Supporting Confidentiality in Process Mining Using Abstraction and Encryption

Majid Rafiei[1](✉), Leopold von Waldthausen[2],
and Wil M. P. van der Aalst[1]

[1] Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
majid.rafiei@pads.rwth-aachen.de
[2] Yale University, New Haven, USA

Abstract. Process mining aims to bridge the gap between data science and process science by providing a variety of powerful data-driven analyses techniques on the basis of event data. These techniques encompass automatically discovering process models, detecting and predicting bottlenecks, and finding process deviations. In process mining, event data containing the full breadth of resource information allows for performance analysis and discovering social networks. On the other hand, event data are often highly sensitive, and when the data contain private information, privacy issues arise. Surprisingly, there has currently been little research toward security methods and encryption techniques for process mining. Therefore, in this paper, using *abstraction*, we propose an approach that allows us to hide confidential information in a controlled manner while ensuring that the desired process mining results can still be obtained. We show how our approach can support confidentiality while discovering control-flow and social networks. A connector method is applied as a technique for storing associations between events securely. We evaluate our approach by applying it on real-life event logs.

Keywords: Responsible process mining · Confidentiality · Process discovery · Directly follows graph · Social network analysis

1 Introduction

Data science is changing the way we do business, socialize, conduct research, and govern society. Data are collected on anything, at any time, and in any place. Therefore, it is not surprising that many people are concerned about the responsible use of data. The Responsible Data Science (RDS) [8] initiative focuses on four main questions: (1) How to avoid unfair conclusions even if they are true?, (2) How to answer questions with a guaranteed level of accuracy?, (3) How to answer questions without revealing secrets?, and (4) How to clarify answers such that they become indisputable? This paper focuses on the confidentiality problem (third question) when applying process mining to event data.

Published by Springer Nature Switzerland AG 2020
P. Ceravolo et al. (Eds.): SIMPDA 2018/2019, LNBIP 379, pp. 101–123, 2020.
https://doi.org/10.1007/978-3-030-46633-6_6

Process mining uses event data to provide novel insights into actual processes [2]. There are many activities and techniques in the field of process mining. However, the three basic types of process mining are; process discovery [1], conformance checking [2], and process re-engineering (enhancement) [7]. Also, four perspectives are considered to analyze the event data including; *control-flow*, *organizational*, *case*, and *time* perspective [2]. In this paper, we focus on *control-flow* and *organizational* perspective side by side. A simple definition for process discovery is learning process models from event logs. In fact, a discovery technique takes an event log and produces process model without using additional information [5]. A social network is a social structure which shows relations among social actors (individuals or organizations) [30]. When event data contain information about resources, not only can it be used to thoroughly analyze bottlenecks, but also it turns to a valuable data to derive social networks among resources, involved in the process. Since such event data contain highly sensitive information about the organization and the people involved in the process, confidentiality is a major concern. Note that by confidentiality in process mining, we aim to deal with two important issues; (1) protecting the sensitive data belonging to the organization, (2) protecting the private information about the individuals.

As we show in this paper, *confidentiality in process mining cannot be achieved by merely encrypting all data.* Since people need to use and see process mining results, the challenge is to retain as little information as possible while still being able to have the same desired result. Here, the desired results are process models and social networks. The discovered models (networks) based on the anonymized event data should be identical to the results obtained from the original event data (assuming proper authorizations).

In this paper, we propose an approach to deal with confidentiality in process mining which is based on *abstractions.* Moreover, we present the *connector* method by which the individual traces of a process stay anonymous, yet, at the same time, process models and social networks are discoverable. The proposed framework allows us to derive the same results from secure event logs when compared to the results from original event logs, while unauthorized persons cannot access confidential information. In addition, this framework can provide a secure solution for process mining when processes are cross-organizational.

The remainder of this paper is organized as follows. Section 2 outlines related work and the problem background. In Sect. 3, we clarify process mining, social network discovery, and cryptography as preliminaries. In Sect. 4, the problem is explained in detail. Our approach is introduced in Sect. 5. In Sect. 6 the approach is evaluated, and Sect. 7 concludes the paper.

2 Related Work

In data science, social networks, and information systems, confidentiality has been a topic of interest in the last decade. In computer science, privacy-preserving algorithms and methods in differential privacy are most applicable to confidentiality in process mining. In sequential pattern mining, the field of data science which arguably close to process mining, there has been work on preserving privacy in settings with distributed databases [15] or in cross-organizational settings [31]. Also, privacy-preservation in social networks is a well-researched topic, and most of the research in this field aims to protect the privacy of the individuals involved in a given social network [17]. However, here, we focus on the confidentiality issues arising when initially discovering social networks from event logs that comprise lots of sensitive private data about the individuals.

Although there have been a lot of breakthroughs in the field of process mining ranging from data preprocessing [28] and process discovery [22] to performance analysis [18] and prediction [25], the research field confidentiality and privacy has received relatively little attention. This is despite the fact that already the Process Mining Manifesto [6] points out that privacy concerns are important to be addressed. In the following, we introduce some research regarding *Responsible Process Mining (RPM)* and few publications which focused specifically on confidentiality issues, in the *control-flow* perspective or during *process discovery.*

The topic of Responsible Process Mining (RPM) [3] has been put forward by several authors thereby raising concerns related to fairness, accuracy, confidentiality, and transparency. In [29], a method for securing event logs to be able to do process discovery by Alpha algorithm has been proposed. In [12], a possible approach toward a solution, allowing the outsourcing of process mining while ensuring the confidentiality of dataset and processes, has been presented. In [20], the authors has used a cross-organizational process discovery setting, where public process model fragments are shared as safe intermediates. In [23], the aim is to provide an overview of privacy challenges when process mining is used in human-centered industrial environments. In [27], the authors introduce a framework for ensuring confidentiality in process mining which is utilized and extended in this paper. In [14], a privacy model is proposed for privacy-aware process discovery. In [26], the organizational perspective in process mining is taken into account, and the aim is to provide a privacy-preserving method for role mining, which can be used for generalizing *resources* as individuals in event data. It is also worth noting that process mining can be used for security analyses, e.g., in [10], process mining is used for security auditing.

3 Background

In this section, we briefly present the main concepts and refer the readers to relevant literature for more detailed explanations.

3.1 Process Mining

In the following, we introduce some basic concepts of process mining to which we will refer in this paper.

Events are the smallest data unit in process mining and occur when an activity in a process is executed. Events comprise of multiple attributes including; *Case ID*, *Timestamp*, *Activity*, *Resource*, etc. In Table 1, each row indicates an event. In the remainder of this paper, we will refer to the activities and resources of Table 1 with their abbreviations, e.g., “R” stands for “Register”.

A trace is a sequence of events and represents how a process is executed in one instance, e.g., in Table 1, case 1 is first registered, then documents are verified, and vacancies are checked. Finally, a decision is made for the case.

An event log is a collection of sequences of events which are used as the input of process mining algorithms. Event data are widely available in current information systems [6].

As you can see in Table 1, a “Timestamp” identifies the moment in time at which an event has taken place, and a “Case ID” is what all events in a trace have in common so that they can be identified as part of that process instance. Event logs can also include additional attributes for the events they record. There are two main attribute types that fall under this category. “Event Attributes” which are specific to an event, and “Case Attributes” which are ones that stay the same throughout an entire trace.

Table 1. Sample event log (each row represents an event).

Case ID	Timestamp	Activity	Resource	Cost
1	01-01-2018:08.00	Register (R)	Frank (F)	1000
2	01-01-2018:10.00	Register (R)	Frank (F)	1000
3	01-01-2018:12.10	Register (R)	Joey (J)	1000
3	01-01-2018:13.00	Verify-Documents (V)	Monica (M)	50
1	01-01-2018:13.55	Verify-Documents (V)	Paolo (P)	50
1	01-01-2018:14.57	Check-Vacancies (C)	Frank (F)	100
2	01-01-2018:15.20	Check-Vacancies (C)	Paolo (P)	100
4	01-01-2018:15.22	Register (R)	Joey (J)	1000
2	01-01-2018:16.00	Verify-Documents (V)	Frank (F)	50
2	01-01-2018:16.10	Decision (D)	Alex (A)	500
5	01-01-2018:16.30	Register (R)	Joey (J)	1000
4	01-01-2018:16.55	Check-Vacancies (C)	Monica (M)	100
1	01-01-2018:17.57	Decision (D)	Alex (A)	500
3	01-01-2018:18.20	Check-Vacancies (C)	Joey (J)	50
3	01-01-2018:19.00	Decision (D)	Alex (A)	500
4	01-01-2018:19.20	Verify-Documents (V)	Joey (J)	50
5	01-01-2018:20.00	Special-Case (S)	Katy (K)	800
5	01-01-2018:20.10	Decision (D)	Katy (K)	500
4	01-01-2018:20.55	Decision (D)	Alex (A)	500

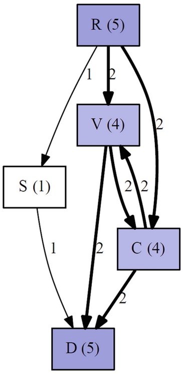
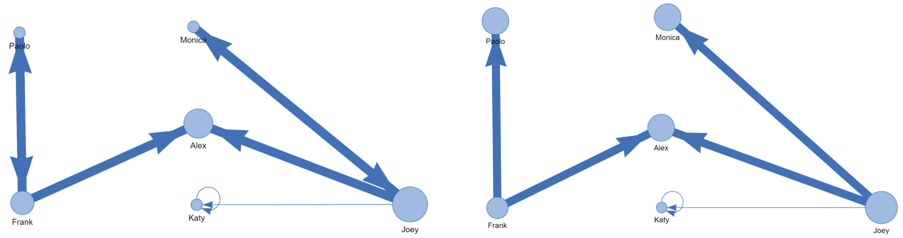


Fig. 1. The DFG resulting from event log Table 1

A Directly Follows Graph (DFG) is a graph where the nodes represent activities and the arcs represent causalities. Activities “a” and “b” are connected by an arrow when “a” is frequently followed by “b”. The weights of the arrows denote the frequency of the relation [19]. Most commercial process mining tools use DFGs. Unlike more advanced process discovery techniques (e.g., implemented in ProM), DFGs cannot express concurrency. Figure 1 shows the DFG resulting from the event log Table 1.

3.2 Discovering Social Networks

There are different methods for discovering social networks from event logs including those based on *causality*, *joint activities*, *joint cases*, etc. [9]. Here,



(a) The handover network resulting from Table 1.

(b) The real-handover network resulting from Table 1 for the real causal dependency threshold 0.1.

Fig. 2. The networks based on causality for the event log Table 1

we, however, focus purely on the metrics based on causality. These metrics monitor for individual cases how work moves from resource to resource. E.g., there is a *handover* relation from individual i to individual j, if there are two subsequent activities where the first is performed by i and the second is performed by j. This relation furthermore becomes a *real-handover* if casual dependency between both activities exists. Note that in this case the directly follows relations between resources are not enough and the real casual dependencies are required. Dependency measure (Eq. 1) can be used to realize whether there is a real casual dependency between two activities (a and b) or not, while a threshold is set as the minimum required value [2]. In Eq. 1, $|a >_L b|$ shows how frequent a is followed by b:

$$|a \Rightarrow_L b| \;=\; \begin{cases} \frac{|a>_L b| - |b>_L a|}{|a>_L b| - |b>_L a| + 1} & \textit{if } a \neq b \\ \frac{|a>_L b|}{|a>_L b| + 1} & \textit{if } a = b \end{cases} \qquad (1)$$

When observing handovers, indirect succession may also be considered. E.g., based on the event log of Table 1, there is a non-real *handover* relation between "Frank" and "Alex" with the depth three. It is non-real due to there is no real casual dependency between all the corresponding activities. Figure 2 shows the networks based on causality having been obtained from event log Table 1.

3.3 Cryptography

Cryptography or cryptology is about constructing and analyzing protocols that prevent third parties or the public from reading private messages [11].

A cryptosystem is a suite of cryptographic algorithms needed to implement a particular security service, most commonly for achieving confidentiality [16]. There are different kinds of cryptosystems. In this paper, we use the following ones.

- *Symmetric Cryptosystem:* The same secret key is used to encrypt and decrypt a message. Data manipulation in symmetric systems is faster than

asymmetric systems as they generally use shorter key lengths. Advanced Encryption Standard (AES) is a symmetric encryption algorithm [13].
- *Asymmetric Cryptosystem:* Asymmetric systems use a public key to encrypt a message and a private key to decrypt it or vice versa. The use of asymmetric systems enhances the security of communication. Rivest-Shamir-Adleman (RSA) is an asymmetric encryption algorithm.
- *Deterministic Cryptosystem:* A deterministic cryptosystem is a cryptosystem which always produces the same ciphertext for a given plaintext and key, even over separate executions of the encryption algorithm.
- *Probabilistic Cryptosystem:* A probabilistic cryptosystem, other than the deterministic cryptosystem, is a cryptosystem which uses randomness when encrypting so that when the same plaintext is encrypted several times, it will produce different ciphertexts.
- *Homomorphic Cryptosystem:* A homomorphic cryptosystem allows computation on ciphertext, e.g., Paillier is a partially homomorphic cryptosystem [24].

4 Problem Definition (Attack Analysis)

To illustrate the challenge of confidentiality in process mining, we start this section with an example. Consider Table 2, describing a totally encrypted event log, belonging to a hospital conducting surgeries. Since we need to preserve difference to find a sequence of activities for each case, discovering process model, and other analyses like social network discovery, "Case ID", "Activity", and "Resource" are encrypted based on a deterministic encryption method. Numerical data (i.e., "Timestamp" and "Cost") are encrypted by a homomorphic encryption method to preserve the ability of basic mathematical computations on the encrypted data. Now suppose that we have background knowledge about surgeons and the approximate cost of different types of surgeries. The question arises whether parts of the log can now be deanonymized.

Owning to the fact that "Cost" is encrypted by a homomorphic encryption method, the maximum value for the "Cost" is the real maximum cost and based on background knowledge we know that e.g., the most expensive event in the hospital was the brain surgery by "Dr. Jone", on "01/09/2018 at 12:00", and the patient name is "Judy". Since "Case ID", "Activity", and "Resource" are encrypted by a deterministic encryption method, we can replace all these encrypted values with the corresponding plain values. Consequently, encrypted data could be made visible without requiring decryption. This example demonstrates that even given completely encrypted event logs small fraction of contextual knowledge can leads to data leakage.

Given domain knowledge, several analyses could be done to identify individuals or extract some sensitive information from an encrypted event log. In the following, we explain couple of them.

- *Exploring the Length of Traces:* One can find the longest/shortest trace, and the related background knowledge can be exploited to realize the actual activities and the related case(s).

- *Frequency Mining:* One can find the most or the less frequent traces and the related background knowledge can be utilized to identify the corresponding case(s) and the actual activities.

These are just some examples demonstrate that encryption alone is not a solution. For example, [21] shows that mobility traces are easily identifiable after encryption. Any approach which is based on solely encrypting the whole event log will furthermore have the following weaknesses:

- *Encrypted Results:* Since results are encrypted, the data analyst is not able to interpret the results. E.g., as data analyst we want to know which paths are the most frequent after "Registration" activity; how can one perform this analysis when the activities are not plain? The only solution is decrypting the results.
- *Impossibility of Accuracy Evaluation:* How can we make sure that a result of the encrypted event log is the same as the result of the plain event log? Again, decryption would be required.

Generally, and as explored by [12], using cryptography is a resource consuming activity, and decryption is even much more resource consuming than encryption. The weaknesses demonstrate that encryption methods should be used wisely and one needs to evaluate closely where they are beneficiary and where unavoidable to provide confidentiality.

Here, we assume that background knowledge could be any contextual knowledge about traces which can result in a *case disclosure* including; frequency of traces, length of traces, exact/approximate time related to the cases, etc. Note that this background knowledge is assumed where unauthorized people can access the anonymized data. For example, given domain knowledge regarding frequency of traces one can guess the actual sequence of activities and possible case(s) (e.g., politicians, celebrities, etc) for the traces which are too rare. Consequently, individuals or minority group of people and their private information would be revealed. Therefore, the *case disclosure* is a crucial type of data leakage which should be prevented.

Table 2. A totally encrypted event log.

Case ID	Activity	Resource	Timestamp
rt!@45	kl56ˆ*	lo09(kl	3125
rt!@45	bn,.ˆq	lo09(kl	3256
)@!1yt	kl56ˆ*	lo09(kl	4879
)@!1yt	bvS(op	/.,ldf	5214
)@!1yt	jhg!676	nb][,b]	6231
erˆ7*	kl56ˆ*	lo09(kl	6534
erˆ7*	2ws34S	v,[]df	7230

5 Approach

Figure 3 illustrates a framework to provide a solution for confidentiality when the desired result is a model. This framework has been inspired by [5], where *abstractions* are introduced as intermediate results for relating models and logs. Here, *abstractions* are directly follows matrix of activities (A-DFM) and directly follows matrix of resources (R-DFM). Figure 4 shows the A-DFM and R-DFM resulting from event log Table 1. A-DFM is considered as the abstraction for relating logs and process models, and R-DFM together with A-DFM are considered as the abstraction for relating logs and social networks which are based on causality. As can be seen in Fig. 3 three different environments and two confidentiality solutions are presented.

- *Forbidden Environment:* In this environment, the actual information system runs that needs to use the real data. The real event logs (EL) produced by this environment contain a lot of valuable confidential information and except some authorized persons no one can access this data.
- *Internal Environment:* This environment is just accessible by authorized stakeholders. A data analyst can be considered as an authorized stakeholder who can access the internal event logs. Event logs in this environment are partially secure, selected results produced in this environment (e.g., a process model) are the same as the results produced in the forbidden environment, and data analyst is able to interpret the results without decryption.
- *External Environment:* In this environment, unauthorized external persons can access the data. Such environments may be used to provide the computing infrastructure dealing with large data sets (e.g., a cloud solution). Event logs in this environment are supposed to be entirely secure, and the results are encrypted. Whenever a data analyst wants to interpret results, they need to be decrypted and converted to an internal version. Furthermore, results from the external environment do not need to be exactly the same as the results from the internal environment, but, the same interpretations need to be provided.

Table 3 shows a summary of our assumptions with respect to the internal and external environments. Note that in the forbidden environment, the main assumption is that only few highly trusted persons can access the data. Therefore, there is no need to employ confidentiality solutions. As described in Sect. 4, contextual knowledge regarding traces is assumed as background knowledge. As can be seen in Fig. 3, the desired results, which are process models (PM) and social networks (SN), can be obtained in each environment. The original event log (EL) is converted to the partially secure event log in the internal environment (EL') and then to the entirely secure event log in the external environment (EL'') by the internal confidentiality solution (ICS) and the external confidentiality solution (ECS) respectively. *Abstractions*, which are intermediate results, are used for proving accuracy. It should be taken into account that since *abstractions* are considered as the outputs of the very last phase before the final results

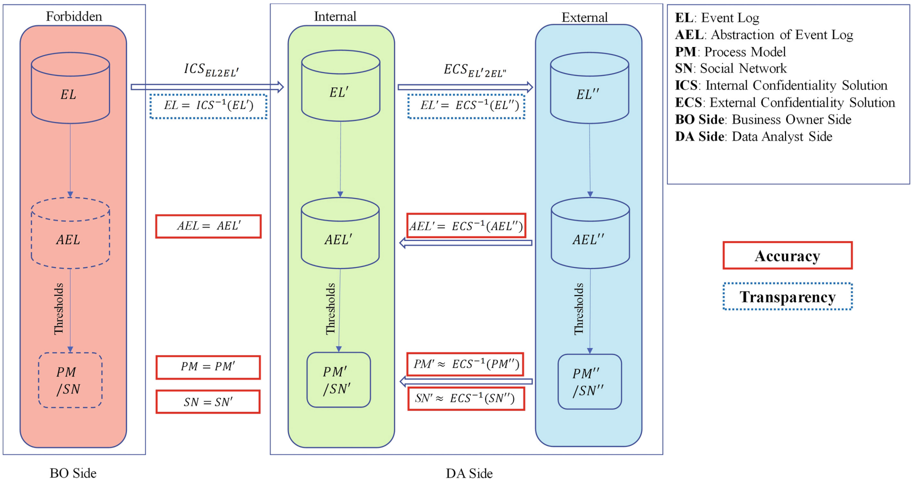


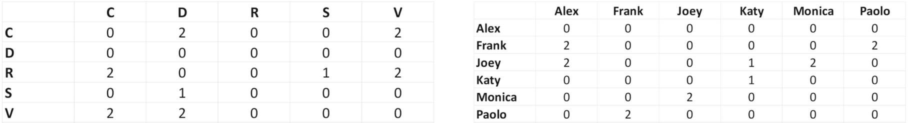
Fig. 3. The proposed framework for confidentiality in process mining.

(only thresholds are required to be applied), when they are equal, the final results would be the same. In addition, *transparency* is provided by the reverse operation of the internal confidentiality solution (ICS^{-1}) and the reverse operation of the external confidentiality solution (ECS^{-1}). In the following, we explain our ICS and ECS in detail.

5.1 Internal Confidentiality Solution (ICS)

For ICS we combine several methods and introduce the connector method. Figure 5 gives an overview of the anonymization procedure.

Filtering and Modifying the Input. The first step to effective anonymization is preparing the data input. To filter the input, simple limits for frequencies can be set, and during loading an event log all traces that do not reach the minimal frequencies are not transferred to the EL'.



	C	D	R	S	V
C	0	2	0	0	2
D	0	0	0	0	0
R	2	0	0	1	2
S	0	1	0	0	0
V	2	2	0	0	0

(a) The A-DFM resulting from Table 1.

	Alex	Frank	Joey	Katy	Monica	Paolo
Alex	0	0	0	0	0	0
Frank	2	0	0	0	0	2
Joey	2	0	0	1	2	0
Katy	0	0	0	1	0	0
Monica	0	0	2	0	0	0
Paolo	0	2	0	0	0	0

(b) The R-DFM resulting from Table 1.

Fig. 4. The *abstractions* from Table 1.

Table 3. The general assumptions based on the environments

	Internal	External
Who has access to the data?	Employees Internal Data Analysts	External Data Analysts Anyone else
Trust to the ones who have data access	High	Low
Background knowledge	Broad	Limited
What data should be kept secure?	Direct individual/organization sensitive data which is not necessary for the desired result	Direct/indirect individual/organization sensitive data
Desired results	-Social network based on causality from EL' -Process model based on DFG from EL'	-Social network based on causality from EL'' -Process model based on DFG from EL''

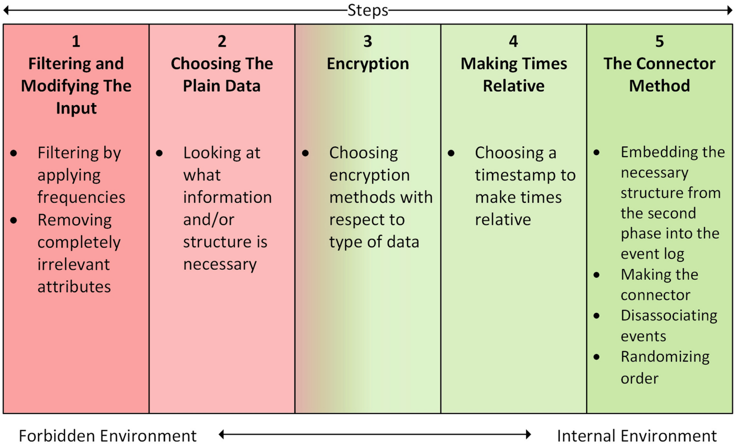

Steps
1
Filtering and Modifying The Input
• Filtering by applying frequencies
• Removing completely irrelevant attributes
2
Choosing The Plain Data
• Looking at what information and/or structure is necessary
3
Encryption
• Choosing encryption methods with respect to type of data
4
Making Times Relative
• Choosing a timestamp to make times relative
5
The Connector Method
• Embedding the necessary structure from the second phase into the event log
• Making the connector
• Disassociating events
• Randomizing order
Forbidden Environment
Internal Environment

Fig. 5. The internal confidentiality solution.

Choosing the Plain Data. As mentioned, we need to produce interpretable results. Hence, some parts of event log remain as plain text in the internal version of the secure event log (EL'). We should decide what information and/or structure is strictly necessary for the desired analysis. Based on our considered abstractions (A-DFM and R-DFM), the only information necessary are directly follows relations between activities/resources.

Encryption. Here there are two important choices. The first choice is which columns of the event log should be encrypted. Second, we need to decide which algorithms should be used. For the internal environment, since we want to keep the capability of applying basic mathematical computations on the encrypted values, we use Paillier for numeric attributes (i.e., "Cost"), and AES-128 with

Table 4. The first 10 rows of Table 1 after encryption and making times relative

Case ID	Timestamp	Activity	Resource	Cost
1	00-00-0000:08.00	Register (R)	Frank (F)	0820315
2	00-00-0000:10.00	Register (R)	Frank (F)	0820315
3	00-00-0000:12.10	Register (R)	Joey (J)	0820315
3	00-00-0000:13.00	Verify-Documents (V)	Monica (M)	0650210
1	00-00-0000:13.55	Verify-Documents (V)	Paolo (P)	0650210
1	00-00-0000:14.57	Check-Vacancies (C)	Frank (F)	0650900
2	00-00-0000:15.20	Check-Vacancies (C)	Paolo (P)	0650900
4	00-00-0000:15.22	Register (R)	Joey (J)	0820315
2	00-00-0000:16.00	Verify-Documents (V)	Frank (F)	0650210
2	00-00-0000:16.10	Decision (D)	Alex (A)	0710155

only ASCII characters as the key is used for other attributes. Note that the encrypted values shown in the paper are not necessarily the real outputs of the encryption methods (they are just unintelligible text).

Making Times Relative. Times need to be modified because keeping the exact epoch time of an event can allow one to identify it. The naive approach, of setting the starting time of every trace to 0, would make it impossible to replay events and reconstruct the original log. Thus, we select another time that all events are made relative to. This time can be kept secure along with the keys for decryption. Table 4 shows the first 10 rows of our sample log after encrypting cost and making times relative to the "01-01-2018:00.00".

Table 5. Adding previous activities/resources and previous IDs.

Case ID	Timestamp	Activity	Prev. Activity	Resource	Prev. Resource	Cost	ID	Prev. ID
1	00-00-0000:08.00	R	START	Frank (F)	START	0820315	31	00
2	00-00-0000:10.00	R	START	Frank (F)	START	0820315	32	00
3	00-00-0000:12.10	R	START	Joey (J)	START	0820315	38	00
3	00-00-0000:13.00	V	R	Monica (M)	Joey (J)	0650210	41	38
1	00-00-0000:13.55	V	R	Paolo (P)	Frank (F)	0650210	55	31
1	00-00-0000:14.57	C	V	Frank (F)	Paolo (P)	0650900	09	55
2	00-00-0000:15.20	C	R	Paolo (P)	Frank (F)	0650900	86	32
4	00-00-0000:15.22	R	START	Joey (J)	START	0820315	47	00
2	00-00-0000:16.00	V	C	Frank (F)	Paolo (P)	0650210	75	86
2	00-00-0000:16.10	D	V	Alex (A)	Frank (F)	0710155	56	75

Table 6. The event log after adding the connector column

Case ID	Timestamp	Activity	Prev. Activity	Resource	Prev. Resource	Cost	ID	Prev. ID	Connector
1	00-00-0000:08.00	R	START	Frank (F)	START	0820315	31	00	1<@sadd21?
2	00-00-0000:10.00	R	START	Frank (F)	START	0820315	32	00	!s*f*+dsf3
3	00-00-0000:12.10	R	START	Joey (J)	START	0820315	38	00	ça/ds23”w’
3	00-00-0000:13.00	V	R	Monica (M)	Joey (J)	0650210	41	38	.,m;lo,mh
1	00-00-0000:13.55	V	R	Paolo (P)	Frank (F)	0650210	55	31	;l4;l,’kjh
1	00-00-0000:14.57	C	V	Frank (F)	Paolo (P)	0650900	09	55	*’;k!kjm.”
2	00-00-0000:15.20	C	R	Paolo (P)	Frank (F)	0650900	86	32	l:mj/.m @p
4	00-00-0000:15.22	R	START	Joey (J)	START	0820315	47	00	;k;lm.lå@,
2	00-00-0000:16.00	V	C	Frank (F)	Paolo (P)	0650210	75	86	=ó@k;d/f.m
2	00-00-0000:16.10	D	V	Alex (A)	Frank (F)	0710155	56	75	’;,lk.;hj!

The Connector Method. Using the connector method we embed the structure, which can be used for extracting directly follows relations, into EL'. Also, the connector method helps us to reconstruct the full original event logs when keys and relative values are given. In the first step, the previous activity (“Prev. Activity”) and the previous resource (“Prev. Resource”) columns are added in order to identify which arcs can be directly connected.

In the second step, we find a way to securely save the information contained in the “Case ID”, without allowing it to link the events. This can be done by giving each row a random ID (“ID”) and a previous ID (“Prev. ID”). These uniquely identify the following event in a trace because the IDs are not generic like activity names. The ID for start activities is always a number of zeros. Table 5 shows the log after adding “Prev. Activity”, “Prev. Resource”, “ID”, and “Prev. ID”.

In the third step, regarding the fact that these columns contain the same information previously found in the “Case ID”, they must be hidden and secured. This can be done by concatenating the “ID” and “Prev. ID” of each row and

Table 7. The output event log after applying ICS

Timestamp	Activity	Prev. Activity	Resource	Prev. Resource	Cost	Connector
08.00	R	START	Frank (F)	START	0820315	1<@sadd21?
01.02	C	V	Frank (F)	Paolo (P)	0650900	!s*f*+dsf3
10.00	R	START	Frank (F)	START	0820315	ça/ds23”w’
15.22	R	START	Joey (J)	START	0820315	.,m;lo,mh
00.50	V	R	Monica (M)	Joey (J)	0650210	;l4;l,’kjh
00.40	V	C	Frank (F)	Paolo (P)	0650210	*’;k!kjm.”
12.10	R	START	Joey (J)	START	0820315	l:mj/.m @p
05.20	C	R	Paolo (P)	Frank (F)	0650900	;k;lm.lå@,
05.55	V	R	Paolo (P)	Frank (F)	0650210	=ó@k;d/f.m
00.10	D	V	Alex (A)	Frank (F)	0710155	’;,lk.;hj!

encrypting those using AES. Due to the nature of AES, neither orders nor sizes of the IDs remain inferable. The concatenation can be done in any style, in this example, we however simply concatenate "ID" and "Prev. ID",e.g., connector of the first row would be "3100". To retain the "ID" and "Prev. ID" one simply needs to decrypt the "Connector" column and cut the resulting number in two equal parts. This method requires that every time the two IDs differ by a factor 10 a zero must be added to guarantee equal length. Table 6 shows the log after concatenating the ID columns and encrypting them as a connector.

In the final step, we use the "Case ID" to anonymize the "Time tamp". The "Time tamp" attribute of events which have the same "Case ID" is made relative to the preceded one. The exception is the first event of each trace which remains unchanged. This allows the complete calculation of all durations of the arcs in a directly follows graph but makes it complicated to identify events based on the epoch times they occurred at. After creating the relative times, we are free to delete the "Case ID" and disarray the order of all rows, ending up with an unconnected log in Table 7.

Table 7 is internally secure event log (EL'), which can be used by a data analyst to create a A-DFM and a R-DFM. It is trivial to see that if process/social network discovery could have been done on the plain event log (EL), AEL would be identical to AEL' (i.e., both are the same A-DFM/R-DFM) and the final desired results would be the same. Note that when the desired result is a process model, resource related information ("Resource" and "Prev. Resource" columns) can be removed from Table 7. Moreover, when the desired result is a handover network, activity related information ("Activity" and "Prev. Activity") can be removed, since the real causal dependencies do not need to be taken into account.

Comparing Table 7 and the original log, one can see that there is no answer for the following questions in EL' anymore: (1) *Who was responsible for doing an activity for case c*? (2) *What is the sequence of activities which has been done for case c*? (3) *How long did it take to process case c*? (4) *What is the cost of activity a which has been done by resource r for case c*? (5) *What is the the length of case c*? (6) *What is the the frequency of case c*?, and many other questions related to the cases.

It is also worth noting that since we assume that the data in the internal environment can be accessed by the internal trustworthy people who already know the organizational structure, the plain resources are not considered as a privacy issue. In fact, EL' is a partially secure version of event log in such a way that it contains the minimum level of information, which a data analyst might need to reach the result. Although ICS does not preserve the standard format of the event log which is used by the current process discovery techniques, the intermediate input it provides can be used by the current tools. In the External Confidentiality Solution (ECS), we need to avoid any form of data leakage and privacy risks based on the assumed background knowledge.

Table 8. The event log after encrypting activities and resources

Timestamp	Activity	Prev. Activity	Resource	Prev. Resource	Cost	Connector
08.00	AgeIRL	1wBo2I	908G2F	1wBo2I	0820315	1<@sadd21?
01.02	5rYd7h	v42jbE	908G2F	9iYoqT	0650900	!s*f*+dsf3
10.00	AgeIRL	1wBo2I	908G2F	1wBo2I	0820315	ça/ds23”w’
15.22	AgeIRL	1wBo2I	RjjZyw	1wBo2I	0820315	.,m;lo,mh
00.50	v42jbE	AgeIRL	eBzosT	RjjZyw	0650210	;l4;l,’kjh
00.40	v42jbE	5rYd7h	908G2F	9iYoqT	0650210	*’;k!kjm.”
12.10	AgeIRL	1wBo2I	RjjZyw	1wBo2I	0820315	l:mj/.m @p
05.20	5rYd7h	AgeIRL	9iYoqT	908G2F	0650900	;k;lm.lå@,
05.55	v42jbE	AgeIRL	9iYoqT	908G2F	0650210	=ó@k;d/f.m
00.10	aUj71B	v42jbE	WLTZqP	908G2F	0710155	’;,lk.;hj!

5.2 External Confidentiality Solution (ECS)

In the external environment, the plain part of the event log may cause data leakage. Therefore, the whole event log gets encrypted. Moreover, some additional attributes, which can lead to data leakage even in the encrypted form, are projected. In the following, our two-steps *ECS* is explained.

Encrypting the Plain Part. In this step, activities and resources are encrypted by a deterministic encryption method like AES. A deterministic encryption method must be used, because for discovering DFMs, differences should be preserved. Table 8 shows the result after encrypting activities and resources.

However, after encrypting, detecting “START” activities seem to be impossible, and without detecting them, finding traces becomes impossible. For identifying the “START” activities, we can go through the “Activity” (“Resource”) and “Prev. Activity” (“Prev. Resource”) columns, the activities (resources) which are appeared in the “Prev. Activity” (“Prev. Resource”) column but not appeared in the “Activity” (“Resource”) column are the “START” activities (resources).

Fortifying Encryption and/or Projecting Event Logs. As mentioned in Sect. 4, since resources are encrypted by a deterministic encryption method, and costs are encrypted by a homomorphic encryption method, which preserves differences, by comparison, one can find the minimum/maximum cost, which can be used as knowledge for extracting confidential or private information (e.g. name of resource). In order to decrease the effect of such analyses, fortifying encryption and/or projecting event logs could be done. Here, we project the costs which are indeed not necessary for the desired results.

6 Evaluation

We consider three evaluation criteria for the proposed approach, yet, at the same time, performance is also taken into account:

- *Ensuring Confidentiality:* As explained in Sect. 5, we can improve confidentiality by defining different environments and indicating a level of information which is accessible in each of these environments. In addition, using multiple encryption methods and our connector method for disassociating events from their cases provide high level of confidentiality with respect to the assumed background knowledge.
- *Reversibility:* When the keys and the value used for making times relative are given, both ICS and ECS are reversible, which means that transparency is addressed by the proposed approach.
- *Accuracy:* To show the accuracy of our approach, by a case study we illustrate that the results obtaining from the secure version of event logs are exactly the same as the results obtaining from the original event logs.

6.1 Correctness of the Approach

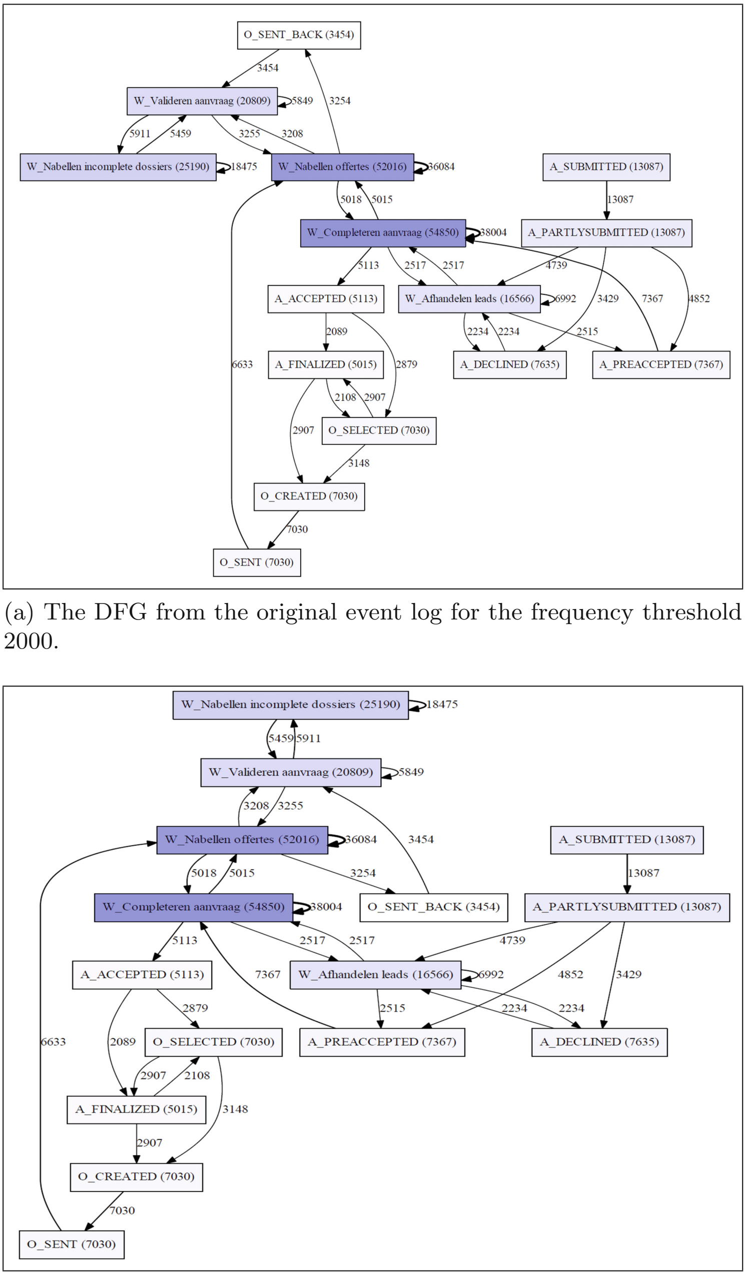
As can be seen in Fig. 3, from accuracy point of view, we need to show that the abstraction of the original event log is the same as the abstraction of the internal event log ($AEL = AEL'$) (rule (2)), and also the abstraction of the internal event log is the same as the abstraction of the external event log, which is encrypted ($AEL' = ECS^{-1}(AEL'')$) (rule (3)). To show that these relations are guaranteed to hold, we have implemented an interactive environment in Python and tested the approach on multiple event logs. In the following, we illustrate the results obtaining by applying the approach on “BPI challenge 2012”.

$$AEL = AEL' \Rightarrow PM = PM' \wedge SN = SN' \tag{2}$$

$$AEL' = ECS^{-1}(AEL'') \Rightarrow PM' \approx ECS^{-1}(PM'') \wedge SN' \approx ECS^{-1}(SN'') \tag{3}$$

In the first step, EL', and EL'' were created. Then, to verify that AEL and AEL' are identical, we created a DFG from the original and the internal version of event log. Figure 6 shows the DFGs resulting from BPI challenge 2012 for the frequency threshold 2000. As one can see both DFGs are the same. Also, Fig. 7 shows the DFG resulting from BPI challenge 2012 for the same frequency threshold (2000) in the external environment. As can be seen, this DFG is also the same as the DFG from the EL and EL' (modulo renaming and layout differences), i.e., all the process discovery algorithms which are based on a DFG would lead to the same process models in the different environments.

In order to demonstrate that the causality based social networks in the secure environments are the same as the actual social networks from the original event log, we have made the real-handover from the original and internal version of event log for BPI challenge 2012. Figure 8 shows the networks for the real causal



(a) The DFG from the original event log for the frequency threshold 2000.

(b) The DFG from the internal event log for the frequency threshold 2000.

Fig. 6. Comparing the DFG from the EL with the DFG from the EL' for BPI challenge 2012: both graphs are identical, only layouts are different.

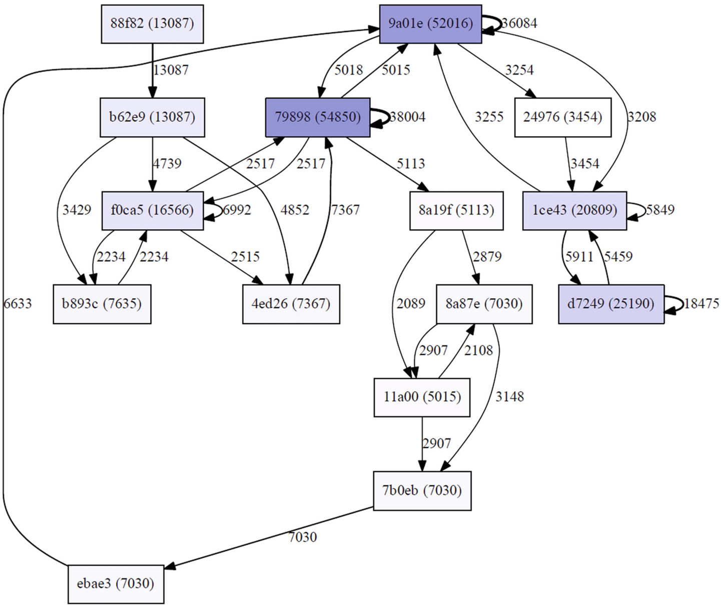


Fig. 7. The DFG from the external event log (BPI challenge 2012) for the frequency threshold 2000.

dependency threshold 0.5 and the frequency threshold 50. The networks are exactly the same. It is obvious that the network from the external version of event log must be the same (while resources are encrypted). Nevertheless, in Fig. 9, we have zoomed in the highlighted parts of Fig. 8 for the networks resulting from the internal and external environment (the same thresholds were applied), and relations are the same except the fact that resources in the external environment are encrypted. As can be seen in Fig. 9 all the relations of the resource "11201" are the same[1].

6.2 Performance

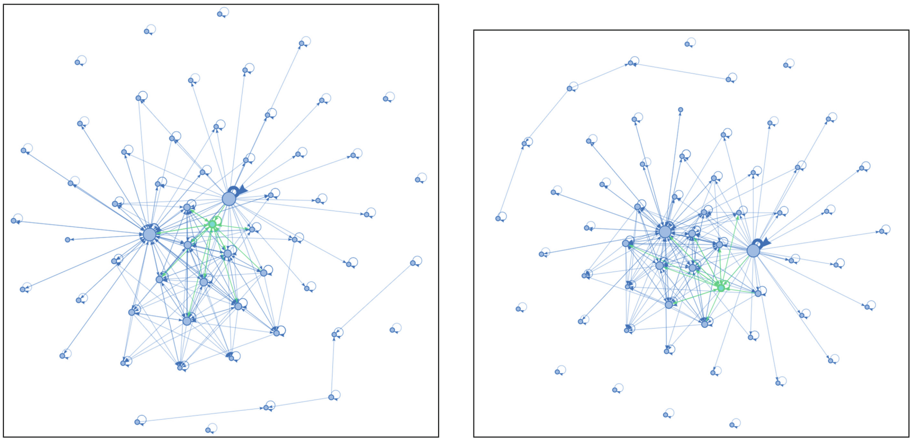
To demonstrate performance of the approach, we apply it on several benchmarking [4] and real-life event logs. Table 9 shows specifications of the used event logs. "BPI Challenge 2012" and "BPI Challenge 2017" are used to evaluate the performance when social networks are discovered, and the benchmarking event logs are used to evaluate the performance of the control-flow discovery.

Figure 10 shows how the control-flow discovery scales when using the benchmarking event logs and increasing the number of events exponentially, and Fig. 11 shows the performance of social network discovery when the approach is applied on the two real-life event logs with different scales. All runtimes are in

[1] It has 11 relations with the resources "112", "11000", "11189", "10913", "10861", "10909", "11181", "11180", "11119", "11203", and "11201".

Table 9. The specifications of the event logs used for evaluation

Event Log	Cases	Events	Variants	Activities	Resources
Choice Loop 1000	1000	7178	436	81	-
Choice Loop 10000	10000	70659	3202	81	-
Choice Loop 100000	100000	706598	21643	81	-
Sequence Loop 1000	1000	40783	1000	80	-
Sequence Loop 10000	10000	407791	9985	80	-
Sequence Loop 100000	100000	4078819	98821	80	-
BPI Challenge 2012	13087	262200	4366	24	69
BPI Challenge 2017	31509	561671	4047	26	145



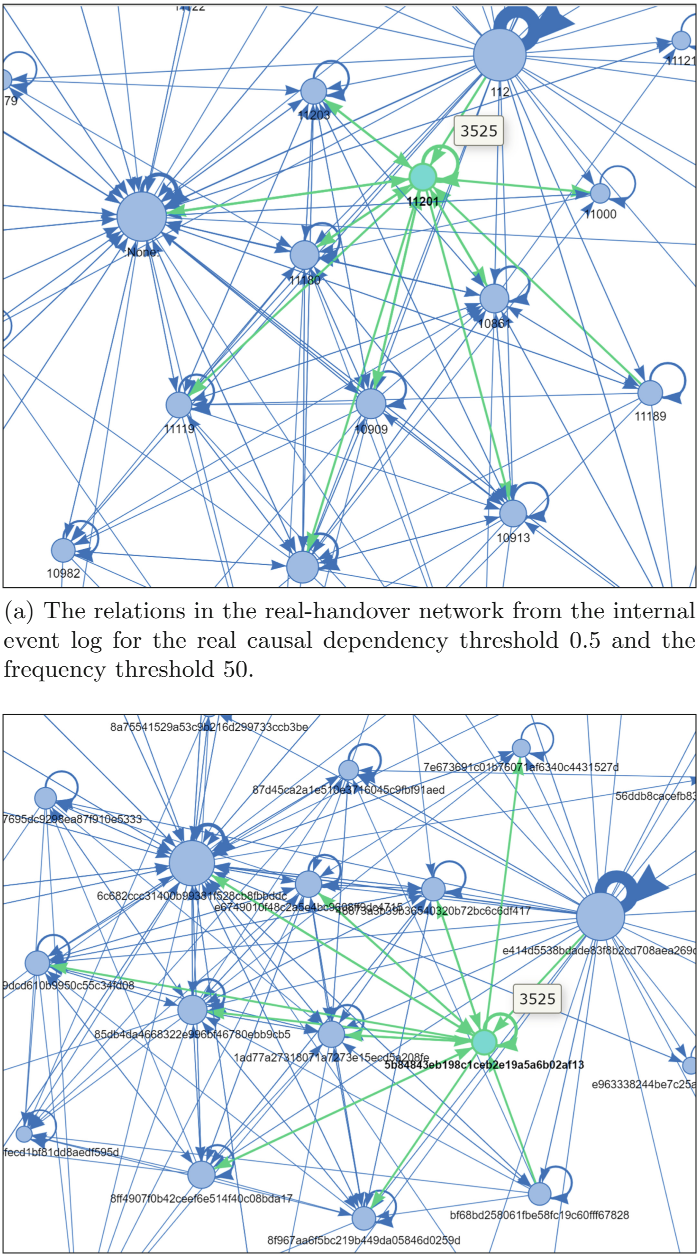
(a) The real-handover network from the original event log for the real causal dependency threshold 0.5 and the frequency threshold 50.

(b) The real-handover network from the internal event log for the real causal dependency threshold 0.5 and the frequency threshold 50.

Fig. 8. Comparing the real-handover networks resulting from BPI challenge 2012: both networks are identical.

milliseconds and have been tested using an Intel i7 Processor with 1.8 GHz and 16 GB RAM.

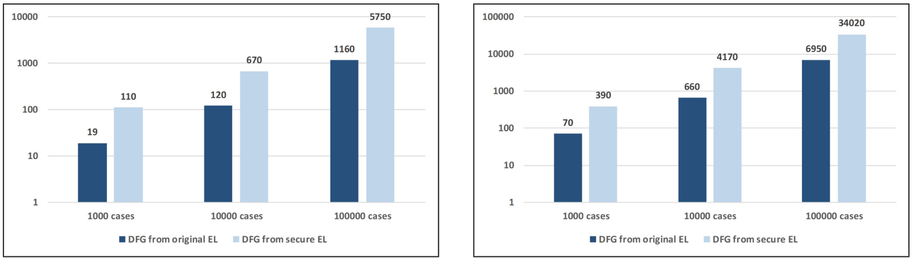
In Fig. 10, the darker bars show the execution time for discovering the DFG from the original event logs, and the lighter bars show the execution time for discovering the DFG from the secure event logs. One can see a linear increase of the runtime in milliseconds when adding choices or loops. In addition, as can be seen in Fig. 11, when the metric is real-handover, the execution time for discovering social networks is higher, since the real causal dependencies between subsequent resources need to be verified.


11122
79
11203
112
11121
3525
11201
11000
None
11180
10861
11119
10909
11189
10913
10982
8a75541529a53c9b216d299733ccb3be
7e673691c01b76071af6340c4431527d
87d45ca2a1e510e3716045c9fbf91aed
56ddb8cacefb83
7695dc9298ea87f910e5333
6c682ccc31400b99381f528cb8fbbddc
e6749010f48c2a5e4bc9608ff9de4715
48873a3b39b36540320b72bc6c6df417
e414d5538bdade83f8b2cd708aea269c
9dcd610b9950c55c34fd08
3525
85db4da4668322e996bf46780ebb9cb5
1ad77a27318071a7273e15ecd5a208fe
5b84843eb198c1ceb2e19a5a6b02af13
e963338244be7c25a
fecd1bf81dd8aedf595d
8ff4907f0b42ceef6e514f40c08bda17
bf68bd258061fbe58fc19c60fff67828
8f967aa6f5bc219b449da05846d0259d

(a) The relations in the real-handover network from the internal event log for the real causal dependency threshold 0.5 and the frequency threshold 50.

(b) The relations in the real-handover network from the external event log for the real causal dependency threshold 0.5 and the frequency threshold 50.

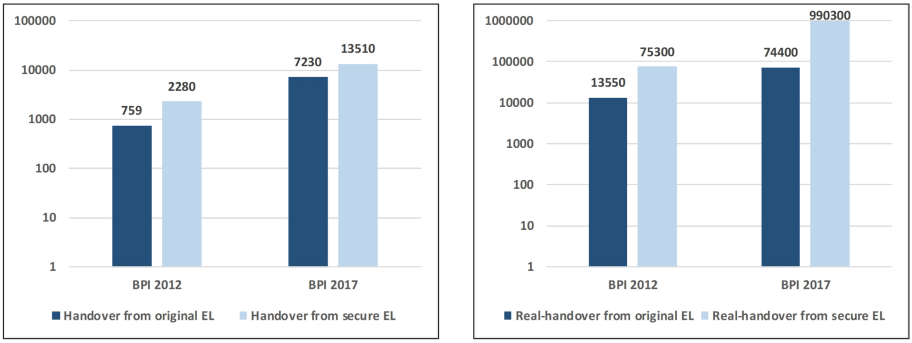
Fig. 9. Comparing the relations of resource “11201” in the real-handover networks resulting from EL' and EL'' for BPI challenge 2012



(a) Execution time for the choice loop events.

(b) Execution time for the sequence loop events.

Fig. 10. The execution time of the control-flow discovery when using the benchmaking event logs.



(a) Execution time for the handover networks.

(b) Execution time for the real-handover networks.

Fig. 11. The execution time of the social network discovery when using the real-life event logs.

7 Conclusions

This paper presented a novel approach to ensure confidentiality in process mining when the desired results are models. We demonstrated that confidentiality in process mining cannot be achieved by only encrypting an event log. We outlined the little related work, most of which use just encryption, and explained the weaknesses of following this approach. The new approach is introduced since there always exists a trade-off between confidentiality and data utility. Therefore, we reasoned backwards from the desired results and how they can be obtained with as little data as possible.

Here, process models and social networks were considered as the desired results, and the confidentiality solutions presented in the context of a framework that can be extended to include other forms of process mining, i.e., different ICS and ECS could be explored for different process mining activities. Moreover, the proposed framework could be utilized in cross-organizational context such that each environment could cover specific constraints and authorizations of a party. In this paper, we focused on causality based social networks, and in the future other metrics could be explored. Moreover, in the future, a measure for confidentiality could be defined so that the effectiveness of different solutions in this research area could be quantified and compared.

We have utilized a new method named "connector", which can be employed in any situation where we need to store associations securely. For evaluating the proposed approach, we have implemented an interactive environment in Python, and a real-life log was used as the case study.

Acknowledgment. We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research interactions.

References

1. van der Aalst, W.M.P.: Business process management: a comprehensive survey. ISRN Softw. Eng. **2013**, 1–37 (2013)
2. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
3. van der Aalst, W.M.P.: Responsible data science: using event data in a "people friendly" manner. In: Hammoudi, S., Maciaszek, L.A., Missikoff, M.M., Camp, O., Cordeiro, J. (eds.) ICEIS 2016. LNBIP, vol. 291, pp. 3–28. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62386-3_1
4. van der Aalst, W.M.P.: Benchmarking logs to test scalability of process discovery algorithms. Eindhoven University of Technology (2017). https://data.4tu.nl/repository/uuid:1cc41f8a-3557-499a-8b34-880c1251bd6e. Accessed 01 Apr 2018
5. van der Aalst, W.M.P.: Process discovery from event data: relating models and logs through abstractions. Wiley Interdiscip. Rev.: Data Mining Knowl. Discov. **8**(3), e1244 (2018)
6. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28108-2_19
7. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdiscip. Rev.: Data Mining Knowl. Discov. **2**(2), 182–192 (2012)
8. van der Aalst, W.M.P., Bichler, M., Heinzl, A.: Responsible data science. Bus. Inf. Syst. Eng. **59**(5), 311–313 (2017)
9. van der Aalst, W.M.P., Reijers, H.A., Song, M.: Discovering social networks from event logs. Comput. Support. Coop. Work (CSCW) **14**(6), 549–593 (2005)
10. Accorsi, R., Stocker, T., Müller, G.: On the exploitation of process mining for security audits: the process discovery case. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 1462–1468. ACM (2013)
11. Bellare, M., Rogaway, P.: Introduction to modern cryptography. UCSD CSE **207**, 207 (2005)
12. Burattin, A., Conti, M., Turato, D.: Toward an anonymous process mining. In: 2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud), pp. 58–63. IEEE (2015)
13. Daemen, J., Rijmen, V.: The design of Rijndael: AES-the advanced encryption standard. Springer, Heidelberg (2013)

14. Fahrenkrog-Petersen, S.A., van der Aa, H., Weidlich, M.: PRETSA: event log sanitization for privacy-aware process discovery. In: International Conference on Process Mining, ICPM 2019, Aachen, Germany, 24–26 June 2019, pp. 1–8 (2019)
15. Kapoor, V., Poncelet, P., Trousset, F., Teisseire, M.: Privacy preserving sequential pattern mining in distributed databases. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 758–767. ACM (2006)
16. Katz, J., Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
17. Kleinberg, J.M.: Challenges in mining social network data: processes, privacy, and paradoxes. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 4–5. ACM (2007)
18. Leemans, M., van der Aalst, W.M.P., van den Brand, M.G.: Hierarchical performance analysis for process mining. In: Proceedings of the 2018 International Conference on Software and System Process, pp. 96–105. ACM (2018)
19. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Scalable process discovery and conformance checking. Softw. Syst. Model. **17**(2), 599–631 (2016). https://doi.org/10.1007/s10270-016-0545-x
20. Liu, C., Duan, H., Qingtian, Z., Zhou, M., Lu, F., Cheng, J.: Towards comprehensive support for privacy preservation cross-organization business process mining. IEEE Trans. Serv. Comput. **1**, 1–1 (2016)
21. Ma, C.Y., Yau, D.K., Yip, N.K., Rao, N.S.: Privacy vulnerability of published anonymous mobility traces. IEEE/ACM Trans. Netw. (TON) **21**(3), 720–733 (2013)
22. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: Guided process discovery-a pattern-based approach. Inf. Syst. **76**, 1–18 (2018)
23. Mannhardt, F., Petersen, S.A., Oliveira, M.F.: Privacy challenges for process mining in human-centered industrial environments. In: 2018 14th International Conference on Intelligent Environments (IE), pp. 64–71. IEEE (2018)
24. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
25. Pourbafrani, M., van Zelst, S.J., van der Aalst, W.M.P.: Scenario-based prediction of business processes using system dynamics. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 422–439. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_27
26. Rafiei, M., van der Aalst, W.M.P.: Mining roles from event logs while preserving privacy. In: Di Francescomarino, C., Dijkman, R., Zdun, U. (eds.) BPM 2019. LNBIP, vol. 362, pp. 676–689. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37453-2_54
27. Rafiei, M., von Waldthausen, L., van der Aalst, W.M.P.: Ensuring confidentiality in process mining. In: Proceedings of the 8th International Symposium on Data-driven Process Discovery and Analysis (SIMPDA 2018), Seville, Spain, 13–14 December 2018, pp. 3–17 (2018). http://ceur-ws.org/Vol-2270/paper1.pdf
28. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Repairing outlier behaviour in event logs. In: Abramowicz, W., Paschke, A. (eds.) BIS 2018. LNBIP, vol. 320, pp. 115–131. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93931-5_9

29. Tillem, G., Erkin, Z., Lagendijk, R.L.: Privacy-preserving alpha algorithm for software analysis. In: 37th WIC Symposium on Information Theory in the Benelux/6th WIC/IEEE SP Symposium on Information Theory and Signal Processing in the Benelux (2016)
30. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
31. Zhan, J.Z., Chang, L., Matwin, S.: Privacy-preserving collaborative sequential pattern mining. Technical report, Ottawa Univ (Ontario) School of Information Technology (2004)