Measuring Privacy Compliance with Process Specifications

Sebastian Banescu
Eindhoven University of Technology
Eindhoven, The Netherlands
s.e.banescu@student.tue.nl

Nicola Zannone
Eindhoven University of Technology
Eindhoven, The Netherlands
n.zannone@tue.nl

***Abstract*—Enforcement relies on the idea that infringements are violations and as such should not be allowed. However, this notion is very restrictive and cannot be applied in unpredictable domains like healthcare in which it is impossible to know in advance when and where an emergency situation will occur. To address this issue, we need conformance metrics for detecting and quantifying infringements of policies and procedures. However, existing metrics usually consider every deviation from specifications equally making them inadequate to measure the severity of infringements. In this paper, we identify a number of factors which can be used to quantify deviations from process specifications. These factors drive the definition of metrics that allow for a more accurate measurement of privacy infringements. We demonstrate how the proposed approach can be adopted to enhance existing conformance metrics through a case study on the provisioning of healthcare treatment.**

***Keywords*-privacy; metric; compliance; business process.**

I. Introduction

Communication problems including the unavailability, inaccuracy and incompleteness of data, represent the most common cause of medical errors. This is a critical issue in healthcare. A study conduced by the Institute of Medicine of the National Academy of Sciences in 1999 [1] shows that medical errors are the fifth leading cause of death in the US and cost healthcare providers billions of dollars per year. The need of mitigating the risk of medical errors and reducing the costs involved in the healthcare process has spurred intensive efforts in the past years into the process of digitizing medical information and in the development of IT systems for healthcare [2].

The digitization of information, however, has a significant impact on the security and privacy requirements of healthcare providers. It makes it possible to collect and store a massive amount of sensitive information which is easily accessible on the network locally or remotely. This increases the risk of privacy breaches. Yet, privacy breaches have serious financial consequences on organizations. A recent study [3] shows that privacy breaches cost $ 3.4 million on average and reduce the customers' trust in organizations that suffered the breaches.

Privacy is becoming more stringently regulated especially in healthcare. Many countries have enacted privacy and data protection laws and regulations (e.g., EU Directive 95/46/EC, HIPAA) which impose stringent requirements on the collection, processing and disclosure of personal data [4]. Accordingly, organizations should implement security measures to ensure that personal data are protected from accidental and unlawful loss and misuse.

The demand of technical means to enforce legal requirements has lead to the definition of several privacy policy languages for the protection of sensitive data (e.g., [5]). The enforcement of these policies often relies on a reference monitor and compensation actions in order to detect possible policy infringements and execute the actions needed to bring the system back to a correct state [6]–[8]. Unfortunately, these enforcement mechanisms are rigid and are not able to assess the consequences of infringements. They only guarantee compliance with specifications by just preventing infringements to occur; however, this may be unacceptable in certain circumstances like emergency situations.

Emergency situations (which are rather common in healthcare) are often dealt with the so-called "break the glass" protocol. This protocol allows users to bypass security mechanisms. However, it can be misused [9], leading to serious privacy violations. Therefore, it is usually complemented with logging mechanisms which record the user behavior. Then, auditors sample and inspect the audit trails recorded by the system to detect possible privacy infringements. However, auditing activities are usually manual, leading to situations in which privacy violations remain undetected. For example, at the Geneva University Hospital more than 20,000 records are opened every day [10]; investigating all accesses manually would not be feasible.

To address this problem, we need methods for analyzing the user behavior against specifications and measuring the severity of infringements. There exist several solutions for detecting system behaviors that do not conform to specifications [11]–[15]. However, these solutions often consider any (type of) deviation from the specification equally, making the obtained measures meaningless from a privacy perspective. To distinguish acceptable infringements (e.g., to save patient's life) from unacceptable infringements (e.g., frauds or privacy invasion), they need to be complemented with metrics able to measure the amount of privacy loss.

A number of metrics for measuring privacy already exist in the literature [16], [17]. These metrics usually measure privacy in terms of anonymity. Another proposal for measur-

ing the severity of infringements is quantitative information flow [18] which aims to measure "how much" information is being leaked. In particular, it measures the probability that an adversary can guess the value of a variable by looking at the value of other variables. These metrics, however, are not appropriate to assess privacy loss in cases where users have access to information and should be able to link it to the data subject in order to provide appropriate medical treatments.

This paper brings two contributions. Primarily, it introduces a framework that facilitates the quantification of privacy violations. The framework is founded on a number of factors that make it possible to measure the magnitude of deviations of the actual behavior from process specifications. These factors are based on information that is usually available in audit trails:

- the task that was executed;
- the user who executed the task together with the role held by the user during the execution of the task;
- the data accessed during the execution of the task.

To the best of our knowledge, there is no prior work that considers such a broad spectrum of factors for compliance checking.

The second contribution of this paper is to show how existing conformance metrics can be improved to measure the severity of privacy infringements. To make the proposed approach more concrete, we revise the Levenshtein distance, a sequence distance metric adopted in [11] to quantify infringements of security policies, using the identified factors. We demonstrate the ability of the revised metric to discriminate privacy infringements by applying it to a case study on the provisioning of healthcare treatment.

To validate the proposed framework and metric, we developed an analysis tool for compliance checking. The tool makes it possible to assess the compliance of audit trails with process models using a number of conformance metrics. We use this tool to evaluate the quality of results obtained using different metrics and factors.

The remainder of the paper is structured as follows. Section II discusses the motivations for this work and introduces our running example. Section III presents a comparative analysis of existing metrics for measuring deviations from process specifications. Section IV introduces the main privacy factors that should be considered in the design of conformance metrics. Section V presents an enhanced conformance metric for measuring the severity of privacy infringements. Section VI presents a tool for the analysis and comparison of conformance metrics. Finally, Section VII concludes the paper, providing directions for future work.

II. Motivations

Nowadays, several organizations including healthcare providers employ security policies and procedures to provide high quality services to customers and minimize privacy risks in handling customer personal data. These policies and procedures serve as guidelines for the employees, specifying best practices and the sequence of activities that need to be performed in order to achieve organizational goals. Accordingly, in this work we assume that process specifications define the users' allowed behavior.

While significant research exists on enforcement mechanisms [6]–[8], the basic notion of enforcement relies on the idea that *infringements* (i.e., deviations from policies and procedures) are violations and as such should not be permitted. However, in domains like healthcare, it may be necessary to do things differently from the specification in order to react to unpredictable circumstances (e.g., emergency situations). Preventing such actions may be critical for the patient's life. At the same time, it is not possible to enumerate all possible exceptions at design time.

As a result most healthcare systems include "break-the-glass" mechanisms which allow a user to bypass access control rules in such situations. However, break-the-glass mechanisms are a weak point in the system and can be misused [9]. Therefore, it is necessary to be able to distinguish acceptable from unacceptable infringements. More generally, infringements can pose different risks depending on the particular deviation from policies and procedures. A measurement of the *severity* of infringements requires analyzing the actual user behavior.

The actual user behavior is usually captured by the system in *audit trails*. When a user breaks the glass, his actions are monitored and recorded, thus making him accountable for his actions. Sample audit trails are then inspected manually for regulatory compliance. Manual auditing, however, presents a number of drawbacks. First, it is costly and time consuming. Second, it is error-prone and may lead to potentially severe infringements to remain undetected. Finally, the infringement detection process is not transparent as the decision whether an infringement is acceptable or not depends on the auditor's judgment. Transparency implies that decisions should be replicable, so that the system can eventually distinguish acceptable infringements from unacceptable infringements. Therefore, to ensure regulatory compliance, it is advisable to develop automated tools able to analyze the user behavior, detect infringements, and prioritize them according to their severity. To this end, we need *metrics* (the focus of this paper) able to measure the severity of infringements.

To illustrate the possible privacy infringements that may occur, let us consider a simple healthcare treatment process. Fig. 1 presents the process in Business Process Model and Notation (BPMN). The process involves four actors: *patient*, *receptionist*, *doctor*, and *lab technician*. Each actor is represented by a *swimlane* which contains the *tasks* to be performed by the actor. Tasks are denoted by a rounded corner rectangle. The execution of a task may require access to *data objects*. Data objects are represented by a rectangle with a folded top-right corner and connected to the tasks

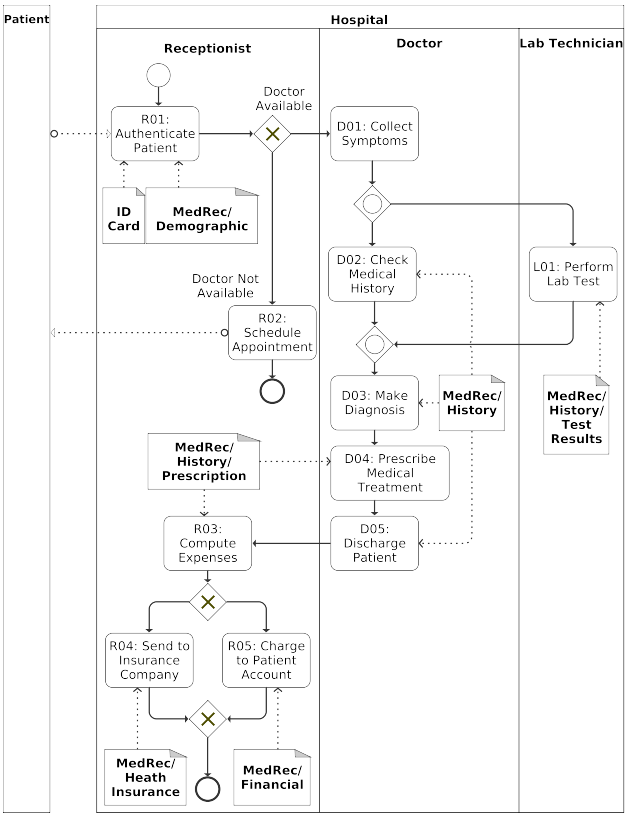

Figure 1. Healthcare Treatment Process

that use them through a dotted line. The process describes the sequence flow, i.e. the flow of a token from a *start event* (denoted by a narrow-border circle) to an *end event* (denoted by wider-border circle) through tasks and gateways. *Gateways* (represented by a diamond marked with a distinctive symbol denoting the type of the gateway) are used to control how the token(s) flows in the business process model.

When the patient arrives at the hospital, some of her personal data need to be disclosed for authentication (R01). Each registered patient has an electronic medical record (*MedRec*) stored in the information system of the hospital. Medical records have a folder structure and store all administrative, financial and clinical regarding a patient. Hospital employees can access only certain parts of a record depending on their role and for well defined purposes.

After the patient is successfully identified, the receptionist checks the availability of the doctor. If no doctors are available, an appointment is scheduled for a later time (R02). Otherwise, the patient is examined by the doctor who collects the patient's symptoms (D01). Afterward, the doctor checks the patients' medical history for other related illnesses (D02) or sends some biological samples to the medical laboratory for analysis (L01). Subsequently, the collected information is used by the doctor to make a diagnosis (D03) and prescribe treatment (D04). Then, the doctor discharges the patient (D05), and the patient is sent to the receptionist for billing matters. After computing the bill for the provided services (R03), the receptionist can either charge the patient directly (R05) or send the bill directly to the patient's insurance company for reimbursement (R04), provided that the patient has a health insurance.

We now consider some possible audit trails for our scenario (Fig. 2) and discuss the severity level of privacy infringements that may occur. Each log entry in the audit trail records an event referring to the task performed by a user. In particular, it specifies the *user* who performed the *task* together with the *role* held by the user at that moment and the *data objects* which were accessed to perform the task. In Fig. 2, R stands for receptionist, D for doctor, and L for lab technician; tasks are identified by an identification number as defined in Fig. 1. For the sake of simplicity, we assume that each audit trail corresponds to a single execution of the process.

1) The audit trail in Fig. 2a is a valid trace of the process in Fig. 1. Accordingly, there is not a privacy violation.
2) The audit trail in Fig. 2b represents a situation in which the doctor (Bob) repeats some tasks (i.e., D02, D03, and D04). For instance, this situation may occur because new symptoms appeared after the doctor had prescribed a treatment. It is worth noting that the doctor accessed the same information that he had accessed previously. We assume that this deviation from the expected behavior is in the interest of the patient and, therefore, does not represent a severe privacy violation.
3) In the audit trail in Fig. 2c, the patient was discharged by receptionist (Alice) instead of by the doctor. Consequently, Alice accessed the patient's full medical history while executing that task. This situation can be seen as a privacy infringement because, according to the specification, the receptionist has only limited access to patients' medical information.
4) The audit trail in Fig. 2d presents a situation where Mallory, a doctor with prior convictions for fraud, accesses the patient's financial information, which he was not allowed to access. In addition, this is done by executing a task which is not related to the provisioning of healthcare treatments (i.e., D06: Assess financial credentials). Accordingly, this audit trail indicates a situation in which patient information has likely been misused; therefore, it should be qualified as a severe privacy infringement.

To discriminate infringements on the basis of their severity, we need metrics able to analyze audit trails against the policies and procedures defined by the organization and assess their deviation from the expected behavior. In the next section, we present a number of existing conformance metrics and study their applicability for measuring the severity of infringements.

#	User	Role	Task	Data Items Accessed
1	Alice	R	R01	{ID, MedRec/ Demographic}
2	Bob	D	D01	
3	Bob	D	D02	{MedRec/ History}
4	Charlie	L	L01	{MedRec/ History/ TestResults}
5	Bob	D	D03	{MedRec/ History}
6	Bob	D	D04	{MedRec/ History/ Prescription}
7	Bob	D	D05	{MedRec/ History}
8	Alice	R	R03	{MedRec/ History/ Prescription}
9	Alice	R	R04	{MedRec/ Health Insurance}

(a) Audit Trail 1

#	User	Role	Task	Data Items Accessed
1	Alice	R	R01	{ID, MedRec/ Demographic}
2	Bob	D	D01	
3	Bob	D	D02	{MedRec/ History}
4	Charlie	L	L01	{MedRec/ History/ TestResults}
5	Bob	D	D03	{MedRec/ History}
6	Bob	D	D04	{MedRec/ History/ Prescription}
7	Bob	D	D02	{MedRec/ History}
8	Bob	D	D03	{MedRec/ History}
9	Bob	D	D04	{MedRec/ History/ Prescription}
10	Bob	D	D05	{MedRec/ History}
11	Alice	R	R03	{MedRec/ History/ Prescription}
12	Alice	R	R04	{MedRec/ Health Insurance}

(b) Audit Trail 2

#	User	Role	Task	Data Items Accessed
1	Alice	R	R01	{ID, MedRec/ Demographic}
2	Bob	D	D01	
3	Bob	D	D02	{MedRec/ History}
4	Bob	D	D03	{MedRec/ History}
5	Bob	D	D04	{MedRec/ History/ Prescription}
6	Alice	R	D05	{MedRec/ History}
7	Alice	R	R03	{MedRec/ History/ Prescription}
8	Alice	R	R04	{MedRec/ Health Insurance}

(c) Audit Trail 3

#	User	Role	Task	Data Items Accessed
1	Alice	R	R01	{ID, MedRec/ Demographic}
2	Mallory	D	D06	{MedRec/ Financial}
3	Mallory	D	D02	{MedRec/ History}
4	Mallory	D	D03	{MedRec/ History}
5	Mallory	D	D04	{MedRec/ History/ Prescription}
6	Mallory	D	D05	{MedRec/ History}
7	Alice	R	R03	{MedRec/ History/ Prescription}
8	Alice	R	R05	{MedRec/ Financial}

(d) Audit Trail 4

Figure 2. Sample Audit Trails

III. Comparative analysis of conformance metrics

In the literature there exist a number of metrics that can be used to quantify the compliance of the user behavior captured in the audit trail with a given process model. Here they are classified into three categories:

- *binary metrics*: determine whether an audit trail is a valid execution of a process model or not [12], [13];
- *sequence distance metrics*: measure the distance between two sequences of events [11];
- *process fitness metrics*: measure the extent to which an audit trail can be associated with an execution path of the process model [14].

Notice that the metrics above operate with different inputs. In particular, sequence distance metrics compare sequences, whereas process fitness metrics compare a sequence of events with a process. Although binary metrics can operate with both inputs, without lack of generality we consider binary metrics that take as input two sequences.

To facilitate the comparison between these metrics, let us introduce the following notation. Let P be the set of business processes defined by an organization, and Σ the set of observable tasks in P. A *trace* is a finite sequence over Σ; the set of traces is denoted by Σ^*. $T(p) \subseteq \Sigma^*$ denotes the set of traces generated by a process $p \in P$.

- *Process metrics*, denoted $\overline{d} : P \times \Sigma^* \rightarrow \mathbb{R} \cup \{\infty\}$, indicate functions that measure the conformance of a trace w.r.t. a process model.
- *Sequence metrics*, denoted $d : \Sigma^* \times \Sigma^* \rightarrow \mathbb{R} \cup \{\infty\}$, indicate functions that measure the distance between two traces.

Symbol ∞ denotes that the inputs are incomparable. In addition, we indicate the empty trace as $\cdot$; given two traces $\alpha, \sigma \in \Sigma^*$, their concatenation is represented as $\alpha\sigma$.

Given a process $p \in P$ and a trace $\sigma \in \Sigma^*$, binary and sequence distance metrics measure the conformance of σ w.r.t. p as follows

$$\overline{d}(p, \sigma) = \min\{d(\tau, \sigma) | \forall \tau \in T(p)\}$$

i.e., the conformance of σ w.r.t. p is the minimum sequence distance between σ and any trace τ generated by process p.

In the remainder of this section, we present the categories of metrics introduced above. Subsequently, we apply them to the example traces and specification given in Section II in order to determine whether they are appropriate for measuring privacy infringements. Notice that in this paper we only consider finite sequences of events. In addition, we assume that compliance is checked after the process execution has terminated. Accordingly, if the process specification allows for more tasks to be executed, an infringement is detected.

A. Binary Metrics

Binary metrics [12], [13] are the simplest type of metric for conformance checking. Intuitively, they return a positive answer if the audit trail is conform to the specification; otherwise, a negative answer is returned. To be able to compare binary metrics with other types of distance metrics,

we use 0 to represent positive answers and ∞ to represent negative answers.

Let $a\sigma$ be a trace representing the expected behavior and $b\sigma'$ the actual execution trace (with $a, b \in \Sigma$ and $\sigma, \sigma' \in \Sigma^*$). Binary metrics calculate the distance between $a\sigma$ and $b\sigma'$ as follows:

$$d_B(a\sigma, b\sigma') = \begin{cases} 0 & \text{if } a\sigma = \cdot \text{ and } b\sigma' = \cdot \\ \infty & \text{if } a\sigma = \cdot \text{ xor } b\sigma' = \cdot \\ d_B(\sigma, \sigma') & \text{if } a = b \\ \infty & \text{if } a \neq b \end{cases}$$

Notice that distance d_B is ∞ in the case where the two strings have a different length.

B. Sequence Distance Metrics

Sequence distance metrics have been proposed for measuring the amount of difference between two sequences, i.e., the number of changes needed to transform one sequence into the other. In this section, we present three sequence distance metrics that were used in [11] to quantify the deviation of user behavior from a security policy (defined as a set of traces).

The *suppressing distance*, d_S, counts the number of actions to be removed from the trace representing the actual behavior, $b\sigma'$, to obtain the trace representing the expected behavior, $a\sigma$. This metric can be formalized as follows:

$$d_S(a\sigma, b\sigma') = \begin{cases} \infty & \text{if } a\sigma \neq \cdot \text{ and } b\sigma' = \cdot \\ |b\sigma'| & \text{if } a\sigma = \cdot \\ d_S(\sigma, \sigma') & \text{if } a = b \\ 1 + d_S(a\sigma, \sigma') & \text{if } a \neq b \end{cases}$$

Notice that the distance from the actual behavior to the expected behavior is ∞ in the case $|a\sigma| > |b\sigma'|$ because no number of suppressions can transform $b\sigma'$ into $a\sigma$.

The *replacing distance*, d_R, counts the number of replacements necessary to obtain one trace from the other. The formula for this metric is:

$$d_R(a\sigma, b\sigma') = \begin{cases} 0 & \text{if } a\sigma = \cdot \text{ and } b\sigma' = \cdot \\ \infty & \text{if } a\sigma = \cdot \text{ xor } b\sigma' = \cdot \\ d_R(\sigma, \sigma') & \text{if } a = b \\ 1 + d_R(\sigma, \sigma') & \text{if } a \neq b \end{cases}$$

Notice that distance d_R is ∞ if the two input strings have a different length.

The *Levenshtein distance*, d_L, counts the number insertions, suppressions and replacements needed to obtain one trace from the other. The analytical formula can be represented as follows:

$$d_L(a\sigma, b\sigma') = \begin{cases} |b\sigma'| & \text{if } a\sigma = \cdot \\ |a\sigma| & \text{if } b\sigma' = \cdot \\ d_L(\sigma, \sigma') & \text{if } a = b \\ 1 + \min(d_L(\sigma, \sigma'), d_L(a\sigma, \sigma'), d_L(\sigma, b\sigma')) & \\ & \text{if } a \neq b \end{cases}$$

C. Process Fitness Metrics

Differently from sequence distance metrics, process fitness metrics assess the degree of compliance of an audit trail with a process model using the process structure rather than comparing the audit trail against the traces that can be generated from the process model. An example of process fitness metric is proposed in [14]. Here, the distance from an audit trail to the process specification is measured by simulating the ordered sequence of tasks recorded in the audit trail in the process model. Simulation is accounted for using tokens. A token is initially *produced* by the start event of the process. Tokens are *consumed* by the execution of a task or by an end event. The execution of a task in the audit trail can be simulated within the process model if it has a token right before it. If such a token does not exist, a *missing* token (i.e., a token created artificially to allow the execution of the process; note that missing tokens do not count as produced tokens) is added and the task is simulated. The execution of the task produces a new token as output. Note that the fitness metric in [14] is based on Petri Nets. Here, gateways are not explicitly modeled; intuitively, they are modeled as multiple sequence flows that converge on or diverge from a task. Accordingly, a task can produce or consume more than one token depending on the semantics of the connected gateway. The simulation terminates when the audit trail is completely analyzed. At the end of the simulation, tokens that have not been consumed are considered *remaining* tokens.

Fitness is assessed by computing an average between the ratio of missing and consumed tokens, and the ratio of remaining and produced tokens. The metric is given by the following formula:

$$d_F(P, l) = \frac{1}{2}\left(\frac{m}{c} + \frac{r}{p}\right)$$

where:

- m is the number of missing tokens;
- r is the number of remaining tokens;
- c is the number of consumed tokens;
- p is the number of produced tokens.

Note that $m \leq c$ and $r \leq p$, and therefore $0 \leq d_F(P, l) \leq 1$ (0 indicates that the trace perfectly fits the process model).

D. Discussion

In this section, we use the metrics presented in the previous sections to measure the distance between the audit trails in Fig. 2 and the process in Fig. 1. A summary of the results is presented in Table I. The columns denoted by *T* represent the case where only the task is used to analyze the audit trails against the process specification.

Binary metrics offer the lowest performance with regard to the quantification of privacy infringements. They can merely discriminate between fully compliant audit trails and non-compliant ones. Accordingly, this type of metrics does

Table I
SEVERITY OF INFRINGEMENT FOR THE AUDIT TRAILS IN FIG. 2

Audit Trail	$\overline{d}_B$		$\overline{d}_S$		$\overline{d}_R$		$\overline{d}_L$		$\overline{d}_F$
	T	TR	T	TR	T	TR	T	TR	T
Fig. 2a	0	0	0	0	0	0	0	0	0
Fig. 2b	∞	∞	3	3	∞	∞	3	3	1/10
Fig. 2c	0	∞	0	∞	0	1	0	1	0
Fig. 2d	∞	∞	∞	∞	1	1	1	1	1/7

not provide a prioritization of privacy infringements and therefore is not suitable for privacy compliance verification.

Sequence distance metrics have a better discrimination power then binary metrics as showed in Table I. However, the measures obtained using sequence distance metrics are meaningless from a privacy perspective. Such metrics return a severity of infringements that corresponds to the number of steps that do not match the specification and therefore give no indication about the severity of the privacy violation that occurred. For instance, the severity of infringement of a trace with a large number of deviations, which are however negligible from a privacy perspective, is much higher than the severity of infringement of a trace with few, but very serious infringements. This can be seen in Table I by comparing the results obtained using the Levenshtein distance on the audit trails of Fig. 2b and Fig. 2d. Nevertheless, one can clearly notice that the Levenshtein distance provides more accurate values than the other two sequence distance metrics. Actually, the Levenshtein distance combines the features of both suppressing and replacing distances with insertions.

Process fitness metrics seem to offer a slightly better quantification than sequence distance metrics. However, this type of metrics (e.g., [14]) is often based on formalisms such as Petri Nets that are not able to capture the full complexity of business process specifications [19]. Moreover, this type of metrics requires that every task recorded in the audit log belongs to the process model, which makes it impossible, for instance, to analyze the audit trail of Fig. 2d. To overcome this issue, we added the additional task (i.e., D06) to the process specification in isolation (i.e., not connected to any other element) and then counted the consumed and missing token that this insertion causes.

It is worth noting that the audit trail in Fig. 2c appears to be a valid trace according to all the conformance metrics. This is due to the fact that these metrics only consider the performed tasks when checking compliance. We repeated the analysis by taking into account both the task and the role held by the user who performed the task as factors for detecting infringements and determining their severity. The results are in the columns denoted by TR in Table I. One can see that most metrics are able to detect the infringement; however, the obtained measures suffer the same drawbacks discussed above. We did not repeat the assessment of severity of infringements using process fitness metrics. This is due to intrinsic difficulties in extending the compliance method in [14] to deal with roles. A discussion of how this method can be extended, however, is out of the scope of this work.

In summary, all these metrics are not useful in practice for measuring the severity of privacy infringements. The main drawback of these metrics is that any deviation from the process model carries the same weight. However, our experiments show that considering additional factors (e.g., the role held by the user at the time the task was performed) helps to better discriminate privacy infringements. In the next section, we identify and discuss the factors that can be used to assess the severity of privacy infringements.

IV. PRIVACY FACTORS

The application of existing compliance metrics to our case study has shown that considering only information about the performed tasks is not sufficient to assess the severity of infringements; additional information has to be considered. However, this information should be available for analysis. Audit trails offer some important information in this sense. They usually record the task which was performed, the user who performed it, the role held by the user and the data being accessed. Accordingly, we have identified three factors for assessing the severity of infringements: (i) user factor, (ii) action factor, and (iii) data factor. In the remainder of this section, we discuss these factors.

A. Data Factor

Personal data play a central role in privacy. Privacy and data protection regulations impose that personal data are relevant with respect to the purpose for which they are collected and processed [4]. In our scenario, doctors should only access data relevant for medical treatment. However, this is not the case in the audit trail of Fig. 2d in which Mallory accessed the financial information of the patient. Since there is no need for such information in the provision of healthcare treatment, this qualifies Mallory's behavior as a privacy violation. This simple example shows the importance of determining which data items have been accessed in the detection and investigation of privacy infringements.

However, certain data items may be particularly sensitive for the data subject. Therefore, the disclosure of different data items should not count equally. There exist a number of qualitative and quantitative approaches which aim to quantify the sensitivity of personal data in order to regulate their disclosure. For example, in [20] users can declare their privacy preferences by specifying if data can be disclosed freely, on a need to know basis, or if they should not be given. In [21], users can specify privacy penalties that represent the cost of disclosing information.

To quantify unauthorized access to data and therefore the amount of privacy loss caused by an infringement, we introduce the notion of *privacy weight*. Similarly to [21], privacy weights are data subject preferences and represent the cost of using a certain personal data item in the execution of the process according to his judgment. In particular, the

higher the privacy weight of a data item is, the less a user wishes to disclose that item.

B. User Factor

Another main factor for assessing the severity of infringements is the user who performed the task. Indeed, when the break-the-glass mechanism is invoked, any user can gain access to patient data. As shown in Section III-D, some infringements can be detected by only considering the user who executed the task. In particular, the *role* that the user held at the time the task was performed allows for a more accurate detection of privacy infringements. A role describes job functions and responsibilities within an organization and is usually associated with the set of access rights necessary to achieve assigned duties. If a task is performed by a user that held a role different from what is defined in the specification, there is a risk of data misuse and therefore this situation should be accounted for as an infringement.

However, not all the situations in which the task is executed by a user holding a role different from the one specified in the process specification may present the same risk level. For instance, in our scenario, only doctors and lab technicians can access test results. Suppose now that a nurse and a receptionist accessed test results. The severity of the infringement should be higher in the second case. Indeed, a nurse could have accessed test results because a patient needs immediate attention and no doctors are available. Therefore, the semantic distance between the role of the user actually executing the task and the role associated to the task in the specification should be considered to better quantify deviations from the process specification.

Several existing methods can be used to compute the semantic similarity between two terms. For instance, *depth length similarity* [22] uses the information content of the least common subsumer. *Path length similarity* [22] computes the shortest path between two concepts in an ontology. *Syntactic similarity* [23] uses lexical databases such as WordNet [24] together with the Jaro distance [25]. These similarity measures can be used and combined to yield more precise values. For instance, *context similarity* measures the similarity of two domain ontologies by considering the similarity of their root nodes. *Neighborhood similarity* is computed by taking the average of similarity between the parents and children of two concepts. In this paper, we use the *Latent Semantic Analysis* (LSA) [26], a semantic relatedness metric which uses a high-dimensional linear associative model to assess the similarity of words.

Another important aspect for quantifying privacy infringements is the trustworthiness of the user who performed the task. Intuitively, deviations from the specification caused by untrusted users present a higher risk of privacy loss. Consider, for instance, the audit trail in Fig. 2d. Here, Mallory, who had previously been convicted for fraud, accessed the patient's financial information. Given Mallory's past behavior, he is most likely to use this information for illegitimate purposes. To capture these concerns in the assessment of the severity of infringements, we employ the concept of *reputation*. Reputation is a measure of the trustworthiness of users based on their past behavior [27]. This measure should be used to boost the severity of the infringement. In particular, the lower the reputation of the user is, the higher the risk of privacy loss is.

C. Action Factor

The action factor refers to the tasks performed by users during the execution of a process. A user may not necessarily follow the process specification, for instance, in emergency situations. As discussed in Section III, existing conformance metrics consider this factor when checking compliance. However, these metrics evaluate every deviation equally. A more accurate measurement of the severity of privacy infringements requires taking into account which tasks have been executed by the users.

Consider, for example, the process in Fig. 1 and the correct audit trail in Fig. 2a. If the doctor, instead of executing the task of prescribing a medical treatment (D04), executes a slightly different task (e.g., providing the medical treatment using the medical supplies available in his own office), this would certainly not qualify as a serious privacy infringement. In contrast, if the doctor executes a significantly different task (instead of D04) like assessing the financial credentials of the patient, this should be accounted for as a severe privacy infringement.

This simple example shows that the semantic difference between the task defined in the process specification and the task which has actually been performed has a considerable impact on the severity of the infringement. The semantic distance between two tasks can be computed using the measures of semantic relatedness used to compute semantic distance between two roles.

V. Measuring Infringements

In this section, we present how sequence distance metrics and, in particular, the Levenshtein distance can be improved in order to assess the severity of privacy infringements using the privacy factors presented in the previous section. We chose the Levenshtein distance because it is the sequence distance metric that better discriminates infringements (Section III). However, the other sequence distance metrics can be extended in a similar way.

The Levenshtein distance, as well as other conformance metrics, only considers the tasks when quantifying the deviation of the actual user behavior from the expected behavior. As a result, any deviation from the specification is counted as 1, making the obtained values meaningless from a privacy perspective. The basic idea is to replace the constant factor 1 in the Levenshtein distance with the extent of the deviation obtained using the privacy factors.

To this end, a conformance metric should take as input all available information about the actual user behavior and process specification. We assume that the user behavior is recorded in an audit trail. As described in Section II, each entry of the audit trail includes information about the task that was executed, the user who executed it, the role held by the user at the time the task was executed, and the data items accessed during its execution. The expected behavior is described by the set of valid traces, that is, the traces that can be generated by the process model. We assume that every element of the valid traces describes the task to be executed together with the role that should be held by the user and the data items to be used in the task execution.

Now, we provide the notation used in the metric:

- $r \in [0,1]$ is the reputation of the user performing the task where 1 means very good and 0 very bad.
- $s_T \in [0,1]$ is the semantic distance between the task which is actually executed and the task defined in the process specification; 0 means that the two tasks are semantically equivalent and 1 that they are completely incompatible.
- $s_R \in [0,1]$ is the semantic distance between the role of the user executing the task and the role associated to the task in the specification; 0 means that the two roles are semantically equivalent and 1 that they are completely incompatible.
- $p \in \mathbb{R}^+$ represents the penalty due to unauthorized access to data during the execution of a task.

The computation of p requires considering three sets of data items:

- A: the set of data items that a user is allowed to access in order to execute the task. In particular, A is equal to the set of data items linked to the task in the process model if the user holds the role associated to the task in the model. Otherwise, A is the empty set; indeed, in this case the user should not have executed the task and, therefore, should not have accessed the data.
- B: the set of data items accessed during the actual execution of the task.
- K_u: the set of data items previously accessed by user u.

Intuitively, penalty p is the sum of the privacy weights (δ_i) of the data items accessed during the actual execution of the task which were not supposed to be accessed according to the specification and were not already accessed by the user. This intuition can be formally represented as

$$p = \sum_{\delta_i \in B \setminus (A \cup K_u)} \delta_i.$$

As done in sequence distance metrics, we analyze an element of a valid trace and an entry of the audit trail at a time. Given an element of a valid trace a and an entry of the audit trail b, the severity of the infringement $\Phi(a,b)$ is defined as follows:

$$\Phi(a,b) = (c_1 - c_2 r)[(1 + c_3 s_R)(1 + c_4 s_T)(1 + c_5 p) - 1]$$

where $c_i \in \mathbb{R}^+, i \in \{1, ..., 5\}$ are constants and $c_1 > c_2$. The rationale of this metric is that the role distance, task distance and sum of privacy weights determine the severity of the infringement. The relevance of these factors on the severity is determined by constants c_3, c_4 and c_5, respectively. The reputation of the user performing the task scales the effects of all other inputs. The constraint $c_1 > c_2$ is needed to capture the infringement when a user with high reputation ($r = 1$) deviates from the specification.

The severity of an infringement recorded in an audit trail can be assessed by combining the Levenshtein distance with metric Φ. The revisited Levenshtein distance can be represented as follows:

$$d_L^\Phi(a\sigma, b\sigma') = \begin{cases} \Phi(\cdot, b) + d_L^\Phi(a\sigma, \sigma') & \text{if } a\sigma = \cdot \\ \Phi(a, \cdot) + d_L^\Phi(\sigma, b\sigma') & \text{if } b\sigma' = \cdot \\ d_L^\Phi(\sigma, \sigma') & \text{if } a = b \\ \Phi(a,b) + \min(d_L^\Phi(\sigma, \sigma'), d_L^\Phi(a\sigma, \sigma'), d_L^\Phi(\sigma, b\sigma')) & \text{if } a \neq b \end{cases}$$

where $a\sigma$ represents a valid trace and $b\sigma'$ the audit trail. The first two rules address the situation in which the two sequences have a different length. To compare the expected and actual behavior in this situation, we introduce a "dummy" element which is maximally distant from any other role and task (the last row and column in Tables III and IV). Similarly, when an additional task is expected to be executed (second rule), we assume that such a task is executed by an untrusted user.

To show the measurement accuracy of our metric, we applied it to the example scenario presented in Section II. The values of the factors used in the experiment are presented in Tables II to V. In particular, Table II defines the reputation of users, Table III the semantic distance between roles, Table IV the semantic distance between tasks, and Table V the privacy weights of data items. The values in Tables III & IV were computed using an implementation of LSA provided by the Rensselaer MSR project.[1] In particular,

Table II
REPUTATION OF HOSPITAL EMPLOYEES

User	Reputation value	Comment
Alice	0.5	newly hired receptionist
Bob	0.9	highly appreciated doctor
Charlie	0.6	hired one year ago
Mallory	0.1	previous fraud convictions

Table III
SEMANTIC DISTANCE BETWEEN ROLES

Roles	Receptionist	Doctor	Lab Worker	·
Receptionist	0.0	0.8	0.6	1.0
Doctor	0.8	0.0	0.5	1.0
Lab Worker	0.6	0.5	0.0	1.0
·	1.0	1.0	1.0	1.0

[1] http://cwl-projects.cogsci.rpi.edu/msr/

Table IV
SEMANTIC DISTANCE BETWEEN TASKS

	R01	R02	R03	R04	R05	D01	D02	D03	D04	D05	D06	L01	·
R01	0.0	0.8	1.0	1.0	0.5	0.6	0.6	0.8	0.3	0.0	0.9	0.9	1.0
R02	0.8	0.0	0.8	0.7	0.6	0.8	0.7	0.7	0.8	0.8	0.7	0.8	1.0
R03	1.0	0.8	0.0	0.6	0.6	0.8	0.8	0.8	0.8	0.9	0.5	1.0	1.0
R04	1.0	0.7	0.6	0.0	0.7	0.8	0.7	0.6	0.8	1.0	0.7	0.9	1.0
R05	0.5	0.6	0.6	0.7	0.0	0.6	0.5	0.6	0.6	0.5	0.6	0.8	1.0
D01	0.6	0.8	0.8	0.8	0.6	0.0	0.6	0.7	0.4	0.6	0.8	0.8	1.0
D02	0.6	0.7	0.8	0.7	0.5	0.6	0.0	0.7	0.4	0.6	0.8	0.8	1.0
D03	0.8	0.7	0.8	0.6	0.6	0.7	0.7	0.0	0.8	0.8	0.8	0.8	1.0
D04	0.3	0.8	0.8	0.8	0.6	0.4	0.4	0.8	0.0	0.3	0.8	0.8	1.0
D05	0.0	0.8	0.9	1.0	0.5	0.6	0.6	0.8	0.3	0.0	0.9	0.9	1.0
D06	0.9	0.7	0.5	0.7	0.6	0.8	0.8	0.8	0.8	0.9	0.0	0.9	1.0
L01	0.9	0.8	1.0	0.9	0.8	0.8	0.8	0.8	0.8	0.9	0.9	0.0	1.0
·	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table V
PRIVACY WEIGHTS

Data Item	Privacy weights
ID Card	3
MedRec/ Demographic	2
MedRec/ History	10
MedRec/ History/ TestResults	7
MedRec/ History/ Prescription	6
MedRec/ HealthInsurance	2
MedRec/ Financial	10

Table VI
SEVERITY OF INFRINGEMENTS FOR THE AUDIT TRAILS IN FIG. 2

Audit Trail	$\overline{d}_L$		$\overline{d}_F^{\Phi}$
	T	TR	
Fig. 2a	0	0	0
Fig. 2b	3	3	0.34
Fig. 2c	0	1	11.28
Fig. 2d	1	1	18.8

the semantic distance between two words is computed as 1 minus the similarity value obtained using LSA. The constants used in metric d_L^{Φ} were set as follows: $c_1 = 1.1$, $c_2 = c_3 = c_4 = c_5 = 1$.

The last column of Table VI presents the measures obtained using the proposed metric ($\overline{d}_L^{\Phi}$). Remark that $\overline{d}_L^{\Phi}$ is the minimum sequence distance d_L^{Φ} between the audit trail and any trace generated by the process. Compared with the measures obtained using the Levenshtein distance ($\overline{d}_L$ in Table VI), our metric provides more accurate measures of the severity of infringements. As discussed in Section II, the audit trail in Fig. 2b is a mere repetition of some tasks done by a user playing the correct role. Accordingly, our metric only reports a slight deviation from the process specification. On the other hand, the audit trail in Fig. 2c represents the case where the receptionist, who discharged the patient instead of the doctor, accessed the patient's medical record illegitimately. Differently from other metrics, our metric quantifies it as a severe privacy infringement because of the unauthorized access to data. Similarly, the trail in Fig. 2d represents the situation in which a crooked doctor accessed the patient's financial information without any professional reason. Due to the low reputation of the doctor, the metric reports an even more severe infringement than the one detected in the prior audit trail.

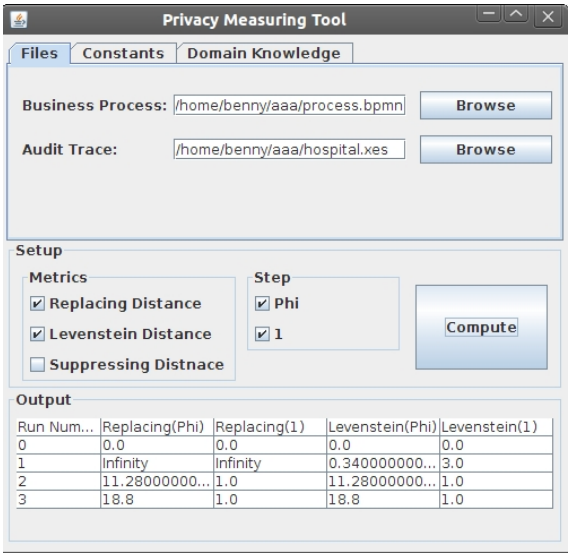

Figure 3. Graphical User Interface

VI. IMPLEMENTATION

To evaluate conformance metrics and privacy factors, we have implemented an analysis tool. The tool provides a graphical user interface (GUI) through which one can set up the input for the analysis and get the results of the evaluation. The GUI is divided into three main areas: a menu for the inputs of the analysis at the top, a menu for selecting the metrics to be used in the middle, and an area at the bottom where the results of the analysis are reported (Fig. 3).

The tool accepts as input a process specification generated by BPMN Modeler[2] and an audit log provided in the eXtensible Event Steam[3] (XES) format (first tab) together with the value of the constants used in metric Φ (second tab) and domain knowledge (third tab). In particular, the third tab allows the specification of the reputation of users as well as the privacy weights of the data items. The semantic distance between two terms can be either precomputed or stored in a database or it can be computed on-the-fly using the Rensselaer MSR API.

At the moment, the tool supports the three sequence distance metrics discussed in this paper; more than one metric can be used at the same time for comparison. In addition, one can select either "*Phi*" or "1" to be used in the analysis or both. Using the first option, the tool uses metric Φ to quantify deviations from the specification. Using the other option, the tool counts 1 for every deviation from the specification (i.e., the task, role and data items in the audit trail differ from the ones specified in the process model).

The tool generates all possible valid traces of the process model (avoiding loops), and then it evaluates the audit log against these traces using the selected metrics. For each

[2]http://www.eclipse.org/bpmn/
[3]http://www.xes-standard.org/

selected metric, the minimum distance is reported in the table at the bottom of the GUI.

VII. Conclusions

In this paper, we have identified a number of factors which can be used to quantify deviations of the actual user behavior from process specifications. We have also discussed how these factors can be accommodated into sequence distance metrics by enhancing the Levenshtein distance. The application of the proposed metric to the case study showed that the identified factors allow for a more accurate discrimination of privacy infringements.

The work presented in this paper provides interesting directions for future work. The Levenshtein distance and, in general, sequence distance metrics require generating all possible traces from the process model in order to compute the severity of infringements. This, however, limits the applicability of the approach as the number of traces can be infinite, for instance when the process has a loop. To this end, we are investigating how to accommodate the identified factors into process fitness metrics which use the process structure for compliance checking.

In the experiments presented in this paper, we considered every factor equally. However, different factors may have a different impact on privacy. We are applying the proposed metric to a number of case studies in order to empirically determine settings for the constants used in metric Φ. Finally, we are extending the analysis tool to include other conformance metrics for a more exhaustive comparison.

Acknowledgments: The authors thank Boris Škorić for his valuable comments and suggestions. This work has been partially funded by the EU-IST-IP-216287 TAS^3 project.

References

[1] Institute of Medicine, *To Err Is Human: Building a Safer Health System*. The National Academies Press, 1999.

[2] R. Noffsinger and S. Chin, "Improving the delivery of care and reducing healthcare costs with the digitization of information," *JHIM*, vol. 14, pp. 23–30, 2000.

[3] "2009 Annual Study: Global Cost of a Data Breach," 2010. [Online]. Available: http://www.securityprivacyandthelaw.com/uploads/file/Ponemon_COB_2009_GL.pdf

[4] P. Guarda and N. Zannone, "Towards the development of privacy-aware systems," *Inf. Softw. Technol.*, vol. 51, no. 2, pp. 337–350, 2009.

[5] P. Ashley, S. Hada, G. Karjoth, and M. Schunter, "E-P3P privacy policies and privacy authorization," in *Proc. of WPES*. ACM, 2002, pp. 103–109.

[6] J. Ligatti, L. Bauer, and D. Walker, "Run-time enforcement of nonsafety policies," *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 3, pp. 1–41, 2009.

[7] K. W. Hamlen, G. Morrisett, and F. B. Schneider, "Computability classes for enforcement mechanisms," *ACM Trans. Program. Lang. Syst.*, vol. 28, no. 1, pp. 175–205, 2006.

[8] F. B. Schneider, "Enforceable security policies," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 1, pp. 30–50, 2000.

[9] C. A. Ardagna, S. D. C. di Vimercati, T. Grandison, S. Jajodia, and P. Samarati, "Regulating exceptions in healthcare using policy spaces," in *Proc. of Conf. on Data and Applications Security*, ser. LNCS, vol. 5094. Springer, 2008, pp. 254–267.

[10] C. Lovis, S. Spahni, N. Cassoni, and A. Geissbuhler, "Comprehensive management of the access to the electronic patient record: Towards trans-institutional networks," *Int. J. of Medical Informatics*, vol. 76, pp. 466–470, 2007.

[11] N. Bielova and F. Massacci, "Predictability of enforcement," in *Proc. of ESSoS*, ser. LNCS, vol. 6542. Springer, 2011, pp. 73–86.

[12] F. Chesani, P. Mello, M. Montali, F. Riguzzi, M. Sebastianis, and S. Storari, "Checking compliance of execution traces to business rules," in *Proc. of BPI*, ser. LNBIP, vol. 17. Springer, 2009, pp. 134–145.

[13] M. Petković, D. Prandi, and N. Zannone, "Purpose Control: Did You Process the Data for the Intended Purpose?" in *Proc. of 8th VLDB Workshop on Secure Data Management*, ser. LNCS, vol. 6933. Springer, 2011, pp. 145–168.

[14] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Inf. Syst.*, vol. 33, no. 1, pp. 64–95, 2008.

[15] H. H. Feng, O. M. Kolesnikov, P. Fogla, W. Lee, and W. Gong, "Anomaly detection using call stack information," in *Proc. of IEEE Symp. on Security and Privacy*. IEEE, 2003, pp. 62–75.

[16] N. Li and T. Li, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," in *Proc. of ICDE*. IEEE, 2007, pp. 106–115.

[17] C. Dwork, "Differential privacy," in *Proc. of ICALP*, ser. LNCS, vol. 4052. Springer, 2006, pp. 1–12.

[18] D. Clark, S. Hunt, and P. Malacaria, "A static analysis for quantifying information flow in a simple imperative language," *J. Comput. Secur.*, vol. 15, no. 3, pp. 321–371, 2007.

[19] D. Prandi, P. Quaglia, and N. Zannone, "Formal analysis of BPMN via a translation into COWS," in *Proc. of COORDINATION*, ser. LNCS, vol. 5052. Springer, 2008, pp. 249–263.

[20] A. Tumer, A. Dogac, and I. H. Toroslu, "A semantic-based user privacy protection framework for web services," in *Proc. of ITWP*, ser. LNCS, vol. 3169. Springer, 2003, pp. 289–305.

[21] F. Massacci, J. Mylopoulos, and N. Zannone, "Hierarchical hippocratic databases with minimal disclosure for virtual organizations," *VLDB J.*, vol. 15, no. 4, pp. 370–387, 2006.

[22] H. Nguyen and H. Al-Mubaid, "A Combination-based Semantic Similarity Measure using Multiple Information Sources," in *Proc. of IRI*. IEEE, 2006, pp. 617–621.

[23] L. D. Ngan, T. M. Hang, and A. Goh, "Semantic Similarity between Concepts from Different OWL Ontologies," in *Proc. of INDIN*. IEEE, 2006, pp. 618–623.

[24] G. A. Miller, "WordNet: a lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[25] M. A. Jaro, "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida," *J. of Amer. Statistical Assoc.*, vol. 84, no. 406, pp. 414–420, 1989.

[26] T. K. Landauer and S. T. Dumais, "Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge," *Psychological Review*, no. 104, 1997.

[27] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Commun. ACM*, vol. 43, no. 12, pp. 45–48, 2000.