A Metric for Evaluating the Privacy Level of a Business Process Logic in a Multi-Cloud Deployment

Amina Ahmed Nacer, Claude Godart, Samir Youcef, Abdelkamel Tari

A Metric for Evaluating the Privacy Level of a Business Process Logic in a Multi-Cloud Deployment

Amina Ahmed Nacer[1,2], Claude Godart[1], Samir Youcef[1], Abdelkamel Tari[2],
[1]LORIA/INRIA,
University of Lorraine,
Nancy France
{amina.ahmed-nacer, claude.godart, samir.youcef}@loria.fr

[2]LIMED,
University of Bejaia,
Algeria
abdelkamel.tari@univ-bejaia.dz

***Abstract*—Some companies are willing to execute their business processes (BP) in the cloud for enjoying its benefits. However, they are also reluctant because of the new security risks that using cloud resources introduces. Security risk includes many dimensions, but this work focus on preserving the privacy of the logic of a BP deployed in a multi-cloud context by preventing a coalition of malicious clouds to re-construct important information from this logic. More precisely, the paper presents a BP logic privacy metric directly supporting the evaluation of the risk a company has its logic hacked in a particular multi-cloud configuration**

Index terms— BP modelling, BP deployment in the cloud, Security risk, BP logic privacy

I. Introduction

Cloud computing has emerged as a dominant technology because it avoids upfront infrastructure costs and helps organizations to focus on their core business activities, instead of their system infrastructure.

In this context, some companies are willing to execute, in the same way as other software, their business processes (BP) in the cloud for enjoying its benefits. However, they are also reluctant for their more valuable software because of the new security risks the cloud introduces. Security risk can include many dimensions (privacy, integrity, availability ...). In this work, we focus on a particular aspect of security which is the privacy of a BP logic deployed on a multi-cloud for preventing a coalition of malicious clouds to re-construct important information by combining their BP logic knowledge.

More precisely the paper presents a BP logic privacy metric to evaluate the risk a company has its BP logic hacked in a particular multi-cloud configuration. The global idea is that, on the one hand the more sensitive information is contained in some more sensitive BP fragments, and on the other hand, more these fragments are distant in a cloud configuration, more the configuration is resistant to privacy leaks. The distance between fragments is measured based on the number of clouds on the paths between sensitive fragments weighted with the reputation of these clouds. The metric considers not only attacks from one malicious cloud provider, but from a coalition of cloud providers.

The rest of the paper is organized as follows. The next section explains the basics on which our metric is designed. Section III describes our risk computing metric. Section IV makes an evaluation of the proposed approach and discusses the obtained results. Section V discusses the state of the art and finally section VI concludes and introduces some future work.

II. Motivation, context and approach

A. Motivating example

Figure 1 depicts[1] a selection process of residents for a hospital. The objective of this process is to accept or reject a candidacy. Depending on the resident's record (medical school performed, scores, internships ...) the candidacy can be accepted, refused or, in case of neutral decision, reevaluated. The final decision is taken on the basis of the decision notification combined with aptitude test results.

The hospital is ready to externalize its business process execution to the cloud, however it wants to preserve its strategy for selecting or rejecting a candidacy. In this objective, the company managing the IT services of the hospital decides to obfuscate the BP model by splitting its logic in a collaborations of BP fragments to be assigned to different cloud.

B. BP obfuscation by splitting its logic

Splitting a BP model in a BP fragments collaboration, each BP fragment being assigned to a different cloud , is a basic idea of BP obfuscation [8] as, in consequence, each cloud provider has only a partial view of the BP model.

To support this process, several works [8][5][1] use separation constraints ($separate(t_i, t_j)$) for requiring a sensitive task t_i and its complementing sensitive task t_j to be assigned to different clouds.

[5] proposes an algorithm for automating the generation of separation constraints which, if applied to our motivating example returns the following separation constraints

[1]We use the BPMN (Business Process Modeling Notation (http://www.bpmn.org)) for modeling our BP models. In addition these process models are supposed to be well structured [12]

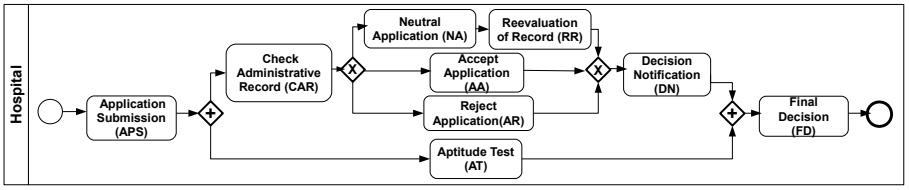

Fig. 1: The *resident selection* Process (orchestration)

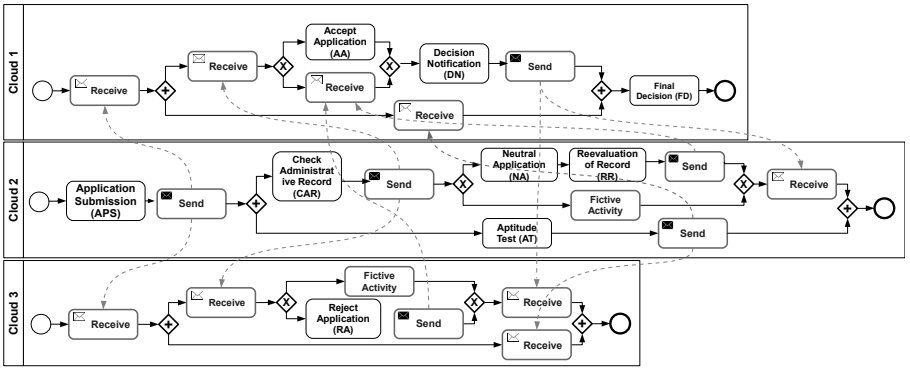

Fig. 2: The *resident selection Process* as a collaboration of BP fragments

- $separate(CAR, DN)$
- $separate(NA, AA)$
- $separate(AA, AR)$
- $separate(AR, RR)$
- $separate(ASP, FD)$
- $separate(NA, AR)$
- $separate(AA, RR)$

Figure 2 describes a collaboration of BP fragments obtained by this splitting, each fragment being assigned to a different cloud, and the fragments being connected with *send* and *receive* messages.

In addition and for increasing the complexity of a BP fragments collaboration, some works [1] add separation constraints with fake fragments. Figure 3 describes another collaboration of the same process while using this principle.

As the BP fragments of the process are assigned to clouds for execution, different assignments leading to different cloud configurations are possible. Therefore, there is a need for comparing these configurations. This is the objective of the metric developed in this paper

C. BP logic privacy violation risk assessment approach

While the splitting of a BP process in BP fragments and other security artifacts (for example fake fragments) provide active support against malicious clouds attacks, they cannot completely ensure the protection of the BP logic and a risk persists. Thus an important issue is to be able to measure this risk. This is the objective of the metric developed in this paper.

Moreover this risk evaluation metric developed is directly related to our BP obfuscation method, and thus enlightens sensitive tasks as first class elements to be preserved from attacks.

Its principle is that for discovering a sensitive information, an attacker has:

- to possess both a sensitive task and the corresponding complementing one (in other terms, to possess the tasks t_i and t_j of a separation constraint $separate(t_i, t_j)$): in the context of a BP fragments collaboration, we formalize this risk as the ability for a cloud executing a sensitive task to discover a path between such a sensitive task and its complementing one, following forward the *send* operation(s) of its fragment (the BP fragment it executes), and/or backward the *receive* operation(s) of this fragment.
- to collude with clouds on the paths between sensitive complementing tasks

More precisely, the metric used for measuring logic privacy is related to number of clouds on the paths between complementing sensitive tasks weighted with the reputation level of these clouds.

III. Metric for BP logic privacy violation risk

In this section we explain how we compute a risk value for a whole collaboration of BP fragments as the sum of the risk for all sensitive tasks to be protected. The risk value for each sensitive task is computed as the sum of the risk for all paths between this task and its complementing one. Such a risk is computed using the general risk formula [3] introduced just below.

A. General definition of a risk

In general, a risk is defined by the likelihood of an incident scenario mapped onto estimated negative impacts.

In the IT context, [3] refine this definition as the likelihood a *threat*, accidentally or intentionally, exploits one or more *vulnerabilities* (flaws or weaknesses in system security procedures, in design, implementation, or internal controls) of the IT environment with a negative *impact* (destruction, alteration, theft, etc.).

Thus the security risk assessment on an artifact *a* consists usually in evaluating the following formula:

$$Risk(a) = V(a) \times T(a) \times I(a) \quad (1)$$

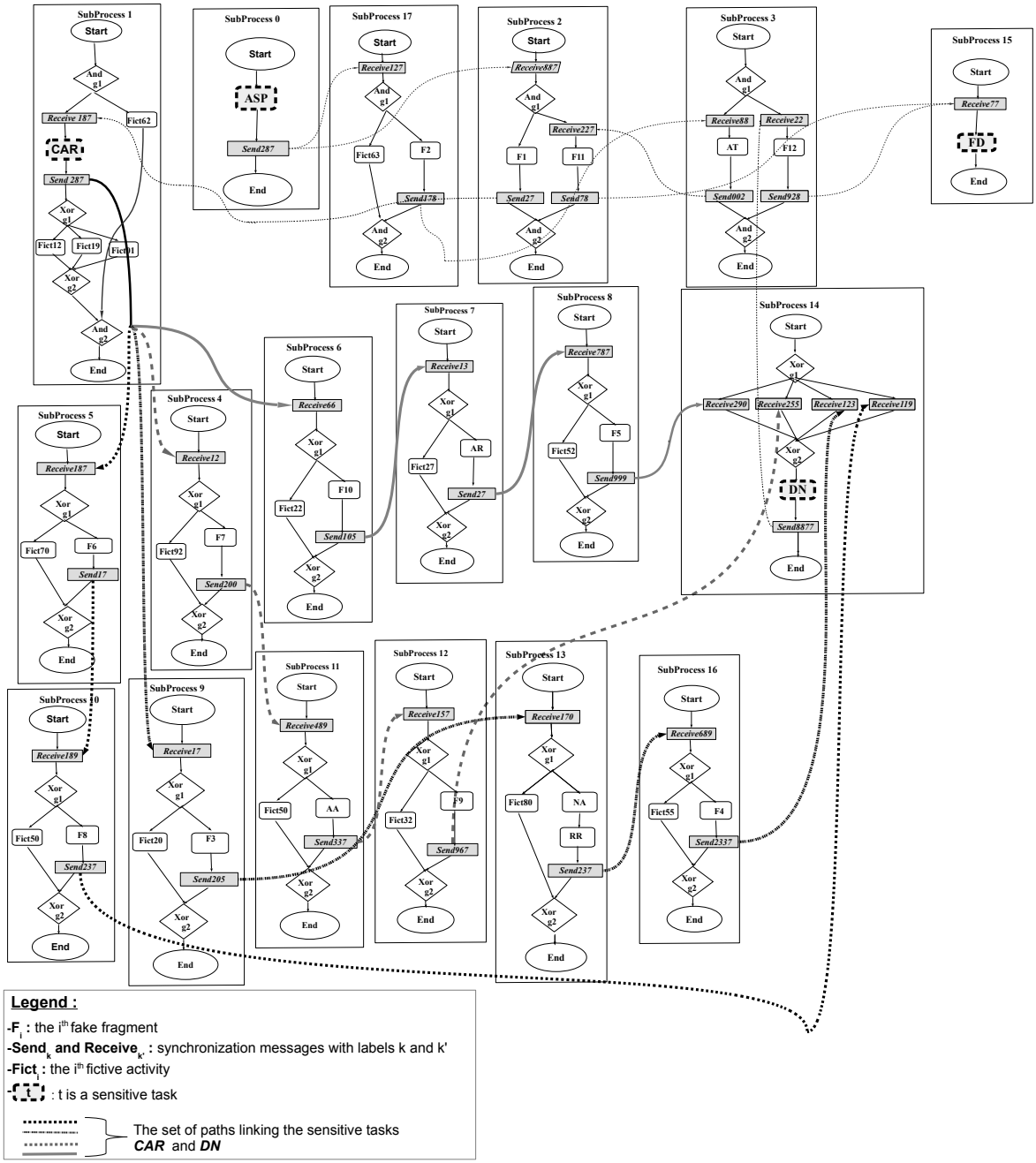

Fig. 3: The BP collaboration corresponding to The *Resident Selection process with fake fragment* deployment

Where

- V: the vulnerability of artifact a
- T: the threat on a
- I: the impact on a

B. Risk level of a collaboration

The risk level to which is exposed the whole process (the collaboration of its process fragments) represents the sum of the risks to which are exposed all the sensitive tasks it includes, and is computed as follows

$$Risk(collaboration) = \sum_{i=1}^{nb} Risk(t_i) \qquad (2)$$

where

- nb: the number of sensitive tasks
- $Risk(t_i)$: the risk level of the sensitive task t_i

C. Global risk level of a sensitive task (considering all paths)

The risk of a task t_i is defined as the sum of the risks following all paths to its complementing task $t_{i'}$.

A path $p_j(t_i)$ between a sensitive task t_i and its complementing sensitive task $t_{i'}$, noted $p_j(t_i)$ is the set of clouds executing the tasks in the j_{th} logical path between t_i and $t_{i'}$ in the BP logic.

$$Risk(t_i) = \sum_{j=1}^{p} Risk_j(t_i) \quad (3)$$

where

- p: the number of paths from t_i to its complementing task $t_{i'}$
- $Risk_j(t_i)$: risk for the task t_i when selecting the path $p_j(t_i)$

D. Risk for a sensitive task on a particular path p_j

Accordingly to definition 1, the risk for a task t_i on path $p_j(t_i)$ is defined as:

$$Risk_j(t_i) = Vulnerability(t_i) \times Threat_j(t_i) \times Impact_j(t_i) \quad (4)$$

where :

- $Vulnerability(t_i)$: the vulnerability of t_i depends on the cloud executing the task, or how it is ready to collude with other clouds. This value is common to all paths.
- $Threat_j(t_i)$: the threat on a task t_i or how the clouds on the path $p_j(t_i)$ are ready to collude
- $Impact_j(t_i)$: the impact of a collusion of the clouds on the path $p_j(t_i)$

1) Vulnerability of a task: As introduced above, the *vulnerability* of a BP fragment comes from the cloud executing it, or how it is ready to collude with other clouds. In practice, we directly relate the vulnerability of a task to the level of reputation of the cloud deploying it, or the more a cloud has a good reputation, the less the task will be exposed to threats. Formally, the *vulnerability* of a task t_i is a normalized value belonging to the interval $[0; 1]$ defined as follows:

$$Vulnerability(t_i) = 1 - rep(c) \quad (5)$$

where:

1) t_i: the considered sensitive task
2) c: the cloud deploying (t_i)
3) rep(c) : the reputation of the cloud which execute (t_i)

Measuring the reputation of a cloud is out of the scope of this paper. Many approaches has been proposed for dynamically measuring such a reputation [9]. Our only hypothesis is that reputation can be normalized in [0,1], 0 the less reputed, 1 the best.

2) Threat of a task t_i relatively to path $p_j(t_i)$: We consider that the threat on a sensitive task relatively to a path is related to:

- the sensitivity[2]: the more a task is sensitive, and the more malicious clouds will be motivated to attack it,
- the reputation of clouds: the better reputation a cloud has, and the less it is ready to collude,
- the number of clouds: the more clouds in the path and the more difficult it is for them to collude; we consider that this difficulty evolve exponentially.

[2]the notion of sensitivity of a task is introduced in [10]: it is a designer defined value assigned to a task taken in [0, 1] : 0 for the less sensitive, 1 fore the more sensitive

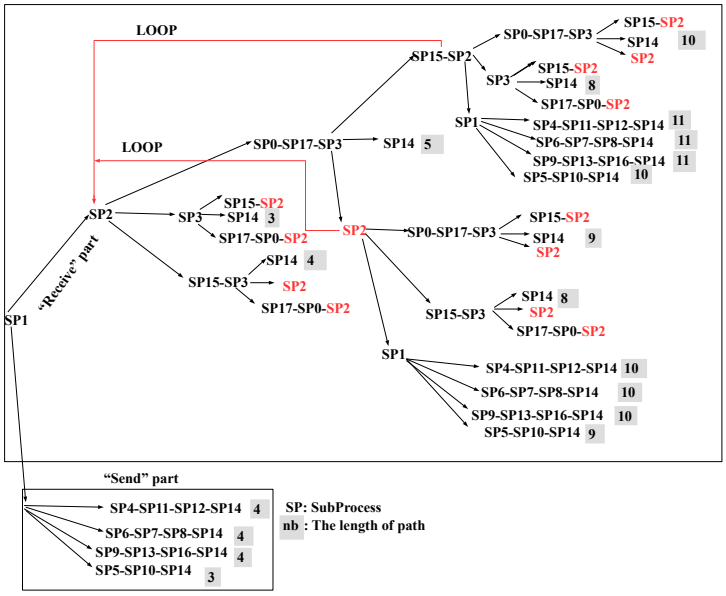

Fig. 4: Possible paths between CAR and DN

This is formalized as follows:

$$Threat_j(t_i) = s_t * \left(\frac{1}{e^{card(p_j(t_i)) \sum_{j=1}^{card(p_j(t_i))} rep(C_j)}}\right) \quad (6)$$

where

- s_t: the sensitivity of the task t ($\in [0, 1]$)
- card($p_j(t_i)$) : the number of clouds in $p_j(t_i)$
- rep(c_j): the reputation of cloud j

3) Impact of a threat on a task: As previously suggested, the impact of threats can be opportunely correlated to the number of clouds implied in an attack, i.e. the more clouds in the coalition, the more important is the impact as each cloud belonging to the malicious coalition can take advantage of the flaw. Thus, we choose to measure the impact of a threat as the rate (percentage) of clouds implied in the attack. Equation 7 gives the impact of a threat for a task t_i relatively to a path p_j.

$$Impact_j(t_i) = \frac{card(p_j(t_i))}{cardSys} \quad (7)$$

where

1) card($p_j(t_i)$) : the number of clouds in $p_j(t_i)$
2) cardSys : the cardinality of the system, i.e. the number of clouds involved in the BP collaboration.

IV. Application and Experimentations

To illustrate the utility and the applicability of our metric, we apply it in a first time to our motivating example (section IV-A) to compute the different risk levels to which is exposed a sensitive task on different paths. In a second time, we simulate its behavior in different execution settings (section IV-B).

Cloud	Reputation	Cloud	Reputation
$SP0$	0,8	$SP9$	0,6
$SP1$	0,5	$SP10$	0,3
$SP2$	0,4	$SP11$	0,7
$SP3$	0,6	$SP12$	0,2
$SP4$	0,3	$SP13$	0,7
$SP5$	0,2	$SP14$	0,8
$SP6$	0,1	$SP15$	0,3
$SP7$	0,4	$SP16$	0,7
$SP8$	0,2	$SP17$	0,3

TABLE I: Cloud's reputation

A. Example: the case of the sensitive task CAR and its complementing one DN

To illustrate the risk model developed above, we come back to our motivating example and we focus on the risk that malicious clouds can link the sensitive task CAR to its complementing one DN.

Figure 4 depicts all existing paths linking the tasks CAR and DN. The top part of the figure depicts the paths which can be discovered using the *receive* operation and the bottom part the ones which can be discovered using the *send* operation.To each path is associated its length i.e. the number of clouds it includes. It must be noted that the task CAR is deployed on subprocess 1 ($SP1$) while DN is deployed on subprocess 14 ($SP14$).

a) ***Step one****:* In the first step, we determine the different paths linking tasks CAR and DN. For example, starting from $SP1$ and following the $send_2 87$ operation, we find the path of length 4 composed of subprocesses ($SP1-SP5-SP10-SP14$) (highlighted in bold in figure 3 and shown in figure 4, bottom part) leading from CAR to DN.

b) ***Step two****:* After selecting the set of paths, we calculate the risk to which is exposed the task CAR *on each path* by computing the vulnerability, the threat and the impact.

For example, applying our equations 5, 6 and 7 to the path ($SP1 - SP5 - SP10 - SP14$) returns respectively the following results for CAR about its vulnerability, its threat and their impact, while using the clouds' reputation depicted on table I :

$Vulnerability(CAR) = (1 - 0,5) = 0,5$

$Threat(CAR) = 0.7 \times (\frac{1}{e^{4\sum_{j=1}^{4} Rep(C_j)}})$

Where $C = \{SP1 - SP5 - SP10 - SP14\}$.

We obtain then

$Threat(CAR) = 0,7 \times (\frac{1}{e^{4\times 1,8}}) = 0.00052$

The impact of CAR is computed as

$impact(CAR) = (\frac{4}{18}) = 0,22$

Then, we compute the risk to which is exposed the task using equation **1**. The risk of CAR on path $SP1 - SP5 - SP10 - SP14$ is computed as

$Risk(CAR) = 0,5 \times 0.00052 \times 0,22 = 57.10^{-}6$.

We compute the risk to which is exposed CAR on each path of figure 4 using the same method.

After computing the risk to which is exposed CAR on each path, we compute the global risk to which it is exposed using equation 3.

c) ***Third step****:* After applying the same method as applied to the task CAR for each sensitive task of the process for global risk computation, we compute the risk of the whole collaboration using equation **2**.

$Risk(Collaboration) = \sum_{i=1}^{n} Risk(t_i), n \in SetOfSensitiveTasks$

B. Experimentation

To observe the behavior of our risk metric, we have evaluated the risk of deploying a BP in the cloud with regards to three deployment approaches:

1) an aleatory distribution of tasks,
2) a distribution respecting the principle of assigning complementing sensitive tasks in different clouds (as introduced in [5] and summarized in section II-B,
3) a distribution as the previous one but extended with fake fragments as introduced in [1].

1) Experimental settings: All algorithms were coded in java language. All experiments were performed on an intel (R) Core(TM) i3- 2310M 2.10 GHz running Windows Seven.

We used different BP with a variable number of tasks. We generate until 1000 instances of collaborations for each of these BP and for each of the three ways of distribution.

As we were not able to find real data information (providers do not reveal the level of reputation of their cloud servers), we generate randomly a set of cloud reputations (table I is an example).

2) Experimental Results: Figures 5 depicts our experimental results in terms of risk level [3] (figure a) and number of clouds (figure b), depending on the number of tasks in processes. As intended, they confirm:

- that configurations separating complementing sensitive fragments are less risky that aleatory defined configurations,
- the fake mechanism coupled to the previous algorithm minimizes even more the risk until obtaining very small values,
- the number of clouds is not correlated to the number of activities, but to the number of sensitive tasks (this explain why the process with 10 activities in figure 5-b needs more clouds than the process with 14 activities.

These results confirm our assumptions i.e. separating sensitive complementing fragments on different clouds minimizes the likelihood and the risk of a coalition. Moreover, the mechanism of fake fragments increases the length and the number of paths separating sensitive fragments, making longer an eventual collusion and therefore less probable.

[3]The risk level is normalized to scale the range in $[0, 1]$ using the following widespread formula $x' = \frac{x-min}{max-min}$ (x'and x are respectively the normalized and original values while min and max are respectively the minimum and the maximum risk values computed)

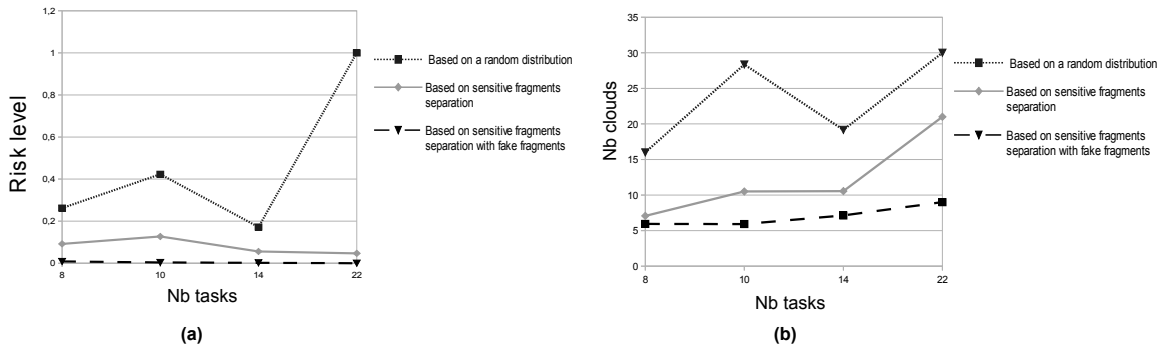

Fig. 5: Risk level evolution with regards to deployment strategies.

V. State of the art

The work described in this paper is related to Business Process Outsourcing (BPO) in a multi-cloud environment. Even if many researches have already be done in the context of BPO [15][2], only few of them are concerned by the cloud computing context.

Directly related to our field, Rekik and al.[13] proposed a framework named *Business Process Outsourcing (BPO2C)* covering the outsourcing process life-cycle to identify the implied business process in the outsourcing decision. However, this work did not consider the risk of know-how disclosure by neither a single cloud provider nor a collusion of several clouds.

The proposed approach of this paper comes in the continuation of our previous works. In [6] we yet defined a security risk metric but based on data given by cloud providers, while in this current work the risk value is based on our proper obfuscation model i.e. in other words based on data from the cloud client himself.

Several other researches are directly concerned with the security risk of a BP but in traditional information systems settings[14] [7][11][4] and as such do not consider the deployment of a BP in a multi-cloud context, and less the conspiracy of resource providers.

VI. Conclusion and future work

This paper has introduced a metric for risk evaluation during a BP deployment in a multi-cloud context. Its originality is to be based on client-side data and not data given by cloud providers, what is a better guarantee for BP owners.

Such a metric is important for cloud clients who need to compare different deployment solutions, as at the one hand security cannot be completely ensured, and on the other hand the security is a parameter to balance with other quality of services (QoS) parameters.

Our future work concerns the development of a larger security risk metric based on our previous work in [6] and this current work, while confronting the security dimension to other QoS parameters.

References

[1] A. Ahmed Nacer, E. Goettelmann, S. Youcef, A. Tari, and C. Godart. Obfuscating a business process by splitting its logic with fake fragments for securing a multi-coud deployment. In *the IEEE international Congress on Services (SERVICES)*, page 8, 2016.

[2] U. Arnold. New dimensions of outsourcing: a combination of transaction cost economics and the core competencies concept. *European Journal of Purchasing & Supply Management*, 6(1):23–29, 2000.

[3] S. Australia. Handbook: Risk management guidelines, companion to as/nzs 4360: 2004. *Standards Australia Internal Ltd, Sydney*, 2004.

[4] R. Conforti, M. de Leoni, M. La Rosa, W. M. van der Aalst, and A. H. ter Hofstede. A recommendation system for predicting risks across multiple business process instances. *Decision Support Systems*, 69:1–19, 2015.

[5] E. Goettelmann, A. Ahmed-Nacer, S. Youcef, and C. Godart. Paving the way towards semi-automatic design-time business process model obfuscation. In *Web Services (ICWS), 2015 IEEE International Conference on*, pages 559–566, 2015.

[6] E. Goettelmann, K. Dahman, B. Gateau, E. Dubois, and C. Godart. A security risk assessment model for business process deployment in the cloud. In *Services Computing (SCC), 2014 IEEE International Conference on*, pages 307–314, 2014.

[7] A. Jallow, B. Majeed, K. Vergidis, A. Tiwari, and R. Roy. Operational risk analysis in business processes. *BT Technology Journal*, 25(1):168–177, 2007.

[8] M. Jensen, J. Schwenk, J.-M. Bohli, N. Gruschka, and L. L. Iacono. Security prospects through cloud computing by adopting multiple clouds. In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pages 565–572. IEEE, 2011.

[9] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision support systems*, 43(2):618–644, 2007.

[10] A. A. Nacer, E. Goettelmann, S. Youcef, A. Tari, and C. Godart. Business process design by reusing business process fragments from the cloud. In *Service-Oriented Computing and Applications (SOCA), 2015 IEEE 8th International Conference on*, pages 193–200. IEEE, 2015.

[11] A. Pika, W. van der Aalst, M. Wynn, C. Fidge, and A. ter Hofstede. Evaluating and predicting overall process risk using event logs. *Information Sciences*, 352:98–120, 2016.

[12] A. Polyvyanyy, L. García-Bañuelos, and M. Dumas. Structuring acyclic process models. *Information Systems*, 37(6):518 – 538, 2012. BPM 2010.

[13] M. Rekik, K. Boukadi, and H. Ben-Abdallah. A comprehensive framework for business process outsourcing to the cloud. In *Services Computing (SCC), 2016 IEEE International Conference on*, pages 179–186. IEEE, 2016.

[14] B. Weber, M. Reichert, W. Wild, and S. Rinderle. Balancing flexibility and security in adaptive process management systems. *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, pages 59–76, 2005.

[15] D.-H. Yang, S. Kim, C. Nam, and J.-W. Min. Developing a decision model for business process outsourcing. *Computers & Operations Research*, 34(12):3769–3778, 2007.