Measuring Privacy Compliance using Fitness Metrics*

Sebastian Banescu[1,2], Milan Petković[1,2], and Nicola Zannone[2]

[1] Philips Research Eindhoven
{sebastian.banescu,milan.petkovic}@philips.com
[2] Eindhoven University of Technology
n.zannone@tue.nl

Abstract. Nowadays, repurposing of personal data is a major privacy issue. Detection of data repurposing requires posteriori mechanisms able to determine how data have been processed. However, current a posteriori solutions for privacy compliance are often manual, leading infringements to remain undetected. In this paper, we propose a privacy compliance technique for detecting privacy infringements and measuring their severity. The approach quantifies infringements by considering a number of deviations from specifications (i.e., insertion, suppression, replacement, and re-ordering).

Keywords: Process compliance management, Security in business processes

1 Introduction

Privacy protection is becoming a major issue in society nowadays. Advances in ICT allows the collection and storage of a huge amount of personal data. Those data are a valuable asset for business corporations. For instance, they can be used to provide customized services and create effective, personalized marketing strategies. However, such practices might be intrusive limiting customers' right to privacy.

The need to protect personal data has spurred the definition of several privacy policy languages. These languages usually extend access control with the concept of purpose to provide a more strict control on access and usage of data. Purpose is used to represent the reason for which data can be collected and processed. Data are usually labeled with the intended purposes and can be accessed only if the purposes for which data are requested is compliant with the label attached to the requested data. However, this approach is preventive and only guarantees policy compliance by preventing infringements to occur. Therefore, it is unacceptable when access to data should be granted to provide critical services such as medical treatment in emergency situations.

Recent privacy legislation recognizes the need of more comprehensive solutions to privacy and has an increased emphasis on accountability. The demand of technical means to enforce legislation requires the development of novel auditing systems which are able to ensure compliance with privacy policies and make users accountable for their actions. The development of such systems requires methods for determining whether the actual usage of personal data is compliant with the intended purpose.

* This work is funded by the Dutch national program COMMIT through the THeCS project and by the European Commission through the FP7 TClouds project (nr. 257243).

A first step to enable purpose control is to provide semantics to purposes, which defines how data can be used. Business process has been proved to be a suitable formalism for the definition of semantic purpose models [1]. This formalism makes it possible to verify user behavior against the intended purpose of data by verifying whether the user actions recorded in a log correspond to a valid trace of the process model corresponding to the intended purpose [1]. This approach to purpose control is more suitable for critical systems because it does not prohibit access to data in emergency situations in contrary to the preventive approaches. However, the corresponding existing solutions are only able to determine that a deviation occurred, but they are not able to identify particular deviations and to assess their privacy severity.

This paper proposes a conformance metric to detect privacy infringements and quantify their severity. In particular the conformance metrics is able to identify four different types of local deviations (i.e., insertions, suppressions, and replacements) and non-local deviations (i.e., re-ordering). Moreover, it adds a transition in isolation for each log event that has no corresponding transition in the process model. Therefore, the algorithm accepts a large range of logs as input. The detected infringements are then quantified using the privacy metric proposed in [2].

2 Privacy Compliance Checking

Deviations between an event log and a process model can have different types. Each type of deviation has a different privacy severity. In this paper, we consider four types of deviation: *insertion* occurs when an activity is executed, but its execution is not allowed by the process model; *suppression* occurs when an activity whose execution is required by the process model is not executed; *replacement* occurs when an activity is executed instead of another activity; *re-ordering* occurs when the order in which activities are executed is different than the order of the same activities specified in the process model.

To identify deviations from the specification, we propose a conformance checking technique based on [3]. In particular, the event log is replayed over the process model to identify user behavior which is not compliant with the intended behavior. Non compliant behaviors are represented in terms of *remaining* tokens (i.e., tokens that were left in the model after the execution of the process) and *missing* tokens (i.e., tokens created artificially for the successful execution of the process).

Conformance checking in [3], however, suffers from drawbacks in terms of accuracy [4]. First, the metric is sensitive to the process structure and does not consider all paths allowed by the process model. In addition, it does not identify the type of deviation that occurred for the measurement of fitness. Moreover, [3] assumes that all events in the log must have a counterpart in the process model, and every node must be on some path from the start to the end of the process. This assumption is too restrictive for our purposes. Indeed, we have to consider cases, for instance, where a user performed a task which he was not allowed to perform.

To overcome these drawbacks, we revise the technique in [3] in several ways. We associate a counter to tokens to represent the time in which an event occurred. This (together with the distinction between consumable and produced tokens) allows the definition of patterns for the identification of the type of deviation (Section 2.4). In

addition, we will add activities in isolation to the process model. Finally, conformance checking is performed on all alternative paths rather than on the entire process model.

The privacy compliance checking proposed in this paper comprises of five phases:

1. *Pre-processing* phase, in which alternative paths in the process model are identified;
2. *Simulation* phase, in which the event log is replayed in the process model;
3. *Pairing* phase, in which behaviors not compliant with the model are detected;
4. *Deviation identification* phase, in which non compliant behaviors are categorized according to the type of deviation;
5. *Quantification* phase, in which deviations are analyzed and quantified.

2.1 Pre-processing Phase

The pre-processing phase takes as input a business process represented as a CPN model (we refer to [5] for the notation for CPN models) and identifies alternative paths in the model. In particular, this phase consists of a non-disjoint "partitioning" of the process model into sub-processes such that each sub-process contains only paths defined in the original model and no two sub-processes contain the same path. The following phases of the algorithm are executed on each identified sub-process.

2.2 Simulation Phase

The goal of the simulation phase is to replay an event log λ over a CPN model N. To facilitate the pairing phase (Section 2.3) and the identification of the type of deviation (Section 2.4), we distinguish the set of consumable tokens (denoted by $Cons$) and the set of produced tokens (denoted by $Prod$). Tokens have the form of $(p, c)@\ell$, where p is the place containing the token and c is the token color. Each token has a counter ℓ associated to it, which is used to represent the sequence in which events occur.

The replay of λ starts with the initial marking. Then, logged events are fired one after another in N. Differently from [3], in our simulation approach, tokens are not consumed by transitions. Instead, every event e in λ leads to the generation of one consumable token in every place in $\bullet t$ and one produced token in every place in $t\bullet$, where t is the transition in e. If t does not belong to N, it is not performed by the intended role, or accessed data are not as intended, a new activity representing the executed event is added to the model in *isolation* (i.e., unconnected from the rest of the model). The output of the simulation phase is a new process model obtained by replaying λ over N together with marking $Cons \cup Prod$.

2.3 Pairing Phase

The aim of the pairing phase is to detect non conformant behaviors that occurred in the execution of the process model. The basic idea is to identify proper information flow by pairing produced and consumed tokens generated in the simulation phase. Intuitively, a consumable and a produced token match if (1) they are located in the same place, (2) they have equal token color, and (3) the counter of the produced token is lower or equal than the one of the consumable token. Paired tokens are removed from the

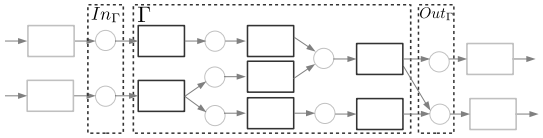

Fig. 1: Graphical representation of grouping

marking. The set of tokens that do not have a counterpart (hereafter called *unpaired tokens*) corresponds to non conformant behaviors. Unpaired tokens are used in the next phases to identify the type of deviation and quantify its severity.

2.4 Deviation Identification Phase

The aim of this phase is to identify the type of deviations that occurred. In particular, the deviation type is identified using patterns based on unpaired tokens. Before presenting the patterns and the identification process, we introduce the following notation.

Definition 1. *Let $N = (P, T, A, C, E, \mathbb{M}, \mathbb{Y}, M_0)$ be a CPN and $\Gamma \subset T$ a set of transitions. The* input boundary *of Γ, denoted by In_Γ, is the set of places $\Pi \subset P$ s.t. for every $p \in \Pi$, $(p,t) \in A$ iff $t \in \Gamma$, and $(t,p) \in A$ iff $t \notin \Gamma$.*

Definition 2. *Let $N = (P, T, A, C, E, \mathbb{M}, \mathbb{Y}, M_0)$ be a CPN and $\Gamma \subset T$ a set of transitions. The* output boundary *of Γ, denoted by Out_Γ, is the set of places $\Pi \subset P$ s.t. for every $p \in \Pi$, $(t,p) \in A$ iff $t \in \Gamma$, and $(p,t) \in A$ iff $t \notin \Gamma$.*

Definition 3. *Let $N = (P, T, A, C, E, \mathbb{M}, \mathbb{Y}, M_0)$ be a CPN. Given a set of transitions $\Gamma \subseteq T$ with input boundary In_Γ and output boundary Out_Γ, we say that Γ forms a* grouping *on N iff for all $p_x \in In_\Gamma$ and $p_y \in Out_\Gamma$ there exists a path from p_x to p_y s.t. all the transitions on the path belong to Γ.*

Fig. 1 represents a grouping together with its input and output boundaries. The grouping is delimited by a dashed rectangle. The places on the left and right sides of the dashed rectangle form the input and output boundaries of the grouping, respectively. Note that input and output boundaries do not have to be disjoint sets.

Deviation Identification Patterns In this section, we define four patterns based on the configuration of unpaired tokens for the identification of *insertions*, *suppressions*, *replacements* and *re-orderings*.

Insertion Pattern A set of transitions $\Gamma \subset T$ is an insertion if Γ is a grouping with input and output boundaries In_Γ and Out_Γ respectively, and the following conditions hold: (1) every place $p_i \in In_\Gamma$ contains a token $(p_i, c_i)@\ell_i \in Cons$; (2) every place $p_j \in Out_\Gamma$ contains a token $(p_j, c_j)@\ell_j \in Prod$; (3) for all i and j, $\ell_i \leq \ell_j$; (4) all places $p \in \{\bullet t \cup t \bullet \,|t \in \Gamma\} \setminus (In_\Gamma \cup Out_\Gamma)$ do not contain any token.

Suppression Pattern A set of transitions $\Gamma \subset T$ is a suppression if Γ is a grouping with input and output boundaries In_Γ and Out_Γ respectively, and the following conditions hold: (1) every place $p_i \in In_\Gamma$ contains a token $(p_i, c_i)@\ell_i \in Prod$; (2) every place $p_j \in Out_\Gamma$ contains a token $(p_j, c_j)@\ell_j \in Cons$; (3) all places $p \in \{\bullet t \cup t \bullet | t \in \Gamma\} \setminus (In_\Gamma \cup Out_\Gamma)$ do not contain any token.

Replacement Pattern A set of transitions $\Gamma_S \subset T$ are replaced by a set of transitions $\Gamma_I \subset T$ if the following conditions hold: (1) Γ_S satisfies the conditions of the suppression pattern with tokens $(p_i, c_i)@\ell_i \in Prod$ s.t. $p_i \in In_{\Gamma_S}$, and tokens $(p_j, c_j)@\ell_j \in Cons$ s.t. $p_j \in Out_{\Gamma_S}$; (2) Γ_I satisfies the conditions of the insertion pattern with tokens $(p_k, c_k)@\ell_k \in Cons$ s.t. $p_k \in In_{\Gamma_I}$ and tokens $(p_l, c_l)@\ell_l \in Prod$ s.t. $p_l \in Out_{\Gamma_I}$; (3) for all i, j, k and l, $\ell_i < \ell_k \leq \ell_l < \ell_j$;

Re-ordering Pattern A set of transitions $\Upsilon \subset T$ are re-ordered if the order in which transitions are executed is different than the order of the same transitions in the process model. A *re-ordering* is characterized by the existence of places $p \in P$ such that p contains tokens $(p, c)@\ell_i \in Prod$ and $(p, c)@\ell_j \in Cons$ s.t. $\ell_j < \ell_i$.

Pattern Identification Process Deviations are identified in two steps: first, insertions, suppressions and re-orderings are identified; then, replacements. Pattern detection starts from the token with the lowest counter in the marking. If more than one token have such a counter, a consumable token is chosen. The process model is traversed to identify (a set of) tokens which, in the combination with the selected token, satisfy a pattern. Identified patterns are stored into a *deviation list* Δ. A deviation in Δ is a tuple (δ, G, C, R) where δ is the deviation type associated to the pattern, and G, C and R are the sets of transitions, consumable and produced tokens that satisfy the pattern, respectively.

After an insertion or a re-ordering has been identified, the tokens forming the pattern are removed from the marking. After a suppression has been identified, the consumable tokens associated to it are used to fire the suppressed transitions. Such tokens flow in the process model until they reach a place containing a produced token associated to the currently identified suppression. If the counter of the consumable token is higher or equal to the one of the produced token, then the tokens are removed from the marking; otherwise, they indicate that a re-ordering also occurred, and such a deviation is added to the deviation list. To determine which transitions have been re-ordered, it is necessary to analyze the event log and token counters. Let p be a place that contains two tokens satisfying the re-ordering pattern. The corresponding re-ordered activities are identified by determining the longest path from p in the model such that it only contains transitions corresponding to events which occurred in the interval defined by the counters of the tokens used to identify the deviation.

If the selected token, in combination with (possibly) different sets of tokens, satisfies the conditions of more than one pattern, a copy of the deviation list for each pattern is created together with the corresponding marking; each identified pattern is stored in a separate deviation list. Then, the process continues on each marking and deviation list. This step ends when no insertion, suppression or re-ordering can be found.

Replacements are identified by checking deviations lists. In particular, a replacement is identified if there exist an insertion and a suppression in the deviation list such

that the insertion occurred within the suppression. If such a pattern is detected, the corresponding insertions and suppressions are removed from the list, and a replacement is added to the same list.

2.5 Quantification Phase

The quantification phase aims to measure the severity of the identified privacy infringements (if the severity is 0, no infringement occurred). The severity assessment is performed on each deviation list associated to a log using the metric d_L^{Φ} presented in [2]. This metric uses a number of factors (e.g., accessed data, executed action, user role) to measure the privacy distance between the intended and actual user behavior (see [2] for details). Every deviation list Δ is traversed separately, and d_L^{Φ} is applied to each deviation in the list. The *insertion* of a sequence of events $\xi \subset E$ is quantified against the empty sequence ϵ, i.e. $d_L^{\Phi}(\epsilon, \xi)$. The *suppression* of a sequence of activities $\chi \subset \mathcal{A}$ is quantified against an empty sequence ϵ, i.e. $d_L^{\Phi}(\chi, \epsilon)$. The *replacement* of a set of (suppressed) activities $\chi \subset \mathcal{A}$ by a sequence of (inserted) events $\xi \subset E$ is quantified as $d_L^{\Phi}(\chi, \xi)$. The *re-ordering* of a sequence of events Υ is quantified by considering the data accessed by events which occurred before they were supposed to have occurred. Let $\xi \subset \Upsilon$ be the sequence of events which occurred before they were supposed to have occurred. Re-ordering is quantified as $d_L^{\Phi}(\xi, \xi')$ where ξ' is equal to ξ except that the user's knowledge about data is the one specified in the process model.

3 Conclusions

In this paper we have proposed a privacy compliance technique for identifying privacy infringements and quantifying their severity. In particular, we have defined patterns for identifying four types of deviations, namely insertions, suppressions, replacements and re-orderings. The proposed technique has been implemented as a ProM 6 plug-in.

We are planning to extend this work by considering silent transitions, which are necessary for the analysis of OR splits. The variety of deviations considered in this work poses the basis for assisting privacy auditors in the investigation of privacy violations. Future work includes the development of a user-friendly auditing tool which allows the visualization of violations and automated generation of privacy assessments.

References

1. Petković, M., Prandi, D., Zannone, N.: Purpose control: Did you process the data for the intended purpose? In: Proceedings of 8th VLDB Workshop on Secure Data Management. LNCS 6933, Springer (2011) 145–168
2. Banescu, S., Zannone, N.: Measuring privacy compliance with process specifications. In: Proceedings of Int. Workshop on Security Measurements and Metrics, IEEE (2011) 41–50
3. Rozinat, A., van der Aalst, W.: Conformance checking of processes based on monitoring real behavior. Information Systems **33**(1) (2008) 64–95
4. Adriansyah, A., van Dongen, B.F., van der Aalst, W.M.P.: Conformance Checking Using Cost-Based Fitness Analysis. In: Proc. of EDOC'11, IEEE (2011) 55–64
5. Lakos, C.: Composing abstractions of coloured Petri nets. In: Proceedings of International Conference on Application and Theory of Petri Nets. LNCS 1825, Springer (2000) 323–342