Knowledge Candidates): Let L be a simple event log. Given $l \in \mathbb{N}_{>0}$ as the size of background knowledge, $cand^l(L) = \{\sigma \in A_L^* \mid |\sigma| = l\}$ denotes the set of candidates for the background knowledge of size l.

For example, given Table 2 as the simple event log L, $cand^1(L) = \{RE, VI, HO, BT, IN, RL\}$. In [7], the authors introduce two main types of disclosure risks associated with such an attack: *case disclosure* and *trace disclosure*.

1) CASE DISCLOSURE

The uniqueness of traces w.r.t. the background knowledge of size l is used to measure the corresponding case disclosure in an event log. Equation (1) calculates the case disclosure which is the average uniqueness based on the candidates of background knowledge.

$$cd^l(L) = \frac{1}{|cand^l(L)|} \sum_{bk \in cand^l(L)} \frac{1}{|match_L(bk)|} \tag{1}$$

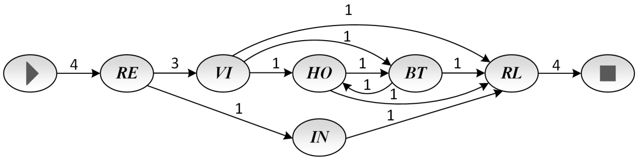


FIGURE 2. The DFG of the event log shown in Table 2. The nodes represent activities, and the arcs represent the directly follows relations.

The uniqueness alone cannot show some risks. Consider a scenario where for a candidate of BK there are several identical traces in the event log matching the knowledge. Since all the matching traces are the same, one can still know the trace of the case without the need for singling out a specific trace. Thus, *trace disclosure* is defined that is based on the entropy of matching traces. The less entropy of matching traces results in a high trace disclosure risk.

2) TRACE DISCLOSURE

The entropy of matching traces w.r.t. the background knowledge of size l is used to measure the corresponding trace disclosure in an event log. Equation (2) calculates the trace disclosure where $max\_ent(match_L(bk))$ is the maximal entropy for the matching traces that is achieved when all the matching traces are unique.

$$td^l(L) = 1 - \frac{1}{|cand^l(L)|} \sum_{bk \in cand^l(L)} \frac{ent(match_L(bk))}{max\_ent(match_L(bk))} \tag{2}$$

Note that in both Equations (1) and (2), equal weights are considered for the candidates of background knowledge. However, one can consider different weights based on different criteria, e.g., the sensitivity of activities. Moreover, the worst cases can be used rather than average values, i.e., the maximal uniqueness in Equation (1) or the minimal entropy in Equation (2).

Unsurprisingly, event logs containing more information provide more opportunities for attackers. If we consider an event log where traces are sequences of events with more attributes rather than only activities, each attribute could be an attack point or sensitive information. For example, in the event log shown in Table 1, if the adversary knows that for a victim patient, the visit activity performed by Doctor 1, the only matching case is 1. Once the case is re-identified all the other attributes are revealed, e.g., *disease* which is a sensitive attribute.

C. FEDERATED PROCESS MINING (FPM)

Federated process mining has been explored by researchers from different angles and in different contexts from EDI-supported inter-organizational business processes [3] to the application of Blockchain technology in cross-organizational process mining [8]. The majority of papers in this field focused on the application of process mining in supply

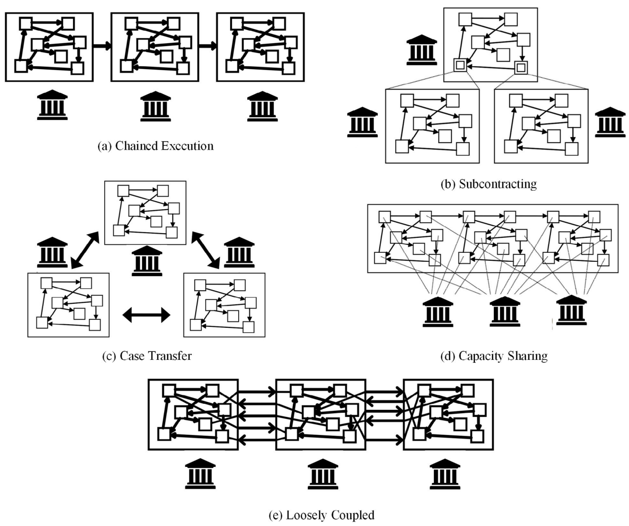


FIGURE 3. Different types of interoperability for inter-organizational process mining.

chains [9], [10], [11], while supply chains are one of the types of interoperability among organizations.

In [2], different types of interoperability are introduced (see Figure 3) including (a) *chained execution*: the process is split into a number of disjoint subprocesses that are executed by different organizations in sequential order, (b) *subcontracting*: one organization subcontracts subprocesses to other organizations, (c) *case transfer*: the process description is the same among organizations. However, cases can be transferred among organizations, and at any time, each case resides at exactly one organization, (d) *capacity sharing*: the process description is the same among organizations and the execution of tasks is distributed among organizations, and (e) *loosely coupled*: the process is cut into subprocesses, and different subprocesses could be active at the same time.

In most of the above-mentioned types of interoperability, the communication type can be *synchronous* or *asynchronous*. In the asynchronous type of communication, a case's process can be simultaneously executed in several organizations. On the contrary, in the synchronous type of communication, a case's process cannot be run in different organizations at the same time. In this paper, we consider the synchronous type of communication and describe our approach based on the different types of interoperability.

III. RELATED WORK

In this section, we provide a short summary of the research that has been done in *privacy-preserving process mining*, *federated process mining*, and *privacy-aware federated process mining*.

A. PRIVACY-PRESERVING PROCESS MINING

Privacy and confidentiality issues in process mining are recently growing in importance. The work having been done covers different aspects of the topic ranging from discussing challenges [12], [13], [14], [15], to providing privacy guarantees [16], [17], [18], [19], [20], [21] and *privacy quantification* [7], [22]. Confidentiality has been introduced as one of the main challenges of Responsible Process Mining (RPM) in [12]. The proposed privacy preservation techniques can be categorized into three different categories: *group-based*, *noise-based*, and *encryption-based*.

The group-based privacy preservation techniques are often based on the concept of k-anonymity [23] and its extensions, e.g., l-diversity [24] and t-closeness [25]. Some examples are as follows. In [19], the authors introduce a group-based privacy preservation technique for preserving the privacy of *resources*, who are performing activities. In [18] and [26], TLKC-privacy is introduced and extended to deal with high variability issues in event logs for applying group-based anonymization techniques. The noise-based privacy preservation techniques are based on the notion of *differential privacy* [27]. For example, in [16], [17], [21], [28], and [29], the notion of *differential privacy* is utilized to provide privacy guarantees in process mining. A general framework for confidentiality in process mining based on encryption and abstraction is proposed in [30].

There are also other work targeting other aspects of privacy and confidentiality in process mining. Some examples are as follows. In [13], the authors provide an overview of privacy challenges for process mining in human-centered industrial environments. The data privacy and utility requirements for healthcare event data are discussed in [14]. In [31], the authors propose a solution that allows the outsourcing of process mining while ensuring confidentiality. In [32], the goal is to propose a privacy-preserving system design for process mining. A privacy-preserving method for discovering roles from event logs is introduced in [33]. The risks regarding privacy degradation of privacy preservation techniques when event data are continuously published are discussed in [34] and [35]. In [22], the authors propose a measure to evaluate the re-identification risk of event logs. Also, in [7], a general privacy quantification framework, and some measures are introduced to evaluate the effectiveness of privacy preservation techniques. In [36], the authors propose a privacy extension for the XES standard (www.xes-standard.org) to manage privacy metadata. Some tools have also been provided to support the proposed techniques in practice such as [37], [38], and [39].

B. FEDERATED PROCESS MINING

In [2], inter-organizational process mining is explained and multiple categories of inter-organizational data flows are characterized. In [3], EDI messages are used to illustrate a case study of effective inter-organizational process mining in the automobile industry. In [4], the authors focus on improving the performance aspect of the process by utilizing the insights gained from cross-organizational process mining. In [40], the authors propose an approach to compare collections of process models and their event logs recorded in different Dutch municipalities. Furthermore, cloud computing [41] and blockchains [8] have been recognized as opportunities within the cross-organizational process mining context. In [9], an approach is proposed to discover distributed processes in supply chains. In [10], the authors describe basic patterns to capture modeling concepts that arise commonly in supply chains. In [11], the authors focused on different case notions in supply chains where objects are grouped, and in [5], the so-called *federated process mining* has been introduced to enable cross-organizational process mining by providing a framework that recommends event log abstractions.

C. PRIVACY-AWARE FEDERATED PROCESS MINING

Most related to our work are [42], [43], and [44]. In [42], the authors propose a technique based on *secure multi-party computation* for discovering *directly follows graphs* considering only two parties. In [43], the authors propose a framework for sharing public process models and discovering organization-specific process models from multiple parties which requires a trusted third party. In [44], the authors propose an approach for discovering process models in inter-organizational settings. This approach relies on a (semi) trusted third party and uses secure multi-party computation algorithms, e.g., for securely computing unions and aggregated values.

IV. PRIVACY AND UTILITY ANALYSIS

In this section, we analyze the data utility and privacy issues when directly follows relations are shared instead of an event log. We propose the risk-aware reveal method to overcome the data utility shortcomings.

A. PRIVACY ANALYSIS

The disclosure risks demonstrated in Subsection II-B are based on sequences of activities. Thus, it seems that removing the concept of trace by using the abstraction function abs_{DFR}, which maps an event log onto a multiset of DFRs, eliminates such risks. However, similar risk analyses can be done based on DFGs obtained from DFRs.

As demonstrated in Definition 7, the main advantage of sharing abstractions such as DFRs is that they impose uncertainty regarding original event logs. However, there may be a situation where certain information about original event logs can be revealed. In the following, we explain such a situation. Given an event log L, the complete paths on DFG_L, i.e., the paths from the start node to the end node, represent trace variants that may or may not be the variants of L. However, given an activity $a \in A_L$ as a node of DFG_L, if there exists only one complete path on DFG_L that contains a, then that path represents a trace variant of the original log L. For example, given the event log shown in Table 2 and the activity IN, the only complete path traversing this activity is $\langle \blacktriangleright, RE, IN, RL, \blacksquare \rangle$ which is a trace variant of the original event log.

Consider a scenario where the background knowledge of an adversary contains an activity that holds the above-mentioned condition. As a result, the whole sequence of activities performed for a victim case is disclosed. For instance, if the background knowledge of an adversary is $bk = \langle IN \rangle$, then the only matching path is $\langle \blacktriangleright, RE, IN, RL, \blacksquare \rangle$ that is the trace variant of case 4.

Such scenarios are more relevant to the disclosure risk analysis of results [45] and privacy preservation techniques for result protection that are out of the scope of this paper.

B. DATA UTILITY LIMITATIONS

Abstracting the control-flow aspect of an event log from a simple event log to a multiset of DFRs implies several data utility limitations such that even the most straightforward analyses that are based on traces cannot be performed. Two examples are shown below:

- *The most frequent traces in an event log*. For example, in the healthcare context, it may be important to know what are the most frequent sequences of activities performed for patients.
- *All traces that include a particular sequence or set of activities*. For example, it may be helpful to know what is the process of treatment for patients who had a blood test before being visited by doctors.

Obviously, it is also impossible to answer inquiries that are based on the attributes removed from an original event log by simplifying the event log. In the following, we provide two types of such inquiries:

- *All traces with a certain case or event attribute*. It may be helpful to know the process of treatment for patients of a specific range of age, or the patients who are visited by particular doctors.
- *A set of attributes based on other attributes*. For example, it may be important to know the set of activities performed by a particular doctor, or the set of joint activities performed by a set of doctors.

We categorize such queries into two main categories: Trace-Based Queries (TBQs) and Attribute-Based Queries (ABQs). All the queries expecting traces as responses are considered as *trace-based*, while *attribute-based* queries are those that expect event or case attributes as responses. In the following, we introduce the risk-aware reveal method that can be used to answer such in-depth questions regarding a process and mitigate the aforementioned utility limitations.

C. RISK-AWARE REVEAL METHOD (RARM)

Figure 4 shows the general overview of our abstraction-based approach for privacy-aware process mining in an intra-organizational setting. DFRs, as an abstraction of the control-flow aspect, are shared with process analysts to perform process discovery and get control-flow insights. To answer more in-depth questions triggered by control-flow insights, the Risk-Aware Reveal Method (RARM) can be used. RARM can provide more information in a selective manner to answer more in-depth questions regarding original traces and also removed attributes from an original event log.

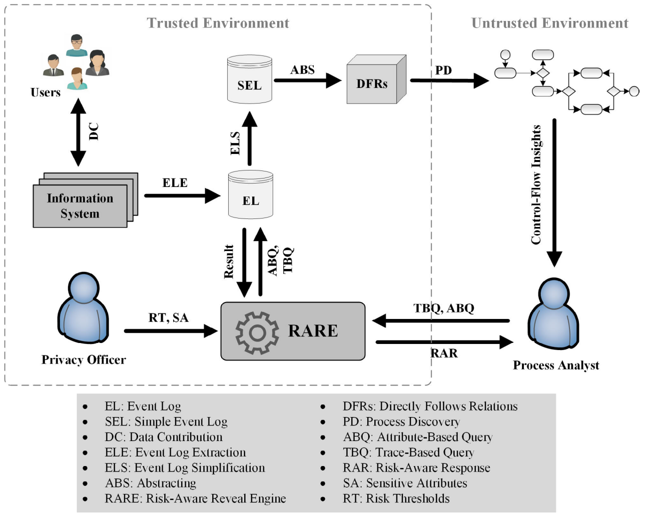


FIGURE 4. The general overview of our abstraction-based approach for privacy-aware process mining in an intra-organizational setting. Inside the dashed squares is considered as the *trusted environment*, and outside these squares is considered as the *untrusted environment*.

It can provide responses without revealing data about cases that are irrelevant to addressing a specific query.

The main component of RARM is the so-called Risk-Aware Reveal Engine (RARE). For each query, RARE first provides a response in the trusted environment. Then, it does a risk analysis before composing a response to the untrusted environment. If the risks are above a preset threshold, the engine refuses to answer the query. For TBQs, the result of a query is a multiset of traces, and RARE does *case disclosure* and *trace disclosure* analyses. For ABQs, RARE checks the sensitivity of the attribute of interest in the query based on the predefined set of sensitive attributes. If the attribute in the query is non-sensitive, the result can be shared. However, the queries regarding the sensitive attributes are refused. In Subsection V-G, we demonstrate the usage of RARM in inter-organizational settings, where several organizations have to be involved to provide answers for *trace-based* and *attribute-based* queries.

V. PRIVACY-AWARE FEDERATED PROCESS MINING

In this section, we expand our abstraction-based approach to enable Privacy-Aware Federated Process Mining (PAFPM) which has the following main properties. It does not limit the number of parties, it does not need a trusted third party, it supports privacy at both the *individual level* and the *organizational level*, and it does not require designing communication protocols among parties.

A. PROBLEM STATEMENT

For describing the main approach, we assume that privacy concerns are at the level of *individuals*. Particularly, as explained in Subsection II-B, traces are considered as sensitive private information. Nevertheless, we later explain that the approach can also support privacy at the level of *organizations*. We also consider the following standard assumptions. The sets of activities of organizations are disjoint, and the involved organizations share the same set of case identifiers for the joint cases. $C_{joint} \subseteq \mathcal{C}$ denotes the set of joint case identifiers. Note that organizations may use different case identifiers for the joint cases in their internal environment and use a mapping to map the shared identifiers to the internal ones. There are two main challenges when abstractions are shared rather than entire event logs: (C1) How to merge abstractions from different organizations such that the merged abstraction is the same as the abstraction that can be obtained by applying the same abstraction function to the merged event logs, and (C2) How to answer more in-depth questions regarding the information not included in the shared abstractions.

Definition 10 (Merging Abstractions Challenge): Let $\mathcal{O}$ be the universe of organization identifiers, and $\mathcal{LA}$ be the universe of abstractions. Consider $C_L = \{L_1, L_2, \cdots, L_n\}$ as an event log collection where $1 \leq i \leq n$ and $L_i \in \mathcal{B}(\mathcal{A}^*)$. Assume $C_{LA} = \{la_1, la_2, \cdots, la_n\}$ as the collection of abstractions, where $la_i = abs(L_i) \in \mathcal{LA}$ represents the abstraction of L_i belonging to the organization $o_i \in \mathcal{O}$ using abs as an abstraction function. If $merge(C_{LA}) \in \mathcal{LA}$ is an overall abstraction obtained by merging the individual abstractions, then $merge(C_{LA})$ have to be the same as $abs(C_L)$.

Since we consider DFRs as abstractions, the challenge of merging abstractions C1 is specialized to the challenge of merging DFRs. In inter-organizational process mining, merging DFRs is a challenge because of the missing so-called *handover relations* related to the interconnections among the organizations. Thus, to address challenge C1, we first define the concept of handovers. Then, we demonstrate the process of retrieving missing handover relations, and finally we explain the merging process based on the different types of interoperability described in Subsection II-C. We adapt RARM in the inter-organizational setting to address challenge C2.

B. HANDOVERS

A so-called *handover* happens when a case moves from one organization to another. A directly follows relation indicating a handover is called a *handover relation*, and the involved activities are called *handover activities*. The first handover activity of a handover relation is called the *handover to* activity that hands over a case to another organization. The second handover activity is called the *handover from* activity that receives the handed-over case from another organization.

Definition 11 (Handover Relation (HoR)): Let L and L' be two simple event logs belonging to two organizations involved in a joint process, $C_{joint} \subseteq \mathcal{C}$ be the set of joint cases, and $c_1 \in C_{joint}$. Consider $\sigma_{c_1} = \langle a_1, a_2, \ldots, a_n \rangle$ as the trace of case c_1 in L, and $\sigma'_{c_1} = \langle b_1, b_2, \ldots, b_m \rangle$ as the trace of case c_1 in L'. $hor = (a_i, b_j)$ is a handover relation, s.t., $1 \leq i \leq n$, $1 \leq j \leq m$, $a_i \in \sigma_{c_1}$, and $b_j \in \sigma'_{c_1}$. Given $hor = (a, b)$ as a handover relation, $\pi_1(hor) = a$ and $\pi_2(hor) = b$ are the projection functions, and considering $a' \in \mathcal{A}$ as an activity, $set_1(hor, a') = (a', b)$ assigns the activity a' to the first handover activity, and $set_2(hor, a') = (a, a')$ assigns a' to the second handover activity.

Since each organization has only access to its own event log, merged DFRs obtained by sharing the DFRs of different organizations in an inter-organizational setting do not include handover relations. To retrieve such missing relations, the organizations need to share the so-called *handover tables* together with DFRs. A handover table is a collection of *handover records* that are defined as follows.

Definition 12 (Handover Record): Let $\mathcal{O}$ be the universe of organizations identifiers including $\perp$ as the null identifier. A handover record is a tuple $rec = (id, c, o_1, a, o_2, o_3)$ where $id \in \mathbb{N}_{>0}$ is the incremental identifier of the record, $c \in \mathcal{C}$ is the case identifier of the case involved in the handover, $o_1 \in \mathcal{O}$ is the identifier of the organization generated the handover record, $a \in \mathcal{A}$ is a handover activity, $o_2 \in \mathcal{O}$ indicates the organization that handed over the case to o_1, and $o_3 \in \mathcal{O}$ indicates the organization that o_1 hands over the case to it. $\mathcal{HR} = \mathbb{N}_{>0} \times \mathcal{C} \times \mathcal{O} \times \mathcal{A} \times \mathcal{O} \times \mathcal{O}$ denotes the universe of handover records.

Given $rec = (id, c, o_1, a, o_2, o_3)$, $\pi_{id}(rec) = id$, $\pi_{case}(rec) = c$, $\pi_{org}(rec) = o_1$, $\pi_{act}(rec) = a$, $\pi_{from}(rec) = o_2$, and $\pi_{to}(rec) = o_3$ are the projections of the record. For any $(id, c, o_1, a, o_2, o_3) \in \mathcal{HR}$, $o_1 \neq o_2$, $o_1 \neq o_3$, $\{o_2\} \cup \{o_3\} \neq \{\perp\}$ and $\{o_2, o_3\} \cap \{\perp\} = \{\perp\}$.

Note the following constraints in Definition 12. An organization cannot have a handover with itself, and a handover record has to indicate one and only one type of handover activity, i.e., *handover from* or *handover to*. Given $(id, c, o_1, a, o_2, o_3) \in \mathcal{HR}$, if $o_2 \neq \perp$, then the record indicates a *handover from* activity, and if $o_3 \neq \perp$, then the record indicates a *handover to* activity.

Definition 13 (Handover Table (HoT)): Let $\mathcal{HR}$ be the universe of handover records. $HoT \subseteq \mathcal{HR}$ is a handover table. If $(id_1, c_1, o_{11}, a, o_{12}, o_{13}) \in HoT$ and $(id_2, c_2, o_{21}, b, o_{22}, o_{23}) \in HoT$, then $o_{11} = o_{21}$ and $id_1 \neq id_2$.

Consider the *chained execution* type of interoperability, where cases can move from one organization predictably to the next one. An example of this type of interoperability in the healthcare context could be a patient arriving at the emergency room, receiving a sepsis treatment, and ultimately a specialty check-up. Figure 5 shows example event logs for such a scenario, and Figure 6 shows the handover tables of the event logs in Figure 5. For example, the first record of EC's handover table shows that by performing IVA activity, EC hands over a case to ST. Consequently, the first record of ST's handover table shows that by performing REG activity, ST receives the handed-over case from EC. Note that *the records of a handover table must be inserted with the order that they happen in reality*.

C. RETRIEVING HANDOVERS

Algorithm 1 demonstrates the process of retrieving missing handover relations. Before explaining the algorithm, we need to define the selection operations over handover tables.

Algorithm 1 The Process of Retrieving Missing Handover Relations

```
Input: A collection of handover tables
       $HoTs = \{HoT_1, HoT_2, \cdots, HoT_n\}$
Input: A set of joint case identifiers $C_{joint} \subseteq \mathcal{C}$
Output: A multiset of handover relations
       $HoRs \in \mathcal{B}(\mathcal{A} \times \mathcal{A})$
foreach $c \in C_{joint}$ do
    foreach $HoT \in HoTs$ do
        $HoT_c = \phi(HoT, (case, c))$; while $HoT_c \neq \emptyset$ do
            $hor = (\perp, \perp)$; $rec = min_{id}(HoT_c)$; if $\pi_{to}(rec) \neq \perp$ then
                $set_1(hor, \pi_{act}(rec))$;
            else
                $HoT_{from} \leftarrow$ get $HoT \in HoTs$ where $\exists_{rec' \in HoT} \pi_{org}(rec') = \pi_{from}(rec)$;
                $HoT_{from_c} = \phi(HoT_{from}, (case, c))$;
                $HoT^{to}_{from_c} = \phi(HoT_{from_c}, (to, \pi_{org}(rec)))$;
                $rec_{to} = min_{id}(HoT^{to}_{from_c})$;
                $set_1(hor, \pi_{act}(rec_{to}))$; remove $rec_{to}$ from $HoT_{from}$;
            end
            if $\pi_{from}(rec) \neq \perp$ then
                $set_2(hor, \pi_{act}(rec))$;
            else
                $HoT_{to} \leftarrow$ get $HoT \in HoTs$ where $\exists_{rec' \in HoT} \pi_{org}(rec') = \pi_{to}(rec)$;
                $HoT_{to_c} = \phi(HoT_{to}, (case, c))$;
                $HoT^{from}_{to_c} = \phi(HoT_{to_c}, (from, \pi_{org}(rec)))$;
                $rec_{from} = min_{id}(HoT^{from}_{to_c})$;
                $set_2(hor, \pi_{act}(rec_{from}))$; remove $rec_{from}$ from $HoT_{to}$;
            end
            remove $rec$ from $HoT$ and $HoT_c$; add $hor$ to $HoRs$;
        end
    end
end
return $HoRs$;
```

Definition 14 (Selections Over Handover Tables): Let $AT = \{id, case, org, act, from, to\}$ be the set of attribute names and $VL = \mathbb{N}_{>0} \cup \mathcal{C} \cup \mathcal{O} \cup \mathcal{A}$ be the universe of values for the attributes of handover tables. Also, let $d : AT \rightarrow 2^{VL}$ be a function that retrieves the domain of an attribute, e.g., $d(id) = \mathbb{N}_{>0}$, and $M = \{m : AT \rightarrow VL \mid \forall_{att \in dom(m)} m(att) \in d(att)\}$ be the set of functions mapping attribute names to values. Given $m \in M$, $\phi : 2^{\mathcal{HR}} \times m \rightarrow 2^{\mathcal{HR}}$ is a selection function such that given $HoT \subseteq \mathcal{HR}$ and $(att, val) \in m$, $\phi(HoT, (att, val)) = \{rec \in HoT \mid \pi_{att}(rec) = val\}$ retrieves a subset of handover records in HoT matching m. $min_{id}(HoT)$

Case ID	Activity	Timestamp
1	ER Registration (ERR)	01.01.2019-08:00:00
1	ER Triage (ERT)	01.01.2019-10:00:00
1	IV Antibiotics (IVA)	01.01.2019-10:30:00
2	ER Registration (ERR)	02.01.2019-08:30:00
2	ER Triage (ERT)	02.01.2019-10:00:00
3	ER Registration (ERR)	03.01.2019-09:00:00
3	ER Triage (ERT)	03.01.2019-10:30:00

(a) Emergency Care (EC)

Case ID	Activity	Timestamp
1	Registration (REG)	02.01.2019-08:10:00
1	CRP	02.01.2019-10:10:00
1	Release (REL)	02.01.2019-10:40:00
2	Registration (REG)	03.01.2019-08:50:00
2	Sepsis Triage (SET)	03.01.2019-09:50:00
2	Release (REL)	03.01.2019-10:10:00
3	Registration (REG)	04.01.2019-08:40:00
3	Sepsis Triage (SET)	04.01.2019-09:40:00
3	Release (REL)	04.01.2019-10:40:00

(b) Sepsis Treatment (ST)

Case ID	Activity	Timestamp
1	Doctor Consultation (DCO)	03.01.2019-08:20:00
1	Prescription (PRE)	03.01.2019-10:20:00
2	Doctor Consultation (DCO)	03.01.2019-16:30:00
3	Doctor Consultation (DCO)	05.01.2019-08:10:00
3	Prescription (PRE)	05.01.2019-10:20:00
3	Doctor Consultation (DCO)	05.01.2019-09:10:00

(c) Specialty Checkup (SC)

FIGURE 5. Event logs of a chained execution scenario where a case (patient) arrives at the emergency care, receives a sepsis treatment, and finally gets a special checkup.

ID	Case ID	Organization	Handover Activity	Handover From	Handover To
1	1	EC	IVA	$\perp$	ST
2	2	EC	ERT	$\perp$	ST
3	3	EC	ERT	$\perp$	ST

(a) Emergency Care (EC)

ID	Case ID	Organization	Handover Activity	Handover From	Handover To
1	1	ST	REG	EC	$\perp$
2	1	ST	REL	$\perp$	SC
3	2	ST	REG	EC	$\perp$
4	2	ST	REL	$\perp$	SC
5	3	ST	REG	EC	$\perp$
6	3	ST	REL	$\perp$	SC

(b) Sepsis Treatment (ST)

ID	Case ID	Organization	Handover Activity	Handover From	Handover To
1	1	SC	DCO	ST	$\perp$
2	2	SC	DCO	ST	$\perp$
3	3	SC	DCO	ST	$\perp$

(c) Specialty Checkup (SC)

FIGURE 6. Handover tables correspond to the event logs shown in Figure 5.

and $max_{id}(HoT)$ retrieve the records with the minimum and maximum id in HoT, respectively.

The retrieving process can be started from the first record of any case in any of the handover tables. Note that handover records in handover tables are ordered based on the timestamps of the handover activities. The handover activity of the record is considered as the *handover from (to) activity* of a handover relation if *handover from (to) organization* of the record is specified. The missing handover activity of the handover relation is retrieved by referring to the first corresponding record of the case in the handover table of the organization specified in the *handover from (to) organization* of the starting record. The processed records are removed from the handover tables of organizations and this process continues until all the handover tables become empty.

For example, the retrieving process for a handover relation w.r.t. the handover tables shown in Figure 6 is as follows. Starting the process from case 1 in the handover table of EC (Figure 6 (a)), since *handover to organization* is specified, IVA is considered as the first handover activity, i.e., *handover to activity*. To retrieve the second handover activity, i.e., *handover from activity*, first, the handover table of the organization specified in the *handover to organization* is obtained (the handover table of ST). In the handover table of ST (Figure 6 (b)), all the records of case 1 are obtained. The *handover activity* of the first record of these records where the *handover from organization* is EC (i.e., REG) is considered as the second handover activity.

The retrieved handover relations need to be added to the merged DFRs obtained through sharing DFRs by each individual organization. Figure 7 shows the overall abstraction merging process to obtain the original DFRs including handover relations. In the following, we demonstrate the *update operation* for the different types of interoperability.

D. THE UPDATE OPERATION FOR CHAINED EXECUTION

We first demonstrate the problem that arises by not sharing handover tables. Consider the event logs shown in Figure 5

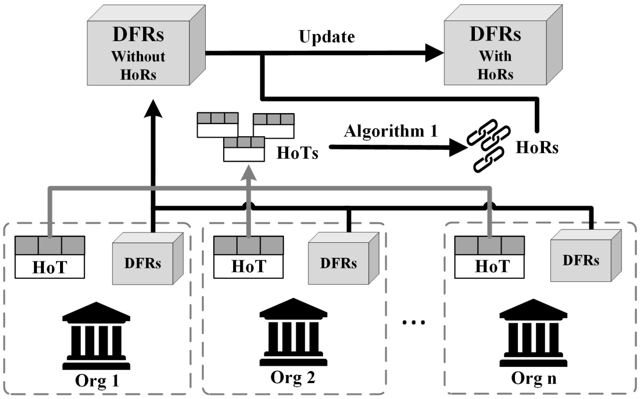


FIGURE 7. The overall process of merging abstractions (i.e., DFRs) to obtain the DFRs including handover relations. Inside the dashed squares is considered as the *trusted environment*, and outside these squares is considered as the *untrusted environment*.

as the event logs of a chained execution scenario. Figure 8 (a), (b), and (c) show the DFRs of these event logs. Figure 8 (d) shows the frequency annotated DFG obtained from the merged DFRs. One can see that the resulting graph does not reflect the real paths followed by the patients in the event logs. For example, the DFG includes three start activities, while ERR is the only start activity for all the patients. That is because handover relations have not been captured. For example, there are handover relations between REL in ST and DOC in SC having been replaced with $(REL, ■)$ and $(▶, DCO)$. Such missing relations can be retrieved by sharing *handover tables*.

Since we consider the *synchronous* type of communication (see Subsection II-C), in chain execution scenarios, handovers cannot happen in the middle of an intra-organizational trace. Thus, each retrieved handover relation is replaced with one *start relation* and one *end relation* matching with the handover relation. For example, (IVA, REG) is a handover relation retrieved by processing the first record of EC and ST in Figure 6. This relation is replaced with $(IVA, ■)$ and $(▶, REG)$. We call this specific type of update a *coupling update* which is defined as follows.

Definition 15 (Coupling Update): Let $L_1, L_2, \ldots, L_n$ be simple event logs belonging to n organizations involved in a process, $HoRs \in \mathcal{B}(\mathcal{A} \times \mathcal{A})$ be a multiset of handover relations among the organizations, and $mDFRs = abs_{DFR}(L_1) \uplus abs_{DFR}(L_2) \uplus \ldots \uplus abs_{DFR}(L_n)$ be the merged

DFRs
$(\blacktriangleright, ERR)$
(ERR, ERT)
(ERT, IVA)
$(IVA, \blacksquare)$
(ERR, ERT)
$(ERT, \blacksquare)$
$(\blacktriangleright, ERR)$
(ERR, ERT)
$(ERT, \blacksquare)$
$(\blacktriangleright, ERR)$

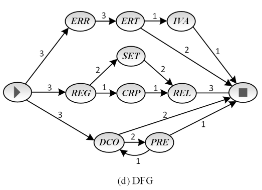
(a) Emergency Care (EC)

DFRs
$(\blacktriangleright, REG)$
(REG, SET)
(SET, REL)
$(REL, \blacksquare)$
(CRP, REL)
$(REL, \blacksquare)$
$(\blacktriangleright, REG)$
(REG, CRP)
$(REL, \blacksquare)$
$(\blacktriangleright, REG)$
(REG, SET)
(SET, REL)

(b) Sepsis Treatment (ST)

DFRs
$(\blacktriangleright, DCO)$
(DCO, PRE)
(PRE, DCO)
$(DCO, \blacksquare)$
$(DCO, \blacksquare)$
$(\blacktriangleright, DCO)$
(DCO, PRE)
$(PRE, \blacksquare)$
$(\blacktriangleright, DCO)$

(c) Specialty Checkup(SC)



(d) DFG

FIGURE 8. **The DFRs of the event logs shown in Figure 5, and the DFG obtained by merging them.**

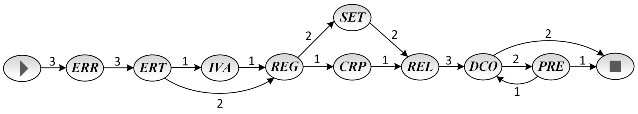


FIGURE 9. **The DFG obtained from the merged DFRs of the chained execution scenario after updating the DFRs with the missing handover relations.**

Case ID	Activity	Timestamp
1	ER Registration (ERR)	01.01.2019-08:00:00
1	ER Triage (ERT)	01.01.2019-10:00:00
1	IV Antibiotics (IVA)	01.01.2019-10:30:00
2	ER Registration (ERR)	02.01.2019-08:30:00
2	ER Triage (ERT)	02.01.2019-10:00:00
1	Doctor Consultation (DCO)	03.01.2019-08:20:00
1	Prescription (PRE)	03.01.2019-10:20:00
3	ER Registration (ERR)	03.01.2019-10:25:00
3	ER Triage (ERT)	03.01.2019-10:30:00
2	Doctor Consultation (DCO)	04.01.2019-16:30:00
3	Doctor Consultation (DCO)	05.01.2019-08:10:00
3	Prescription (PRE)	05.01.2019-10:20:00
3	Doctor Consultation (DCO)	05.01.2019-10:40:00

(a) Emergency Care (EC)

Case ID	Activity	Timestamp
1	Registration (REG)	02.01.2019-08:10:00
1	ESR	02.01.2019-10:10:00
1	Cytology (CYT)	02.01.2019-10:40:00
1	Results (RES)	02.01.2019-18:40:00
2	Registration (REG)	03.01.2019-08:50:00
2	Blood Test (BLT)	03.01.2019-09:50:00
2	ESR	03.01.2019-10:10:00
2	Results (RES)	03.01.2019-19:40:00
3	Registration (REG)	04.01.2019-08:40:00
3	Blood Test (BLT)	04.01.2019-09:40:00
3	Results (RES)	04.01.2019-20:00:00

(b) Laboratory (LB)

FIGURE 10. **Event logs of a subcontracting scenario where a patient arrives at emergency care, goes to a laboratory for certain tests and returns to the emergency care.**

DFRs. $update_{cp}(mDFRs, HoRs)$ updates $mDFRs$ based on $HoRs$ as follows:

$$update_{cp}(mDFRs, HoRs) = [dfr \in mDFRs \mid \forall_{dfr_1, dfr_2 \in mDFRs} \nexists_{hor \in HoRs}(\pi_1(dfr_1) = \pi_1(hor) \wedge \pi_2(dfr_1) = \blacksquare) \wedge (\pi_1(dfr_2) = \blacktriangleright \wedge \pi_2(dfr_2) = \pi_2(hor))] \uplus HoRs$$

Figure 9 shows the DFG after updating the merged DFRs with the missing handover relations. All the redundant start/end relations have been removed, and the DFG reflects a chained execution scenario.

E. THE UPDATE OPERATION FOR SUBCONTRACTING

In the subcontracting type of interoperability, an organization hands over a part of the process to a sub-organization or a third party. In this type of interoperability, handovers can happen within the middle of an intra-organizational trace. Note that we still assume the synchronous type of communication, i.e., a case's process can not be continued in the main organization while it runs in a sub-organization. An example of this type of interoperability in the healthcare context is a patient registered at emergency care, goes to a laboratory for certain tests and returns to the emergency care.

Consider the event logs shown in Figure 10 as the event logs for such a scenario. Figure 11 shows the handover tables for this scenario, and Figure 12 (a) and (b) show the DFRs.

ID	Case ID	Organization	Handover Activity	Handover From	Handover To
1	1	EC	IVA	$\perp$	LB
2	2	EC	ERT	$\perp$	LB
3	1	EC	DCO	LB	$\perp$
4	3	EC	ERT	$\perp$	LB
5	2	EC	DCO	LB	$\perp$
6	3	EC	DCO	LB	$\perp$

(a) Emergency Care (EC)

ID	Case ID	Organization	Handover Activity	Handover From	Handover To
1	1	LB	REG	EC	$\perp$
2	1	LB	RES	$\perp$	EC
3	2	LB	REG	EC	$\perp$
4	2	LB	RES	$\perp$	EC
5	3	LB	REG	EC	$\perp$
6	3	LB	RES	$\perp$	EC

(b) Laboratory (LB)

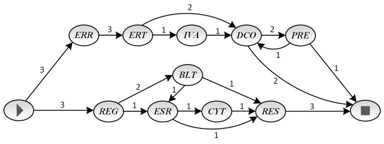
FIGURE 11. **The handover tables of the event logs shown in Figure 10.**

DFRs
$(\blacktriangleright, ERR)$
(ERR, ERT)
(ERT, IVA)
(IVA, DCO)
(DCO, PRE)
$(PRE, \blacksquare)$
$(\blacktriangleright, ERR)$
(ERR, ERT)
(ERT, DCO)
$(DCO, \blacksquare)$
$(DCO, \blacksquare)$
$(\blacktriangleright, ERR)$
(ERR, ERT)
(ERT, DCO)
(DCO, PRE)
(PRE, DCO)

(a) Emergency Care (EC)

DFRs
$(\blacktriangleright, REG)$
(REG, ESR)
(CYT, RES)
(ESR, CYT)
$(RES, \blacksquare)$
(REG, BLT)
$(RES, \blacksquare)$
$(\blacktriangleright, REG)$
(BLT, ESR)
(ESR, RES)
$(RES, \blacksquare)$
$(\blacktriangleright, REG)$
(REG, BLT)
(BLT, RES)

(b) Laboratory (LB)



(c) DFG

FIGURE 12. **The DFRs of the event logs shown in Figure 10, and the DFG obtained by merging them.**

Figure 12 (c) shows the frequency annotated DFG that is obtained from the merged DFRs without handover relations. One can see that due to missing handover relations, the resulting graph does not reflect the real paths followed by the patients in the event logs. For example, there are handover relations between IVA in EC and REG in LB and between RES in LB and DCO in EC that have been replaced with (IVA, DCO), $(\blacktriangleright, REG)$, and $(RES, \blacksquare)$.

The process of updating the merged DFRs using handovers is based on two main properties of the synchronous subcontracting scenarios: (P1) handovers happen within the middle of traces of the main organization that outsources part of the process, (P2) a case's process in a sub-organization starts by receiving the first handover relation from the main organization, and it ends by the last handover to the main organization. P1 implies that two handover relations need to be replaced with one directly follows relation, and P2 shows that start and end relations in a sub-organization need to be removed. For example, (IVA, REG) and (RES, DCO) are the handover relations retrieved by processing handover tables shown in Figure 11. These relation are replaced with $(RES, \blacksquare)$ and $(\blacktriangleright, REG)$, and (IVA, DCO) in the merged DFRs. We call this specific type of update a *decoupling update* which is defined as follows.

Definition 16 (Decoupling Update): Let $L_1, L_2, \ldots, L_n$ be simple event logs belonging to n organizations involved in a process, $HoRs \in \mathcal{B}(\mathcal{A} \times \mathcal{A})$ be a multiset of handover relations among the organizations, and $mDFRs = abs_{DFR}(L_1) \uplus abs_{DFR}(L_2) \uplus \ldots \uplus abs_{DFR}(L_n)$ be the merged DFRs. $update_{dcp}(mDFRs, HoRs)$ updates $mDFRs$ based on $HoRs$ as follows:

$$update_{dcp}(mDFRS, HoRs) = [dfr \in mDFRs \mid \forall_{hor_1, hor_2 \in HoRs} \nexists_{dfr_1, dfr_2, dfr_3 \in mDFRs}(\pi_1(dfr_1) = \pi_1(hor_1) \wedge \pi_2(dfr_1) = \pi_2(hor_2)) \wedge dfr_2 = (\blacktriangleright, \pi_2(hor_1)) \wedge dfr_3 = (\pi_1(hor_2), \blacksquare)] \uplus HoRs$$

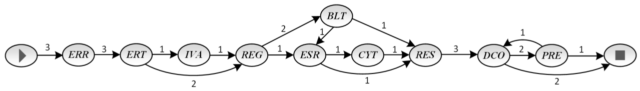


FIGURE 13. The DFG obtained from the merged DFRs of the subcontracting scenario after updating the DFRs with the missing handover relations.

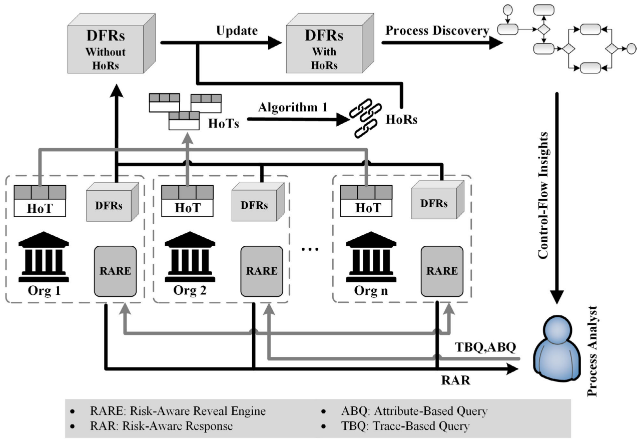


FIGURE 14. The general overview of our abstraction-based approach for PAFPM. Inside the dashed squares is considered as the *trusted environment*, and outside these squares is considered as the *untrusted environment*.

Figure 13 shows the DFG after updating the merged DFRs with missing handover relations. As can be seen, ERR is the only start activity, and the activities of LB appear in between the activities of EC.

F. THE UPDATE OPERATION CASE TRANSFER, CAPACITY SHARING, AND LOOSELY COUPLED

In this subsection, we focus on the *case transfer*, *capacity sharing*, and *loosely coupled* types of interoperability. In all three scenarios, organizations can take part in the events of a process in random order. Thus, the main difference between these scenarios and the ones discussed in the previous subsections is that handovers resulting from these types of interoperability happen randomly, and they do not follow any specific rule.

Random handovers mean that we cannot follow a specific rule to update merged DFRs with missing handovers. Hence, extra information needs to be provided by the organizations involved in the process. In particular, each organization needs to specify the DFRs involved in handovers. A DFR is involved in handovers if its non-dummy activities are involved in handovers. For instance, in the event logs shown in Figure 10, (IVA, DCO) is indicated as a DFR involved in handovers because IVA is involved in a handover from EC to LB, and DCO is involved in a handover from LB to EC. $(\blacktriangleright, REG)$ is also involved in handovers because REG, as a non-dummy activity, is involved in a handover from EC to LB.

Definition 17 (DFRs With Handover Indicators): Let $(\mathcal{A}\cup\{\blacktriangleright\} \times \mathcal{A}\cup\{\blacksquare\}) \times \{0,1\}$ be the universe of DRFs with handover indicators, where DFRs involved in a handover are indicated with 1 and the others are indicated with 0. $abs_{DFR_h} : \mathcal{B}(\mathcal{A}^*) \rightarrow \mathcal{B}((\mathcal{A}\cup\{\blacktriangleright\} \times \mathcal{A}\cup\{\blacksquare\}) \times \{0,1\})$ is an abstraction function that maps a given simple event log onto a multiset of DFRs with handover indicators. Given $dfr_h = ((a,b),i) \in (\mathcal{A}\cup\{\blacktriangleright\} \times \mathcal{A}\cup\{\blacksquare\}) \times \{0,1\}$, $\pi_{dfr}(dfr_h) = (a,b)$ and $\pi_{hor}(dfr_h) = i$ are the projections of dfr_h onto the directly follows relation and the handover indicator, respectively.

Note that indicating handover DFRs does not reveal sensitive information. It only imposes extra effort on the organizations, yet, at the same time, it drastically simplifies the process of updating merged DFRs with missing handover relations. Definition 18 demonstrates the process of updating merged DFRs with handover relations, where the DFRs involved in handovers are indicated.

Definition 18 (Update): Let $L_1, L_2, \ldots, L_n$ be simple event logs belonging to n organizations involved in a process, $HoRs \in \mathcal{B}(\mathcal{A} \times \mathcal{A})$ be a multiset of handover relations among the organizations, and $mDFRs = abs_{DFR_h}(L_1) \uplus abs_{DFR_h}(L_2) \uplus \ldots \uplus abs_{DFR_h}(L_n)$ be the merged DFRs. $update(mDFRs, HoRs)$ updates $mDFRs$ based on $HoRs$ as follows:

$$update(mDFRs, HoRs) = [\pi_{dfr}(dfr_h) \mid dfr_h \in mDFRs \wedge \pi_{hor}(dfr_h) = 0] \uplus HoRs$$

The general idea of updating merged DFRs is to add missing handover relations and remove the wrong DFRs added because of unknown interconnections among organizations. In Definition 15 and Definition 16, we exploited some properties of the interconnections to update merged DFRs with the minimum available information. However, when there is no specific property for interconnections, we utilize DFRs with handover indicators to update DFRs. Since the DFRs involved in handovers are indicated, one can simply remove all of them and add missing handover relations.

G. RARM FOR FEDERATED PROCESS MINING

In this subsection, we demonstrate the risk-aware reveal method for answering more in-depth questions about a process. We explain the general approach for two main types of queries, i.e., *attribute-based* and *trace-based*.

1) ATTRIBUTE-BASED QUERIES

A query is sent to the risk-aware reveal engines of all the involved organizations. RARE of the respective organization

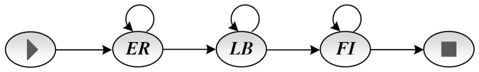


FIGURE 15. The DFG at the department level for the chained execution type of interoperability based on three departments in the Sepsis event log.

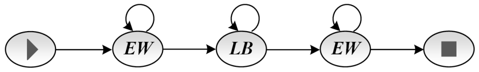


FIGURE 16. The DFG at the department level for the subcontracting type of interoperability based on two departments in the Sepsis event log.

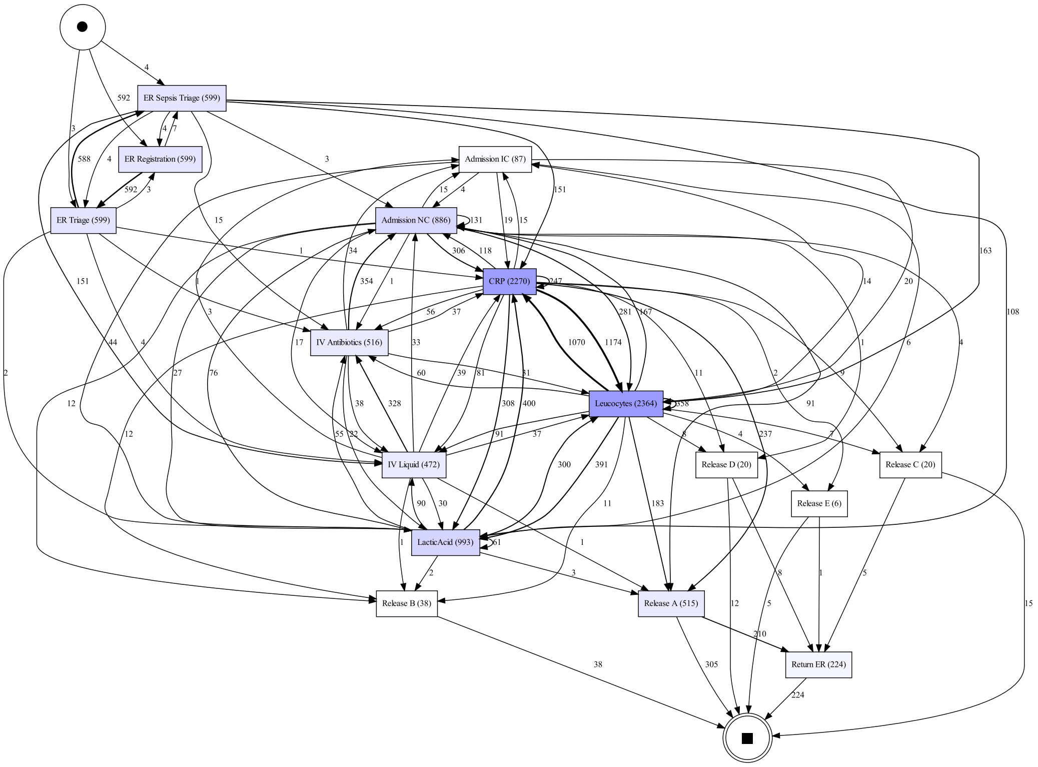


FIGURE 17. The original DFG for the main event log (Sepsis-CE/SB).

verifies the sensitivity of the requested attribute and either provides an answer or refuses the query. For example, consider the chained execution type of interoperability, the event logs shown in Figure 5, and the following query: *what are the activities performed on* 02.01.2019? The answers provided by EC, ST, and SC are as follows: $Res_{EC} = [ERR, ERT]$, $Res_{ST} = [REG, CRP, REL]$, and $Res_{SC} = []$. The sum of these multisets provides the aggregated response, i.e., $Res = Res_{EC} \uplus Res_{ST} \uplus Res_{SC}$.

Definition 19 (Attribute-Based Response): Let $\mathcal{O}$ be the universe of organizations, and $O \subseteq \mathcal{O}$ be the set of organizations involved in a joint process. A response provided by an organization $o \in O$ for a query regarding an attribute value $\mathcal{V} \subseteq \mathcal{C} \cup \mathcal{A} \cup \mathcal{T} \cup \mathcal{R} \cup \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_m$ is a multiset $Res_o \in \mathcal{B}(\mathcal{V})$. The aggregated response is $Res = \biguplus_{o \in O} Res_o$.

2) TRACE-BASED QUERIES

To get a complete response for the trace-based queries, a process analyst may need to send several queries to different organizations in a specific order depending on the responses received from each single organization. Similar to the attribute-based queries, a query is first sent to the risk-aware reveal engines of all the involved organizations. RARE of each organization verifies the risk associated with a response. If the risk is above a predefined threshold, the RARE refuses the query. Otherwise, it provides a response with the corresponding handover tables for the cases whose data are included in the response. Such handover tables are utilized by the process analyst to get the possible missing pieces of the response from other organizations. A response provided for the trace-based queries by an organization with the identifier $o \in \mathcal{O}$ is a set $Res_o \subseteq \mathcal{C} \times \mathcal{A}^* \times 2^{\mathcal{HR}}$ (see Definition 22).

Consider the chained execution type of interoperability, the event logs shown in Figure 5, and the following query: *what are the traces of cases whose treatment process contains IVA?* Assuming that the risks are acceptable for the organizations, the reponses provided by EC, ST, and SC are as follows: $Res_{EC} = \{(1, \langle ERR, ERT, IVA \rangle, \{(1, 1, EC, IVA, \bot, ST)\})\}$, $Res_{ST} = \emptyset$, and $Res_{SC} = \emptyset$. Using the handover table of EC, the process analyst realizes that another query needs to be sent to ST to obtain the trace of case 1. The response of such a query is as follows: $Res_{ST} = \{(1, \langle REG, CRP, REL \rangle, \{(1, 1, ST, REG, EC, \bot), (2, 1, ST, REL, \bot, SC)\})\}$. Verifying this response, the process analyst needs to send another query to SC to obtain the missing part of the trace. The response of such a query is as follows: $Res_{SC} = \{(1, \langle DCO, PRE \rangle, \{(1, 1, SC, DCO, ST, \bot)\})\}$.

The process of sending queries stops when both *launcher* and *terminator* organizations of cases in all the responses are visited. Given a case c, the launcher organization is the one that starts the process of the case c (Definition 20), and the terminator organization is the one that ends the process of the case c (Definition 21). For each case, the process analyst joins

TABLE 3. Our categorization for departments and their activities in the Sepsis event log.

Department	Activities
Emergency Room (ER)	ER Registration, ER Triage, ER Sepsis Triage
Laboratory (LB)	IV Liquid, IV Antibiotics, LacticAcid, CRP, Leucocytes, Admission NC, Admission IC
Financial (FI)	Release A, Release B, Release C, Release D, Release E, Return ER
Emergency Ward (EW) ER + FI	ER Registration, ER Triage, ER Sepsis Triage, Release A, Release B, Release C, Release D, Release E, Return ER

TABLE 4. The general statistics of the event logs used in the experiments.

Event Log	#Cases	#Variants	#Events	#Unique Activities
Sepsis-CE/SB	599	546	10208	16
Sepsis-CE-ER	599	4	1797	3
Sepsis-CE-FI	599	9	823	6
Sepsis-CE/SB-LB	599	514	7588	7
Sepsis-SB-EW	599	17	2620	9

the traces in the individual responses. The joining process for each case starts from the launcher organization and ends at the terminator organization. The complete trace regarding the only case involved in the above-mentioned example is $Res_1 = \langle ERR, ERT, IVA \rangle \oplus \langle REG, CRP, REL \rangle \oplus \langle DCO, PRE \rangle$.

Definition 20 (Launcher Organization): Let $c \in C_{joint}$ be a case, and $HoTs = \{HoT_1, HoT_2, \cdots, HoT_n\}$ be a collection of handover tables of the organizations involved in the process of case c. An organization with the handover table $HoT_i \in HoTs$ is the launcher organization of case c iff $\pi_{from}(min_{id}(\phi(HoT_i, (case, c)))) = \perp$.

Definition 21 (Terminator Organization): Let $c \in C_{joint}$ be a case, and $HoTs = \{HoT_1, HoT_2, \cdots, HoT_n\}$ be a collection of handover tables of the organizations involved in the process of case c. An organization with the handover table $HoT_i \in HoTs$ is the terminator organization of case c iff $\pi_{to}(max_{id}(\phi(HoT_i, (case, c)))) = \perp$.

Definition 22 (Trace-Based Response): Let $\mathcal{O}$ be the universe of organizations, $o_1, o_2, \ldots, o_n \subseteq \mathcal{O}$ be the organizations involved in the process of some joint cases $C_{joint} \subseteq \mathcal{C}$. A response provided for a trace-based query by an organization o_i, $1 \leq i \leq n$, is a set $Res_{o_i} \subseteq \mathcal{C} \times \mathcal{A}^* \times 2^{\mathcal{HR}}$, s.t., if $(c, \sigma, HoT) \in Res_{o_i}$, then for all $rec \in HoT$, $\pi_{case}(rec) = c$. The response provided regarding a case $c \in C_{joint}$ is $Res_c = \sigma_1 \oplus \sigma_2 \oplus \cdots \oplus \sigma_n$, where $(c, \sigma_i, HoT_i) \in Res_{o_i}$, o_1 is the launcher organization of case c, and o_n is the terminator organization of the case.

Figure 14 shows an overview of our approach for PAFPM for all the types of interoperability where cases can be shared among organizations. The general approach for all the mentioned types of interoperability is the same. The only difference is different update operations for merged DFRs based on the different types of interoperability. Note that the generic update definition (Definition 18) can be used for all the types of interoperability if the DFRs involved in handovers are indicated. The federated DFG obtained from DFRs

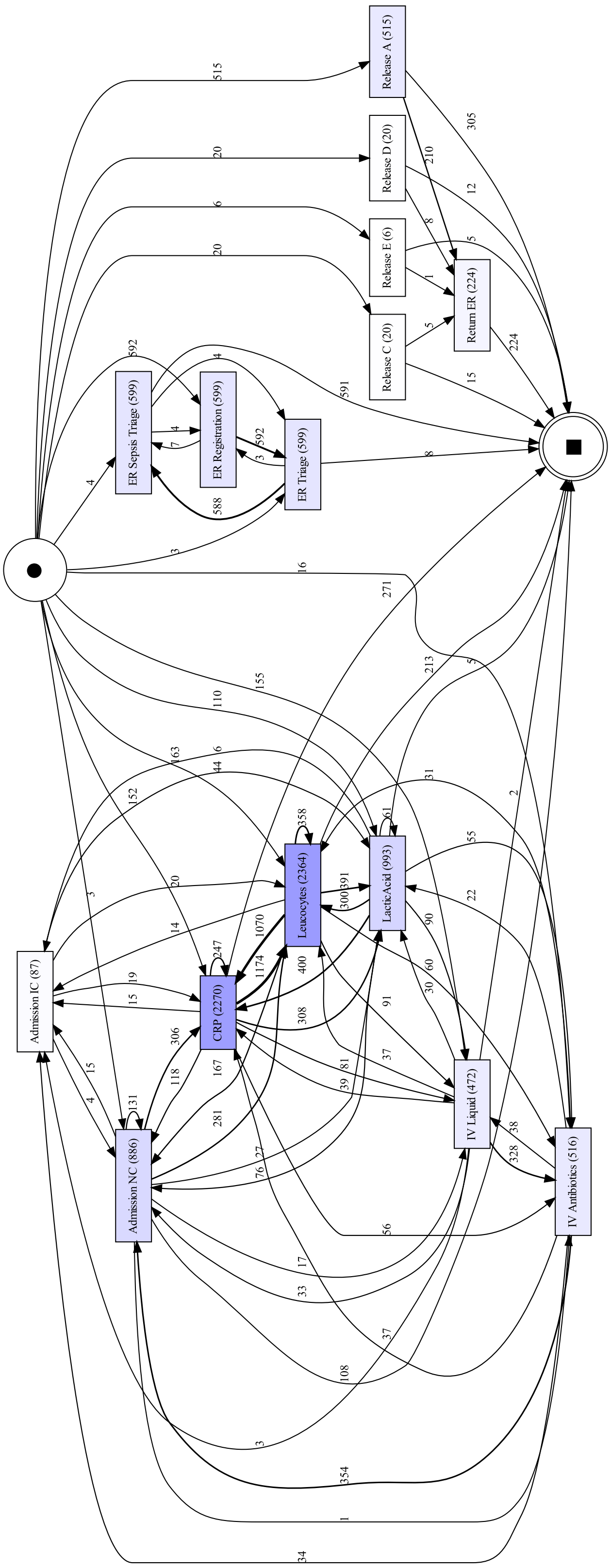


FIGURE 18. The DFG of the merged DFRs without handover relations for the chained execution scenario.

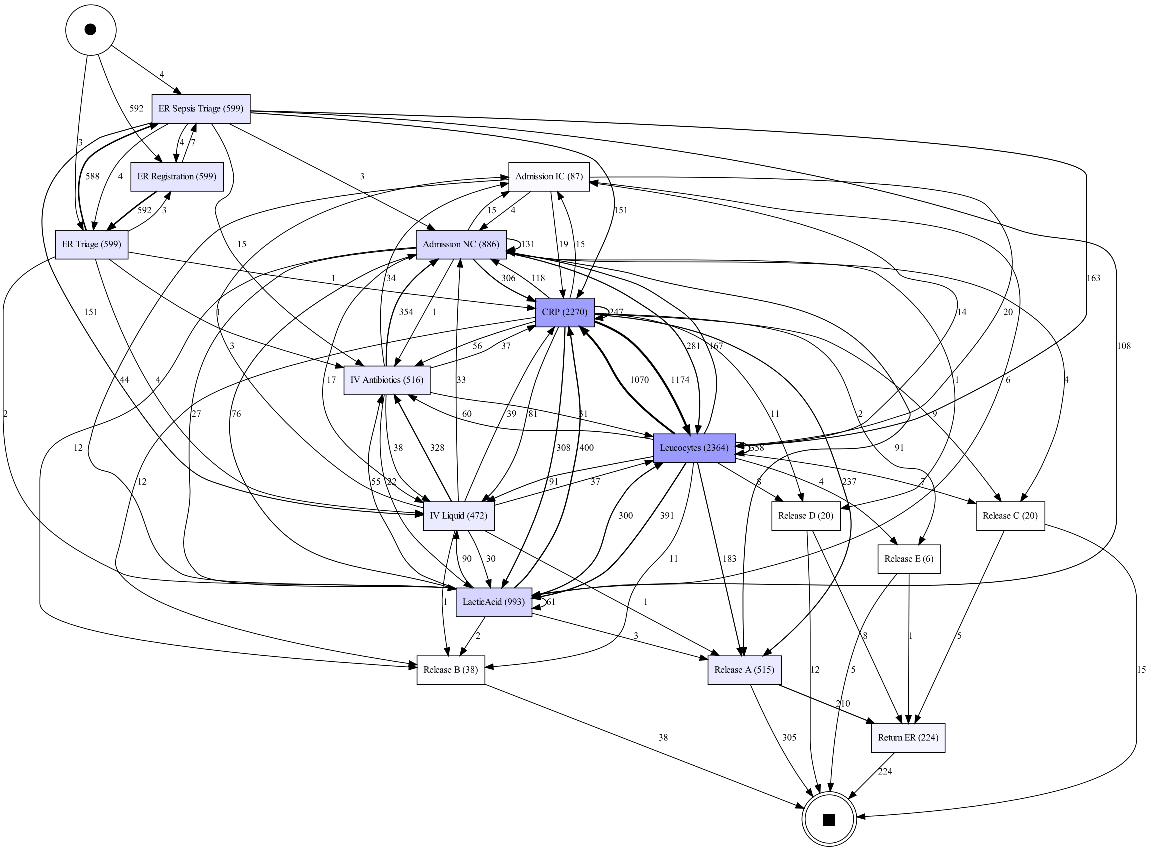


FIGURE 19. The DFG of the updated merged DFRs with handover relations for the chained execution scenario.

containing handovers can provide a high-level understanding of the joint process. However, to analyze more complicated aspects of the process, e.g., which activities are performed simultaneously, the process discovery step needs to be done.

VI. IMPLEMENTATION AND EVALUATION

In general, three criteria can be considered for evaluating different aspects of our approach: data utility, privacy, and inter-organizationality. In Section IV, we explained the privacy and data utility aspects of the abstraction used in this paper, i.e., DFRs. Since DFRs remove the concept of trace, they can mitigate the disclosure risks w.r.t. the control-flow aspect (see Subsection II-B). Nevertheless, as we explained, there are situations where the original trace variants of an event log can be revealed based on its DFG. We also demonstrated the data utility shortcomings of abstractions and introduced the risk-aware reveal method to overcome the shortcomings. Since the main focus of this paper is on inter-organizational process mining, in this section, we evaluate the inter-organizationality aspect that also incorporates the other aspects.

We employ Sepsis as a real-life event log for our experiments [46]. Sepsis is an event log recorded by an information system in a hospital that contains 15214 events and 16 unique activities performed for 1050 patients (cases). We demonstrated five different types of interoperability including *chained execution*, *subcontracting*, *case transfer*, *capacity sharing*, and *loosely coupled*. Assuming that handover tables are provided by the organizations involved in a process, the most challenging part of the abstraction-based approach is the update operation. The update operation for *case transfer*, *capacity sharing*, and *loosely coupled* relies on the information regarding handover indications in directly follows relations. Given such information, the update operation is a straightforward task. Thus, we mainly focus on the *chained execution* and *subcontracting* types of interoperability. We implemented a Python script for our evaluation. The implementation is available as a GitLab repository[1] and can be installed as a Python package.[2]

A. SCENARIO DISCOVERY

In this subsection, we demonstrate the process of discovering the *chained execution* and *subcontracting* types of interoperability from the Sepsis event log. As described in [47], Sepsis is an event log collected from three main departments: *Emergency Room* (ER), *Labratory* (LB), and *Financial* (FI). Table 3 shows our categorization for the activities in the Sepsis event log. Note that to avoid having uncategorized activities, our categorization for the activities is more general compared to the categories discussed in [47]. Namely, we

[1] https://git.rwth-aachen.de/majid.rafiei/pp-iopm/
[2] https://pypi.org/project/pp-iopm/

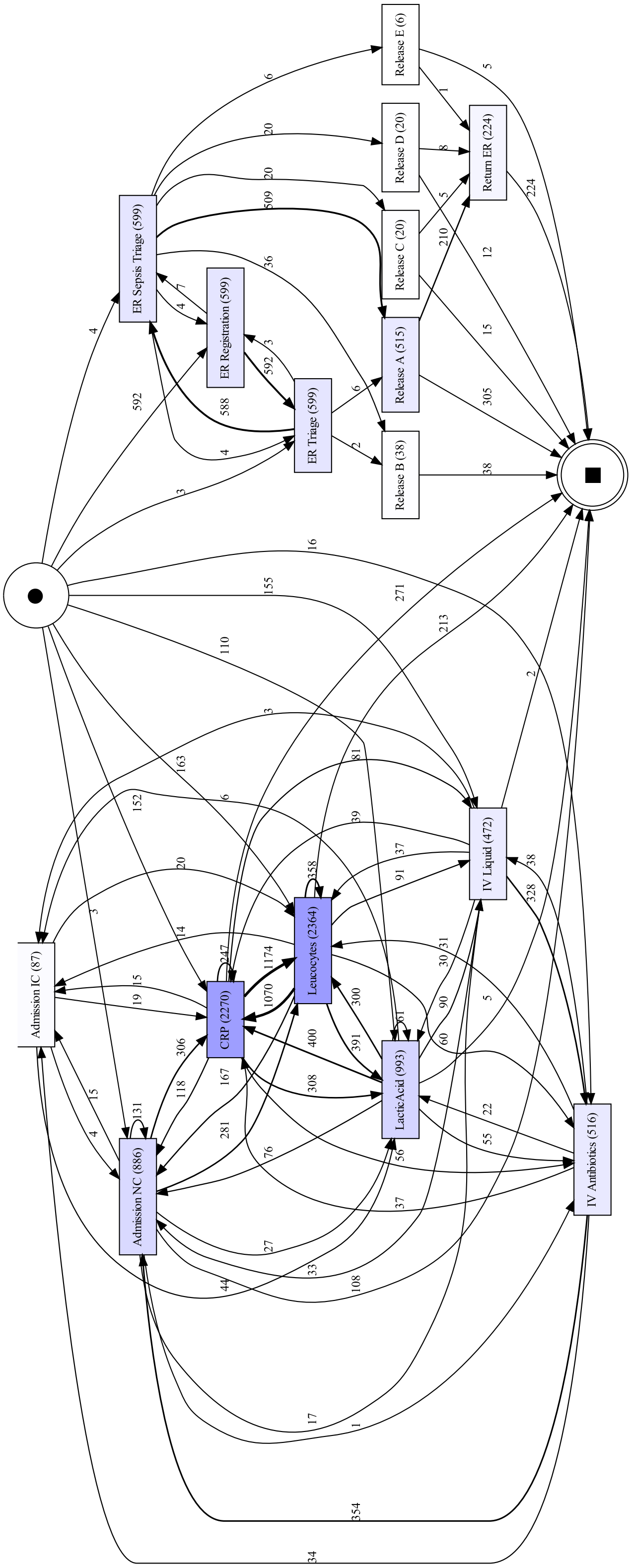


FIGURE 20. The DFG of the merged DFRs without handover relations for the subcontracting scenario.

consider "Return ER" as an activity performed by the financial department and admission activities performed by the laboratory.

To discover the chained execution scenario based on the Sepsis event log, we generalize the activities to their department level. A DFG discovered from the generalized event log shows the paths that patients (cases) follow at the department level. The set of cases following a path that corresponds to the department-level DFG shown in Figure 15 generates a sub-event-long from Sepsis that matches a chained execution scenario. We name this event log Sepsis-CE. By projecting Sepsis-CE onto the activities of each department, we get three event logs for three different departments. We name these event logs Sepsis-CE-ER, Sepsis-CE-LB, and Sepsis-CE-FI.

To discover a subcontracting scenario, we generalize ER and FI to one department, called Emergency Ward (EW). After this generalization, the same set of cases, as the ones in Sepsis-CE, follow a subcontracting scenario, i.e., the cases follow a path that corresponds to the department-level DFG shown in Figure 16. Although the set of cases is the same, for the sake of simplicity, we name this event log Sepsis-SB. By projecting Sepsis-SB onto the activities of each department, i.e., EW and LB, we get two event logs for two different departments. We name these event logs Sepsis-SB-EW and Sepsis-SB-LB. Note that Sepsis-SB-LB is also the same as Sepsis-CE-LB. Table 4 shows the general statistics of the event logs that we obtained for the above-mentioned scenarios.

B. THE MERGING CHALLENGE

In this subsection, we show the results of applying our approach to the event logs explained in the previous section. Figure 17 shows the original DFG for the main event log in both scenarios, i.e., Sepsis-CE/SB. As explained in Section V, the first step in all the scenarios is that each organization applies the abstraction function to its own private event log and shares the resulting DFRs.

Figure 18 shows the DFG of the merged DFRs without handover relations for the chained execution scenario. Since handover relations are missing, one can see three submodels with their own start and end activities, and there is no connection between the activities of different departments. In fact, the concept of chained execution has completely vanished. Based on our scenario for the chained execution type of interoperability, all the cases follow a path matching the DFG shown in Figure 15. Thus, we expect to see the activities of the ER department at the beginning and the activities of the FI department at the end (as shown in Figure 17). By applying Algorithm 1 to the handover tables of all the departments, we retrieved 1198 missing handover relations. Figure 19 shows the DFG of the merged DFRs updated with these

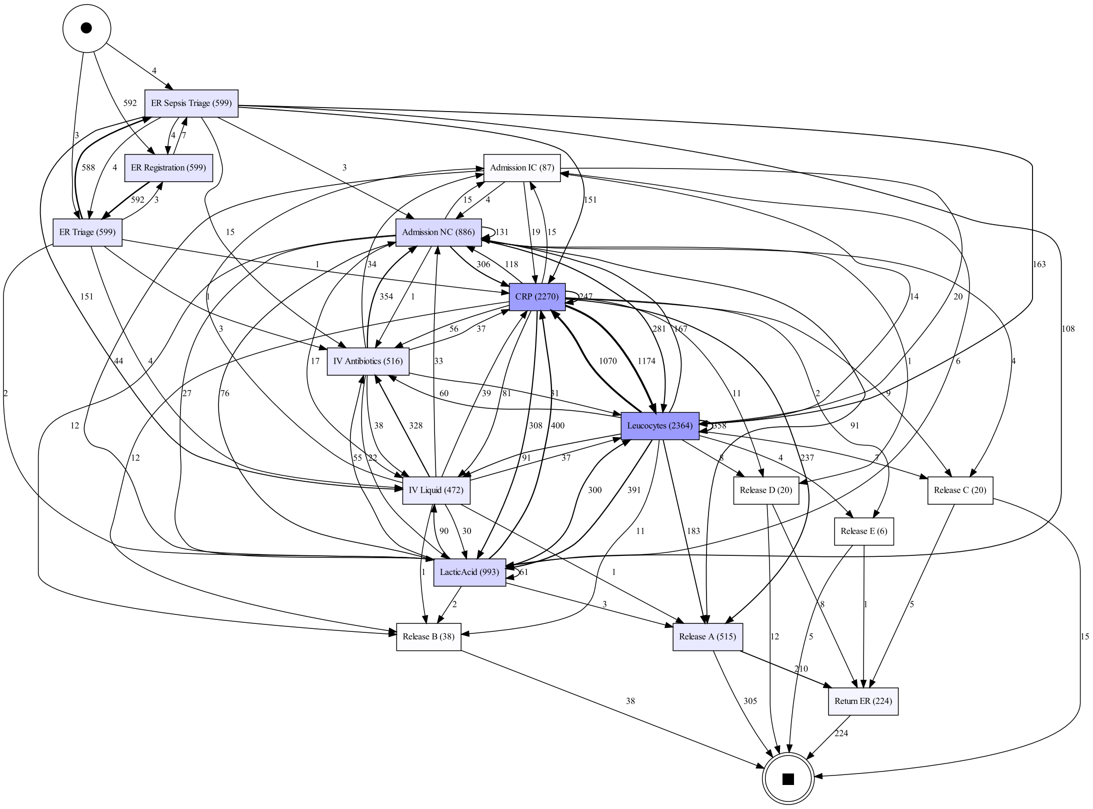


FIGURE 21. The DFG of the updated merged DFRs with handover relations for the subcontracting scenario.

missing handover relations exploiting the *coupling update* operation. This DFG is exactly the same as the original one.

Figure 20 shows the DFG of the merged DFRs without handover relations for the subcontracting scenario. One can see two submodels corresponding to two departments, EW and LB. There is no connection between the activities of the two departments, and the submodels have their own start and end activities. Based on our scenario for the subcontracting type of interoperability, the activities of the LB department should appear between the activities of EW. Similar to the chained execution scenario, we apply Algorithm 1 to retrieve the missing handover relations. Then, the *decoupling update* operation is applied to update the merged DFRs with the missing handovers. Figure 21 shows the DFG of the merged DFRs which is exactly the same as the original event log.

VII. CONCLUSION AND DISCUSSION

In this paper, we proposed an abstraction-based approach for privacy-aware federated process mining. We employed DFRs as abstractions of event logs. We introduced the risk-aware reveal method to overcome its data utility limitations. In Section V, we specialized our approach to federated process mining for five different interoperability scenarios. We introduced the concept of *handover relations* and *handover tables* and demonstrated an algorithm for retrieving missing handover relations in an inter-organizational setting. We also demonstrated update operations to update directly follows relations with missing handover relations. We employed Sepsis as a real-life event log to evaluate our approach for reproducible scenarios.

In our problem setting, we assumed that privacy concerns are at the level of individuals, i.e., traces are sensitive information that need to be protected. However, our approach can also support the department level of sensitive information. If we assume that the entire internal activities of an organization are private, the organization can share only its handover table. As a result, the generated DFG in the untrusted environment only represents the communication points of the organization.

For explaining the risk-aware reveal method, we focused on an intuitive type of attack and the corresponding disclosure risks. However, attack scenarios and their corresponding risk analysis can be done more extensively. Since organizations are not aware of the event logs and risk thresholds required by other organizations, they may provide responses that violate the risk requirements of one another. Namely, *intersection-based* attacks can be launched [15]. For example, consider the following scenario. In organization o_1, the age attribute is considered as a sensitive attribute and it gets generalized before publishing. However, in organization o_2, age is not considered as a sensitive attribute and it is shared without generalization. If there exists only one case in a specific range in the response provided by o_2, the privacy requirement of o_1 is violated.

Moreover, provided privacy guarantees can be degraded by integrating individual responses. Consider a scenario where there are two organizations o_1 and o_2 that provide a trace-based response including three cases c_1, c_2, and c_3. Assume these responses to be as follows: $Res_{o_1} = \{(c_1, \langle a, b, c\rangle, \{(1, c_1, o_1, c, \perp, o_2)\}), (c_2, \langle a, b\rangle, \{(1, c_2, o_1, b, \perp, o_2)\}), (c_3, \langle b, c\rangle, \{(1, c_3, o_1, c, \perp, o_2)\})\}$, $Res_{o_2} = \{(c_1, \langle d, e, f\rangle, \{(1, c_1, o_2, d, o_1, \perp)\}), (c_2, \langle d, e\rangle, \{(1, c_2, o_2, d, o_1, \perp)\}), (c_3, \langle e, f\rangle, \{(1, c_3, o_2, e, o_1, \perp)\})\}$.

Each individual response contains more than one case considering a sequence of activities with a maximum length of 2 as the background knowledge. However, the integrated responses for cases are as follows:

$Res_{c_1} = \langle a, b, c, d, e, f\rangle$, $Res_{c_2} = \langle a, b, d, e\rangle$, $Res_{c_3} = \langle b, c, e, f\rangle$. As can be seen, there are sequences of activities with length 2 that single out a case. For example, $\langle c, e\rangle$ singles out the case $c3$, or $\langle c, d\rangle$ singles out the case c_1.

Such risks can be mitigated using an integration engine that considers the risk thresholds of all the organizations and re-evaluates the risks associated with integrated responses before exposing them to the untrusted environment. Such an integration engine can be considered as a semi-trusted third party that never gets unprotected information and may not misbehave. Nevertheless, we still need to realize third-party independent solutions for such scenarios.

Moreover, the current risk-aware reveal engine employs no privacy-preserving technique. It solely analyzes the risks associated with a single response and either refuses the corresponding request or shares the response. In the future, the engine can be equipped with privacy preservation techniques, e.g., *differential privacy*, that provide privacy guarantees for responses. The engine is also stateless, i.e., it does not keep the track of queries. In the future, the engine can be upgraded to a stateful one that tracks queries to avoid privacy leakage resulting from responses provided to several queries.

We described our approach for the synchronous type of communication. However, the approach can also support the asynchronous type of communication using modeling techniques that can represent concurrencies. In the future, we plan to extend this technique with other representation models to support the asynchronous type of communication. We also plan to perform a case study to comprehensively evaluate the effectiveness of the risk-aware reveal method.

ACKNOWLEDGMENT

The authors would like to thank the Alexander von Humboldt Stiftung for supporting their research.

REFERENCES

[1] W. M. P. van der Aalst, *Process Mining Data Science in Action*, 2nd ed. Cham, Switzerland: Springer, 2016.

[2] W. M. P. van der Aalst, "Intra- and inter-organizational process mining: Discovering processes within and between organizations," in *Proc. IFIP Working Conf. Pract. Enterprise Modeling* in Lecture Notes in Business Information Processing, vol. 92, P. Johannesson, J. Krogstie, and A. L. Opdahl, Eds. Cham, Switzerland: Springer, 2011, pp. 1–11, doi: 10.1007/978-3-642-24849-8_1.

[3] R. Engel, W. M. P. van der Aalst, M. Zapletal, C. Pichler, and H. Werthner, "Mining inter-organizational business process models from EDI messages: A case study from the automotive sector," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* in Lecture Notes in Computer Science, vol. 7328, J. Ralyté, X. Franch, S. Brinkkemper, and S. Wrycza, Eds. Cham, Switzerland: Springer, 2012, pp. 222–237, doi: 10.1007/978-3-642-31095-9_15.

[4] O. Yilmaz and P. Karagoz, "Generating performance improvement suggestions by using cross-organizational process mining," in *Proc. 5th Int. Symp. Data-Driven Process Discovery Anal. (SIMPDA)*, vol. 1527, P. Ceravolo and S. Rinderle-Ma, Eds. Vienna, Austria, Dec. 2015, pp. 3–17. [Online]. Available: http://ceur-ws.org/Vol-1527/paper1.pdf

[5] W. M. P. van der Aalst, "Federated process mining: Exploiting event data across organizational boundaries," in *Proc. IEEE Int. Conf. Smart Data Services (SMDS)*, Sep. 2021, pp. 1–7, doi: 10.1109/SMDS53860.2021.00011.

[6] W. M. P. van der Aalst, "Process discovery from event data: Relating models and logs through abstractions," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 3, May 2018, Art. no. e1244.

[7] M. Rafiei and W. M. P. van der Aalst, "Towards quantifying privacy in process mining," in *Proc. Int. Conf. Process Mining* in Lecture Notes in Business Information Processing, S. J. J. Leemans and H. Leopold, Eds. Cham, Switzerland: Springer, 2020, pp. 385–397, doi: 10.1007/978-3-030-72693-5_29.

[8] S. Tönnissen and F. Teuteberg, "Using blockchain technology for cross-organizational process mining—Concept and case study," in *Proc. Int. Conf. Bus. Inf. Syst.* in Lecture Notes in Business Information Processing, vol. 354, W. Abramowicz and R. Corchuelo, Eds. Cham, Switzerland: Springer, 2019, pp. 121–131, doi: 10.1007/978-3-030-20482-2_11.

[9] L. Maruster, J. C. Wortmann, A. J. M. M. Weijters, and W. M. P. van der Aalst, "Discovering distributed processes in supply chains," in *Collaborative Systems for Production Management*, vol. 257, H. Jagdev, J. C. Wortmann, and H. J. Pels, Eds. Eindhoven, The Netherlands: Kluwer, 2002, pp. 219–230.

[10] R. Liu, A. Kumar, and W. van der Aalst, "A formal modeling approach for supply chain event management," *Decis. Support Syst.*, vol. 43, no. 3, pp. 761–778, Apr. 2007, doi: 10.1016/j.dss.2006.12.009.

[11] K. Gerke, J. Mendling, and K. Tarmyshov, "Case construction for mining supply chain processes," in *Proc. Int. Conf. Bus. Inf. Syst.* in Lecture Notes in Business Information Processing, vol. 21, W. Abramowicz, Ed. Poznan, Poland: Springer, 2009, pp. 181–192, doi: 10.1007/978-3-642-01190-0_16.

[12] W. M. P. van der Aalst, "Responsible data science: Using event data in a 'people friendly' manner," in *Proc. Int. Conf. Enterprise Inf. Syst.* in Lecture Notes in Business Information Processing, vol. 291, S. Hammoudi, L. A. Maciaszek, M. Missikoff, O. Camp, and J. Cordeiro, Eds. Cham, Switzerland: Springer, Apr. 2016 Lecture Notes in Business Information Processing, 2016, pp. 3–28.

[13] F. Mannhardt, S. A. Petersen, and M. F. Oliveira, "Privacy challenges for process mining in human-centered industrial environments," in *Proc. 14th Int. Conf. Intell. Environ. (IE)*, Jun. 2018, pp. 64–71.

[14] A. Pika, M. T. Wynn, S. Budiono, A. H. M. ter Hofstede, W. M. P. van der Aalst, and H. A. Reijers, "Towards privacy-preserving process mining in healthcare," in *Proc. Int. Conf. Bus. Process Manag.* Lecture Notes in Business Information Processing, vol. 362, C. D. Francescomarino, R. M. Dijkman, and U. Zdun, Eds. Cham, Switzerland: Springer, 2019, pp. 483–495.

[15] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. F. Sani, A. Koschmider, F. Mannhardt, S. N. von Voigt, M. Rafiei, and L. von Waldthausen, "Privacy and confidentiality in process mining: Threats and research challenges," *ACM Trans. Manag. Inf. Syst.*, vol. 13, no. 1, pp. 11:1–11:17, 2022, doi: 10.1145/3468877.

[16] F. Mannhardt, A. Koschmider, N. Baracaldo, M. Weidlich, and J. Michael, "Privacy-preserving process mining - differential privacy for event logs," *Bus. Inf. Syst. Eng.*, vol. 61, no. 5, pp. 595–614, 2019.

[17] S. A. Fahrenkrog-Petersen, H. van der Aa, and M. Weidlich, "PRIPEL: Privacy-preserving event log publishing including contextual information," in *Proc. Int. Conf. Bus. Process Manag.* in Lecture Notes in Computer Science, vol. 12168, D. Fahland, C. Ghidini, J. Becker, and M. Dumas, Eds. Cham, Switzerland: Springer, 2020, pp. 111–128, doi: 10.1007/978-3-030-58666-9_7.

[18] M. Rafiei, M. Wagner, and W. M. P. van der Aalst, "TLKC-privacy model for process mining," in *Proc. Int. Conf. Res. Challenges Inf. Sci.* in Lecture Notes in Business Information Processing, vol. 385, F. Dalpiaz, J. Zdravkovic, and P. Loucopoulos, Eds. Cham, Switzerland: Springer, 2020, pp. 398–416.

[19] S. A. Fahrenkrog-Petersen, H. van der Aa, and M. Weidlich, "PRETSA: Event log sanitization for privacy-aware process discovery," in *Proc. Int. Conf. Process Mining (ICPM)*, Jun. 2019, pp. 1–8.

[20] E. Batista and A. Solanas, "A uniformization-based approach to preserve individuals' privacy during process mining analyses," *Peer Peer Netw. Appl.*, vol. 14, no. 3, pp. 1500–1519, May 2021, doi: 10.1007/s12083-020-01059-1.

[21] G. Elkoumy, A. Pankova, and M. Dumas, "Mine me but don't single me out: Differentially private event logs for process mining," in *Proc. 3rd Int. Conf. Process Mining (ICPM)*, Oct. 2021, pp. 80–87, doi: 10.1109/ICPM53251.2021.9576852.

[22] S. N. 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—Empiricial evaluation paper," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* in Lecture Notes in Computer Science, vol. 12127, S. Dustdar, E. Yu, C. Salinesi, D. Rieu, and V. Pant, Eds. Grenoble, France: Springer, Jun. 2020, pp. 252–267.

[23] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[24] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discovery From Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[25] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Apr. 2007, pp. 106–115, doi: 10.1109/ICDE.2007.367856.

[26] M. Rafiei and W. M. P. van der Aalst, "Group-based privacy preservation techniques for process mining," *Data Knowl. Eng.*, vol. 134, Jul. 2021, Art. no. 101908, doi: 10.1016/j.datak.2021.101908.

[27] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, vol. 4978. Berlin, Germany: Springer, 2008, pp. 1–19, doi: 10.1007/978-3-540-79228-4_1.

[28] G. Elkoumy and M. Dumas, "Libra: High-utility anonymization of event logs for process mining via subsampling," 2022, *arXiv:2206.13050*.

[29] M. Rafiei, F. Wangelik, and W. M. P. van der Aalst, "Travas: Differentially private trace variant selection for process mining," in *Proc. Process Mining Workshops*. Cham, Switzerland: Springer, 2023, pp. 114–126, doi: 10.1007/978-3-031-27815-0_9.

[30] M. Rafiei, L. von Waldthausen, and W. M. P. van der Aalst, "Supporting confidentiality in process mining using abstraction and encryption," in *Proc. Int. Symp. Data-Driven Process Discovery Anal.* in Lecture Notes in Business Information Processing, vol. 379, P. Ceravolo, M. van Keulen, and M. T. G. López, Eds. Bled, Slovenia: Springer, Sep. 2019, pp. 101–123.

[31] A. Burattin, M. Conti, and D. Turato, "Toward an anonymous process mining," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, Aug. 2015, pp. 58–63.

[32] J. Michael, A. Koschmider, F. Mannhardt, N. Baracaldo, and B. Rumpe, "User-centered and privacy-driven process mining system design for IoT," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* in Lecture Notes in Business Information Processing, vol. 350, C. Cappiello and M. Ruiz, Eds. Rome, Italy: Springer, Jun. 2019, pp. 194–206.

[33] M. Rafiei and W. M. P. van der Aalst, "Mining roles from event logs while preserving privacy," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* in Lecture Notes in Business Information Processing, vol. 362, C. D. Francescomarino, R. M. Dijkman, and U. Zdun, Eds. Vienna, Austria: Springer, Sep. 2019, pp. 676–689.

[34] M. Rafiei and W. M. P. van der Aalst, "Privacy-preserving continuous event data publishing," in *Proc. Int. Conf. Bus. Process Manag.* in Lecture Notes in Business Information Processing, vol. 427, A. Polyvyanyy, M. T. Wynn, A. V. Looy, and M. Reichert, Eds. Rome, Italy: Springer, Sep. 2021, pp. 178–194, doi: 10.1007/978-3-030-85440-9_11.

[35] M. Rafiei, G. Elkoumy, and W. M. P. van der Aalst, "Quantifying temporal privacy leakage in continuous event data publishing," in *Proc. Int. Conf. Cooperat. Inf. Syst. (CoopIS)* in Lecture Notes in Computer Science, vol. 13591, Bozen-Bolzano, Italy: Springer, Oct. 2022, pp. 75–94.

[36] M. Rafiei and W. M. P. van der Aalst, "Privacy-preserving data publishing in process mining," in *Business Process Management Forum*. Cham, Switzerland: Springer, 2020, pp. 122–138.

[37] M. Rafiei, A. Schnitzler, and W. M. P. van der Aalst, "PC4PM: A tool for privacy/confidentiality preservation in process mining," in *Proc. 19th Int. Conf. Bus. Process Manag. (BPM)*, vol. 2973. Rome, Italy, Sep. 2021, pp. 106–110.

[38] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. Dumas, P. Laud, A. Pankova, and M. Weidlich, "Shareprom: A tool for privacy-preserving inter-organizational process mining," in *Proc. 18th Int. Conf. Bus. Process Manag. (BPM)*, vol. 2673, Sep. 2020, pp. 72–76.

[39] M. Bauer, S. A. Fahrenkrog-Petersen, A. Koschmider, F. Mannhardt, H. van der Aa, and M. Weidlich, "ELPaaS: Event log privacy as a service," in *Proc. CEUR Workshop*, 2019, pp. 159–163.

[40] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Towards cross-organizational process mining in collections of process models and their executions," in *Proc. Int. Conf. Bus. Process Manag.* in Lecture Notes in Business Information Processing, vol. 100, F. Daniel, K. Barkaoui, and S. Dustdar, Eds. Clermont-Ferrand, France: Springer, Aug. 2011, pp. 2–13, doi: 10.1007/978-3-642-28115-0_2.

[41] W. M. P. van der Aalst, "Configurable services in the cloud: Supporting variability while enabling cross-organizational process mining," in *Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst.* in Lecture Notes in Computer Science, vol. 6426, R. Meersman, T. S. Dillon, and P. Herrero, Eds. Crete, Greece: Springer, Oct. 2010, pp. 8–25, doi: 10.1007/978-3-642-16934-2_5.

[42] G. Elkoumy, S. A. Fahrenkrog-Petersen, M. Dumas, P. Laud, A. Pankova, and M. Weidlich, "Secure multi-party computation for inter-organizational process mining," in *Proc. Int. Conf. Bus. Process Modeling, Develop. Support (EMMSAD)* in Lecture Notes in Business Information Processing, vol. 387, S. Nurcan, I. Reinhartz-Berger, P. Soffer, and J. Zdravkovic, Eds. Grenoble, France: Springer, Jun. 2020, pp. 166–181.

[43] C. Liu, H. Duan, Z. Qingtian, M. Zhou, F. Lu, and J. Cheng, "Towards comprehensive support for privacy preservation cross-organization business process mining," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 639–653, Jul./Aug. 2019.

[44] A. Khan, A. Ghose, and H. K. Dam, "Cross-silo process mining with federated learning," in *Proc. 19th Int. Conf. Service-Oriented Comput. (ICSOC)* in Lecture Notes in Computer Science, vol. 13121. Cham, Switzerland: Springer, Nov. 2021, pp. 612–626.

[45] K. Maatouk and F. Mannhardt, "Quantifying the re-identification risk in published process models," in *Proc. Int. Conf. Process Mining Int. Workshops* in Lecture Notes in Business Information Processing, J. Munoz-Gama and X. Lu, Eds. Eindhoven, vol. 433. The Netherlands: Springer, Oct. 2021, pp. 382–394, doi: 10.1007/978-3-030-98581-3_28.

[46] F. Mannhardt, *Sepsis Cases-Event Log*. Eindhoven, The Netherlands: Eindhoven Univ. Technology, 2016, doi: 10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460.

[47] F. Mannhardt and D. Blinde, "Analyzing the trajectories of patients with sepsis using process mining," in *Proc. CEUR Workshop*, vol. 1859, 2017, pp. 72–80.

MAJID RAFIEI received the M.E. degree in electronic commerce from the Amirkabir University of Technology, Tehran. He is currently a Scientific Assistant (Ph.D. Student) with the Chair of Process and Data Science (PADS), RWTH Aachen University. He is also focusing on process mining and specifically on responsible process mining (RPM), where the aim is to use process mining with respect to fairness, accuracy, confidentiality, and transparency (FACT).

WIL M. P. VAN DER AALST (Fellow, IEEE) is currently a Full Professor with RWTH Aachen University, leading the Process and Data Science (PADS) Group. He is also the Chief Scientist at Celonis, part-time affiliated with the Fraunhofer FIT, and a member of the Board of Governors of Tilburg University. He holds unpaid professorship positions with the Queensland University of Technology, since 2003, and Technische Universiteit Eindhoven (TU/e). He is a Distinguished Fellow of Fondazione Bruno Kessler (FBK), Trento, the Deputy CEO of the Internet of Production (IoP) Cluster of Excellence, and the Co-Director of the RWTH Center for Artificial Intelligence.