Privacy Preserving Business Process Fusion

Roberto Guanciale(✉) and Dilian Gurov

KTH Royal Institute of Technology, Stockholm, Sweden
{robertog,dilian}@csc.kth.se

Abstract. Virtual enterprises (VEs) are temporary and loosely coupled alliances of businesses that join their skills to catch new business opportunities. However, the dependencies among the activities of a prospective VE cross the boundaries of the VE constituents. It is therefore crucial to allow the VE constituents to discover their local views of the interorganizational workflow, enabling each company to re-shape, optimize and analyze the possible local flows that are consistent with the processes of the other VE constituents. We refer to this problem as *VE process fusion*. Even if it has been widely investigated, no previous work addresses VE process fusion in the presence of privacy constraints. In this paper we demonstrate how private intersection of regular languages can be used as the main building block to implement the privacy preserving fusion of business processes modeled by means of bounded Petri nets.

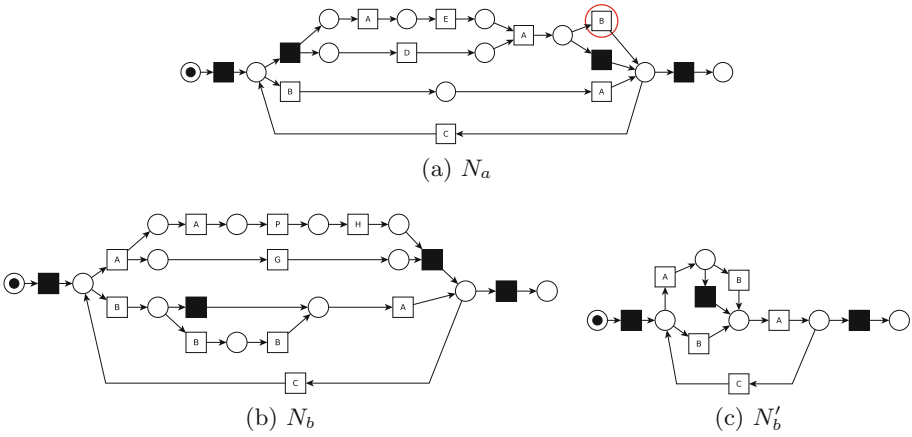
Keywords: Business process fusion · Petri nets · Privacy · SMC

1 Introduction

In the world of business, several potentially competitive enterprises can share their knowledge and skills to form a temporary alliance, usually called a virtual enterprise (VE), in order to catch new business opportunities. Virtual enterprises can be part of long-term strategic alliances or short-term collaborations. To effectively manage a virtual enterprise, receiving well-founded support from business process engineering techniques is critical. In particular, it is necessary to establish the cross-organizational business process, that is, to identify for each participant what can or is to be performed locally. In other words, one needs to compute the contributing subset of the existing local business process that is consistent with the processes of the other VE constituents. We refer to this problem as *VE process fusion*.

To illustrate VE process fusion, consider the following running example. Let a and b be two enterprises, with business processes as shown in Fig. 1a and 1b, respectively, modeled as labeled Petri nets. The two processes contain: (i) internal tasks and events of the enterprises (the boxes labelled E, D, G, H and P, standing for tasks such as the packaging of goods, the receipt of a payment and the like), (ii) shared events and interactions between the two enterprises (A, B and C, representing tasks such as the exchange of electronic documents or the departure of a carrier from the harbor), (iii) silent events (black boxes, usually

F. Fournier and J. Mendling (Eds.): BPM 2014 Workshops, LNBIP 202, pp. 96–101, 2015.
DOI: 10.1007/978-3-319-15895-2_9



(a) N_a (b) N_b (c) N'_b

Fig. 1. Business processes modeled with Petri nets

used to simplify net structure). Intuitively, when the enterprise a is fused with enterprise b, its business process must be updated so as to satisfy the partner's constraints. For instance, an analysis of the fusion suggested above will reveal that the encircled activity B in Fig. 1a can not be executed any more after the fusion.

One of the main obstacles to VE process fusion is the perceived threat to the participants' autonomy. In particular, the participants can be reluctant to expose their internal processes, since this knowledge can be analyzed by the other participants to reveal sensitive information such as efficiency secrets or weaknesses in responding to a market demand. Moreover, the value of confidentiality of business processes is widely recognized, and many enterprises have started to use the patent mechanism to protect the investment required to optimize their workflows.

In this work we consider two mutually distrustful parties, each following a local business process, that wish to compute their local view of the VE process, assuming no trusted third party is available. The VE process is modeled as the synchronous composition of the participant work-flows, and each participant's local view is represented by a process that is trace equivalent to the VE process up to transitions that are not observable by the participant itself. The two parties are reluctant to reveal any information about their own business process that is not strictly deducible from the local view of the other party. For example, regardless whether enterprise b owns the business process N_b or N'_b from Fig. 1, the sub-process of N_a that is consistent with the observable partner's constraints is one and the same (Fig. 2); therefore, the mechanism used to implement process fusion should not allow party a to distinguish between N_b and N'_b or any other partner process that gives rise to the same local view of a.

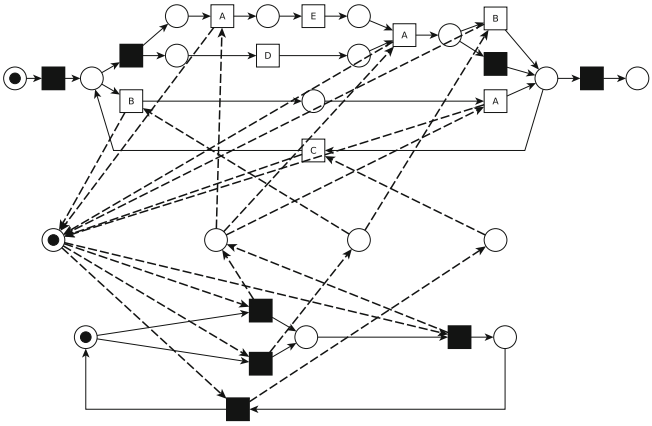


Fig. 2. The local view of a after its fusion with b

To satisfy these security constraints, our work is built on top of our previous results on private regular language intersection and process fusion in a secure multiparty computation setting (SMC) [4].

Here, we demonstrate how these results can be extended to deal with business processes that are formally modeled with bounded Petri nets. Furthermore, we provide a prototype implementation of our proposed technique, developed as a plug-in of ProM [8], a well known business process analysis platform.

2 Private Fusion of Virtual Enterprise Business Processes

We employ bounded labeled Petri nets to formally represent business processes. There is a general agreement (see e.g. [7]) that well-formed business processes correspond to bounded Petri nets (or more specifically, sound workflow nets), and several proposals (e.g. [3]) demonstrate techniques to convert high-level models (such as BPMN) to Petri nets.

Assume two enterprises a and b, with their own business processes, that cooperate to build a VE. For each of the two enterprises we are given a local alphabet, Σ_a respectively Σ_b. The symbols of the alphabets can represent various types of actions or events: (i) an internal task of the enterprise (e.g. packaging of goods), (ii) an interaction between the two enterprises (e.g. exchange of electronic documents), (iii) an event observed by one of the enterprises only (e.g. the receipt of a payment), or (iv) an event observed by both enterprises (e.g. that a carrier has left the harbor). Each enterprise also owns a local business process, representing all possible licit executions, that is given as a bounded labelled Petri net, N_a respectively N_b, defined over the corresponding local alphabet.

The problem of *VE process fusion* can be defined as computing the mapping:

$$N_i \mapsto N_i' \qquad (i \in \{a, b\})$$

Algorithm 1. $Protocol_i(N_a, N_b)$

1. $N_i^p := \mathbf{proj}_{\Sigma_a \cap \Sigma_b}(N_i)$ // project the private net on the common alphabet
2. $A_i^p := NFA(N_i^p)$ // obtain the equivalent nondeterministic automaton // by computing the reachability graph of the net
3. $A_i^d := SC(A_i^p)$ // determinize the automaton
4. send A_i^d to the secure multiparty protocol of [4]
5. receive $A := SMC_\times(A_a^d, A_b^d)$
6. $N := Reg(A)$ // synthesize the corresponding Petri net
7. return $N_i \times N$ // apply the external constraints to the initial net

where $N_i' \sim \mathbf{proj}_{\Sigma_i}(N_a \times N_b)$. and $\sim$ is trace equivalence. Here, the global VE business process is represented by the synchronous composition $\times$ and $\mathbf{proj}_{\Sigma_i}$ means making silent every transition that has label not in Σ_i.

Here we are interested in preserving the participants' privacy. In particular, we wish the two participants to obtain N_a' and N_b', respectively, without being able to learn about the other enterprise's processes more than what can be deduced from the own process (i.e., the *private input*) and the obtained result (i.e., the *private output*). Apart from the processes, we also consider private the alphabet differences; that is, we consider public just the common alphabet $\Sigma_a \cap \Sigma_b$ (i.e., the events of type (ii) and (iv)). Moreover, we assume that no trusted third party is available to serve as an intermediary.

To compute the VE process fusion without compromising the participants' privacy we take benefit from our previous result on private intersection of regular languages.

Algorithm 1 gives the protocol executed by the participant $i \in \{a, b\}$.

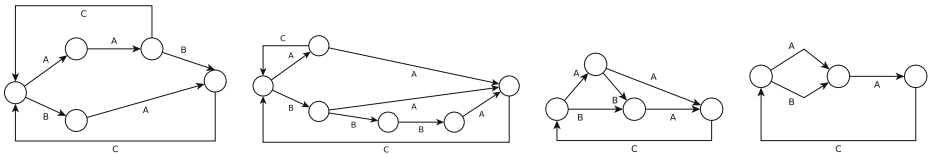
Our protocol is built on two main ideas. Since the input Petri nets are bound, their reachability graphs are finite and can be used to compute the DFAs representing the net languages (steps 1, 2 and 3). Moreover, as proved by Theorem 2, disclosing the intermediate language (step 5) does not leak any information that can not be directly deduced from the private output.

The following result establishes the *correctness* of the protocol, in the sense that the resulting network correctly represents the executions obtained by the synchronous product of the two business process after hiding all internal transitions of the other participant.

Theorem 1. $Protocol_i(N_a, N_b) \sim \mathbf{proj}_{\Sigma_i}(N_a \times N_b)$

The next result shows that the protocol preserves the participants' *privacy*, namely that the two participants are not able to learn about the other enterprise's processes more than what can be deduced from the own processes and the private output.

Theorem 2. *Let N_a, N_b and N_b' be three labeled Petri nets defined over the alphabets Σ_a, Σ_b and Σ_b' respectively. If $\mathbf{proj}_{\Sigma_a}(N_a \times N_b) \sim \mathbf{proj}_{\Sigma_a}(N_a \times N_b')$ and $\Sigma_a \cap \Sigma_b = \Sigma_a \cap \Sigma_b'$ then $Protocol_a(N_a, N_b)$ is indistinguishable from $Protocol_a(N_a, N_b')$.*



(a) A_a^d (b) A_b^d (c) $A_b'^d$ (d) A

Fig. 3. DFAs

Example. We demonstrate the protocol using our running example. Starting from the input Petri nets N_a and N_b of Fig. 1, the two participants compute the DFAs A_a^d and A_b^d of Fig. 3 (steps 1, 2 and 3), hiding all transitions whose labels are not in $\Sigma_a \cap \Sigma_b$, computing the reachability graph, and then determinizing the result. Step 4 requires the execution of the secure multi-party protocol for regular language intersection of [4], which returns the automaton A to both participants. After the termination of the SMC protocol, the participants can proceed independently. Figure 2 depicts the final network obtained by the participant a, which is computed by synthesizing a Petri net from A, and by using the product operator $\times$.

Now, if the participant b owns the Petri net N_b', then the computed intermediate automaton is $A_b'^d$. Notice that the SMC protocol for regular language intersection still yields the automaton A. In fact, A is the minimal automaton of both $A_a^d \times A_b^d$ and $A_a^d \times A_b'^d$. This guarantees that the participant a learns nothing more than the expected result.

3 Prototype Implementation

We developed a prototype implementation by integrating the business process analysis platform ProM [8] with the secure multiparty computation platform Sharemind 2 [1]. Each enterprise hosts an instance of ProM. The enterprise business process can be imported either by using the native ProM support for standard Petri net file formats, or by using existing plug-ins (see e.g. [2,5]) that transform BPMN diagrams to Petri nets. Steps 1, 2, 3, 6 and 7 of the protocol are executed in ProM by using existing plug-ins (e.g. PNAnalisys and TSPetrinet) and a new plug-in encapsulates the functionality of the JAutomata Library.

The SMC protocol for regular language intersection has been implemented using Sharemind. To enable ProM (developed in Java) to interact with the Sharemind client (developed in C++) we encapsulated the latter in a REST web service and we used standard libraries to implement the HTTP protocol and data serialization.

The execution time of the algorithm is dominated by the SMC protocol for regular language intersection. In our experiments on a Linux virtual machine, the protocol requires 185 seconds to handle the Petri nets in Fig. 1.

4 Conclusion

In this paper we present the first privacy preserving protocol that allows the participants to discover their local views of the composition of their workflows, when the latter are modeled with Petri nets. Even if the composition of Petri nets has been widely studied in the contexts of business processes, concurrent systems and Web services, no previous work takes into account privacy from a workflow perspective.

Our ongoing research efforts includes the support for higher abstraction levels. The enterprises can use the existing techniques to transform high level models (e.g. BPMN diagrams) to Petri nets and to enable our protocol. We plan to identify suitable techniques to project back to the high level model the local view of the interorganizational workflow. Finally, the composition of workflows can yield unsound processes, such as interorganizational interactions that manifest deadlocks and livelocks. To support the creation of virtual enterprises, we plan to extend our results to enable suitable analysis techniques (e.g. [6]) without fully revealing the interorganizational workflow.

Acknowledgments. This work has been supported by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 284731 "Usable and Efficient Secure Multiparty Computation (UaESMC)".

References

1. Bogdanov, D., Laur, S., Willemson, J.: Sharemind: a framework for fast privacy-preserving computations. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 192–206. Springer, Heidelberg (2008)
2. Bruni, R., Corradini, A., Ferrari, G., Flagella, T., Guanciale, R., Spagnolo, G.: Applying process analysis to the italian egovernment enterprise architecture. In: Carbone, M., Petit, J.-M. (eds.) WS-FM 2011. LNCS, vol. 7176, pp. 111–127. Springer, Heidelberg (2012)
3. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in bpmn. Inf. Softw. Technol. **50**(12), 1281–1294 (2008)
4. Guanciale, R., Gurov, D., Laud, P.: Private intersection of regular languages. In: PST. IEEE (2014)
5. Kalenkova, A.: ProM BPMN conversions (2014). http://www.promtools.org/prom6/packages/BPMNConversions
6. Van der Aalst, W.: Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries. Inf. Manage. **37**(2), 67–75 (2000)
7. van der Aalst, W.M.: The application of petri nets to workflow management. J. Circuits, Syst. Comput. **8**(01), 21–66 (1998)
8. van Dongen, B.F., de Medeiros, A.K.A., Verbeek, H.M.W.E., Weijters, A.J.M.M.T., van der Aalst, W.M.P.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005)