A Decision-Making Approach for Improving Organizations' Cloud Forensic Readiness

Stavros Simou[1(✉)], Ioannis Troumpis[1], Christos Kalloniatis[1], Dimitris Kavroudakis[2], and Stefanos Gritzalis[3]

[1] Privacy Engineering and Social Informatics (PrivaSI) Laboratory,
Department of Cultural Technology and Communication,
University of the Aegean, University Hill, 81100 Mytilene, Greece
{ssimou,chkallon}@aegean.gr, envm6l5000l@env.aegean.gr

[2] Department of Geography, University of the Aegean,
University Hill, 81100 Mytilene, Greece
dimitrisk@aegean.gr

[3] Information and Communication Systems Security Laboratory,
Department of Information and Communications Systems Engineering,
University of the Aegean, 83200 Samos, Greece
sgritz@aegean.gr

Abstract. Cloud forensic investigation involves a number of different people and requires a lot of effort to resolve an incident. In order for an organization to have better chances to succeed in the investigation, it needs to be well-prepared. Hence, the organization needs to develop cloud forensic readiness. This paper introduces a decision-making approach to calculate the forensic readiness and the compliance level of an organization, and in parallel to classify the unimplemented tasks in a cloud service. The specific approach is based on a decision-based algorithm, the organization's forensic compliance and a number of decision-making criteria.

Keywords: Cloud forensic readiness · Decision-based algorithm
Cloud forensic constraints · Cloud forensics

1 Introduction

Cloud computing has managed to gain popularity among consumers, due to its ability to provide on-demand services with minimal cost from anywhere in the world. The extensive use of cloud computing has attracted many users aiming to gain control on private information and commit malicious actions [1]. Incidents occurring in cloud environments such as stealing confidential information, trafficking illegal material, draining system's resources, etc. are even more demanding, in terms of complexity and expertise needed, and difficult to resolve in comparison to traditional digital environments. Cloud forensic investigation has to deal with evidence that reside everywhere in the world in distributed and virtualized environments [2]. This makes the investigation complex since different jurisdictions are involved in multi-tenancy environments and ambiguous transparency [3–6].

S. Furnell et al. (Eds.): TrustBus 2018, LNCS 11033, pp. 150–164, 2018.
https://doi.org/10.1007/978-3-319-98385-1_11

The past years extensive research has been conducted regarding digital forensic investigation (DFI) in cloud environments. Several researchers have proposed various frameworks, models and solutions. A detailed review concerning the cloud forensic investigation area can be found in Simou et al. [7]. The main goal of these propositions is to make cloud a safer and secure place, where an incident can be solved in a forensically sound manner and a cloud service can achieve and maintain forensicability. In this paper, the term forensicability describes a service that can be forensic-enabled. To succeed in this difficult part, it is necessary to understand the investigation process in relation to the cloud services provided by the Cloud Service Providers (CSPs).

The increasing number of cyber-crime incidents in cloud environments has raised concern for data privacy and security issues. According to a RightScale 2018 report [8], security is still the top challenge for the 77% of the respondents, while 29% see it as a significant challenge. The same report found that 96% of the respondents (small, medium business and enterprises) are using cloud services. Due to data security issues and cloud incidents, organizations need to evaluate their forensic readiness. Forensic readiness describes how well-prepared an organization is to perform a forensic investigation, in case of an incident.

Tan [9] is one of the first researchers who defined the term forensic readiness as, the organization's achievement to maximize the ability of acquiring digital evidence, while minimizing the cost of any digital forensic investigation. Rowlingson et al. [10] moves a step forward by introducing a number of steps that an organization should take, to implement a forensic readiness. ISO 27043:2015 [11] defines readiness as the process that deals with pre-incident investigation, i.e. of being prepared for a digital investigation before an incident has occurred. There are various definitions for cloud forensic readiness by numerous researchers [12–14]. For this paper purposes, cloud forensic readiness is defined as: "*The organization's preparations to minimize the impact of an incident in a cloud forensic investigation, while identifying and acquiring the maximum amount of digital evidence.*"

Simou et al. [15], introduced a framework regarding cloud forensic-enabled services, which identifies an organization's forensic readiness in relation to cloud services and calculates if a service is forensic-enabled. However, it is expected that several organizations will not be able to achieve full forensic readiness. In that case, a metric should be introduced to illustrate how close an organization is to become fully forensic ready. Furthermore, another metric is needed to help an organization decide which steps it needs to take, to maximize its forensic readiness, given its resources.

The main purpose of this paper is to provide a reliable methodology to calculate the cloud forensic readiness of an organization and to quantify the priority that the unimplemented tasks should have. The measure of these variables concerns the tasks that are not implemented to be considered as forensic-enabled. The proposed methodology can either be used by organizations with huge impact in the cloud business market, such as Google and Amazon, or smaller in-house cloud service providers.

Our contribution is the introduction of an algorithm that calculates the cloud forensic readiness of an organization by identifying the importance of all the tasks in a cloud service, and classifying the unimplemented tasks in order to prioritize them accordingly. The classification and prioritization is used to decide the order in which the tasks should be implemented.

This paper is organized as follows. Section 2 presents a cloud investigation process as derived from [16] and matches a set of forensic constraints (high-level requirements) to every step of the cloud investigation process in order to understand which constraints should be included in the implementation of a given cloud-service for the latter to be considered as forensic enable. In Sect. 3, a novel approach is described using a decision-making algorithm that calculates the cloud forensic readiness of a service considering a set of predefined decision-making criteria. Section 4, validates the approach, using a real case study. Finally, Sect. 5 provides conclusions and raises discussions for future work.

2 Cloud Forensic-Enabled Framework and Investigation Process

2.1 Cloud Investigation Process

To provide proper and efficient investigation in cloud environments, organizations need to be prepared and act in accordance with forensic standards and principals. One of the most important task an organization needs to be informed is about the level of its cloud forensic readiness. Once its forensic readiness is identified, the organization is aware about its systems' vulnerabilities against the forensic process that would be triggered by a security incident and can act accordingly. Malicious actors use cloud services and infrastructures to explore systems' vulnerabilities and gain control over consumers' data. When an incident occurs, a team is formed to investigate the specific incident. The investigation team could be employees assigned by the organization, Law Enforcement Agents (LEA), or external contractors. All members of the team are responsible to resolve the incident in a forensically sound manner, using guidelines, methods and procedures that meet specific forensic investigation standards.

Simou et al. [16] proposed a generic cloud forensic investigation process, based on a comparison framework and the literature review conducted in [7]. The process consisted of five sequential steps: Incident Confirmation, Incident Identification, Collection-Acquisition, Examination-Analysis, and Presentation, together with three more parallel activities/steps that are running concurrently: Preservation of evidence, Documentation, and Training and Planning. Figure 1 illustrates the forensic investigation process in cloud environments.

Based on the investigation process presented above, we adapt our previously published framework [15], regarding cloud forensic-enabled services and present the role of the forensic constraints, identified in that work, in relation to the investigation process. The seven forensic constraints identified in that work are: internal disciplinary procedures, accountability, transparency, legal matters, access rights, isolation, and traceability. For each forensic constraint a feature diagram was introduced for expressing the basic tasks that need to be realized in order for every forensic constraint to be addressed. All tasks in a forensic constraint must be implemented in order for it to meet the forensic standards. Thus, each feature diagram has only a single valid configuration and no alternatives can be introduced when applying it on every cloud service. All the aforementioned cloud forensic constraints should be applied on a cloud service in order to be forensic-enabled.

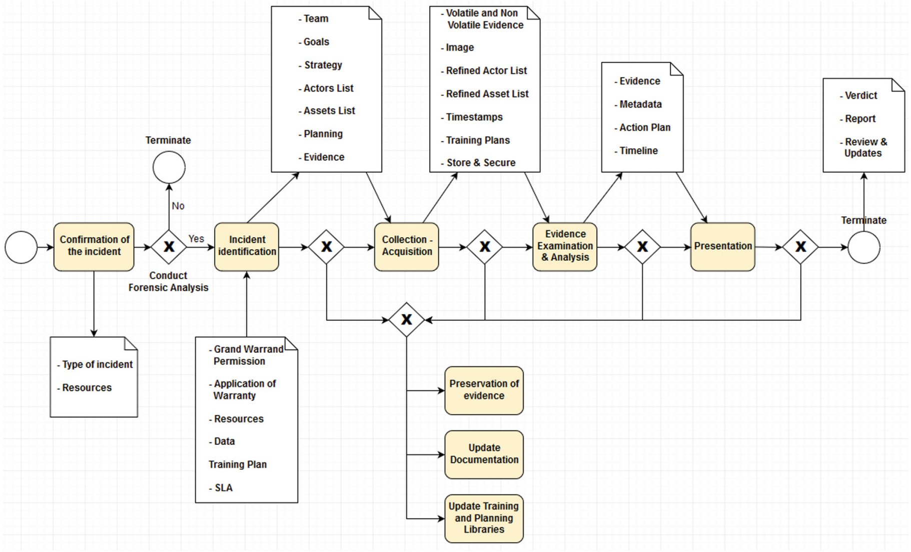

Fig. 1. Process for cloud forensic investigation [16]

2.2 Forensic Constraints in the Investigation Process

In order to identify the seven forensic constraints a detailed study [7] was conducted to identify cloud methodologies, challenges, and solutions and gave us the opportunity to introduce a new category based on the forensic constraints, the forensic requirements. Forensic constraints are requirements related to the forensicability of a service and specify the quality attributes of the service. For each forensic constraint, a feature diagram is created for expressing the basic tasks that need to be realized. Feature modelling assists engineers in modelling the properties of concepts and their interdependencies and organizing them into a coherent model referred to as a feature model [17].

The role of the seven forensic constraints is of vital importance, since the tasks of the constraints fit into specific stages of the investigation process and are used to assist the investigation team. A categorization of the constraints and their tasks in relation to the stages of the cloud forensic investigation process is presented in Table 1. The proposed seven forensic constraints interfere/influence the cloud forensic investigation by contributing to all stages of the investigation process.

Table 1. Forensic constraints' contribution to cloud investigation process

Constraint	Task	Fulfilment
Internal disciplinary procedures	Implement discipline rules	Incident identification – collection and acquisition
	Enable access rights	Preservation of evidence – collection and acquisition
	Enforce legal contracts	Incident identification – collection and acquisition
Accountability	Ensure agreements	Incident identification – evidence examination & analysis – documentation
	Provide assurance	Evidence examination & analysis – documentation
	Monitor actions	Incident identification – evidence examination & analysis – documentation
	Provide attributability	Confirmation of the incident – evidence examination & analysis – documentation
Transparency	Ensure visibility	Evidence examination & analysis – documentation
	Provide procedures and policies of treating data	Evidence examination & analysis – documentation
	Provide notification on policy violation	Evidence examination & analysis – documentation
Legal matters	Define SLAs	Incident identification – presentation
	Ensure jurisdiction	Incident identification – presentation
	Maintain trained personnel	Preservation of evidence – update training & planning libraries – presentation
Access rights	Ensure registration and validation control	Preservation of evidence – evidence examination & analysis
	Enable authentication and authorization control	Preservation of evidence – evidence examination & analysis
	Enable access control	Preservation of evidence – evidence examination & analysis
Isolation	Ensure users do not have access to each other	Collection and acquisition
	Prevent contamination of other users	Collection and acquisition
	Provide confidentiality	Collection and acquisition
Traceability	Monitor users activities	Confirmation of the incident – evidence examination & analysis
	Monitor data logs	Confirmation of the incident – evidence examination & analysis
	Store and secure logs	Evidence examination & analysis
	Link users to data	Evidence examination & analysis

3 Decision Making Approach

All cloud services provided to the users by a CSP should be forensic-enabled for increasing both the security and the trustworthiness of the service. The main purpose of the forensic requirements' contribution to the investigation process is to identify the degree of the cloud forensic readiness of a specific cloud service regarding a forensic investigation. This is necessary for the software analysts of an organisation to understand the forensic readiness and forensic vulnerabilities of their system. In order to calculate that, and in parallel to classify the unimplemented tasks in a cloud service, a decision-making approach is introduced. The specific approach is based on a decision-based algorithm, the organization's forensic compliance and a number of decision-making criteria. In the following section, a detailed description concerning the approach is presented.

3.1 The Decision-Making Algorithm

The first phase of the approach is to identify the importance of all the tasks in a cloud service, classify, and prioritize them accordingly. The algorithm introduces four different steps for calculating specific values that are considered highly important for an organization's forensic readiness. The fifth step of the algorithm concerns the implementation of the selected task and the re-application of the process to identify the next one. The complexity of the algorithm is O(n), and given that the number of tasks is small, should be considered trivial. The algorithm is illustrated as follows.

Step 1. Calculate the Forensic Compliance of each investigation step of the cloud service in question and the Total Compliance Level of the service.

```
For i 1 to Number of Investigation Stages
  For j 1 to Number of Tasks in $Investigation\_Step_i$
    If $Task_j$ is implemented then
      $task\_sum = task\_sum + 1$
    End If
  End For
  $\mathbf{CIS_i} = task\_sum / (\text{Number of Tasks in } Investigation\_Step_i)$
  $TCL = TCL + CIS_i$
End For
$\mathbf{TCL} = TCL + /$ Number of Investigation Steps
```

Step 2. Calculate the effort of each unimplemented task.

```
For i 1 to Number of Unimplemented Tasks
  For j 1 to Number of Variables
    $x_j = 0$
    For k 1 to Number of Stakeholders
      $x_j = x_j + v_k$
    End For
    $e_i = e_i + x_j /$ Number of Stakeholders
  End For
  $\mathbf{e_i} = e_i /$ Number of Variables
End For
```

Step 3. Calculate the necessity of each unimplemented task.

```
For i 1 to Number of Unimplemented Tasks
  For j 1 to Number of Investigation Steps i is in
    $step\_impact = (1 + FT_j) / TFT_j$
    $step\_completion = (1-CIS_j)$
    $step\_score = step\_impact * step\_completion$
  End For
  $\mathbf{n_i} = n_i + step\_score$
End For
```

Step 4. Calculate the priority of each unimplemented task.

```
For i 1 to Number of Unimplemented Tasks
  $\mathbf{p_i} = n_i / e_i$
End For
```

Step 5. Implement the task with the highest p score and recalculate.

Specifically, in the first step, the compliance of each investigation step is calculated as the fraction of the satisfied tasks to the total number of tasks in each step. The total compliance level is calculated as the average value of the compliance level of each investigation step. Afterwards, necessity is calculated as the contribution of the task in question in each investigation step it participates in, inversely weighed by the completion of the investigation step. Then, the score of each effort variable is calculated as the average score the stakeholders give, and the effort is calculated as the mean value of the effort variables' scores. Finally, the priority score for each task is calculated as the fraction of necessity divided over the effort, and the stakeholders should choose to implement the task with the highest priority.

3.2 Forensic Compliance

To calculate the forensic readiness of a cloud service, and specifically to maintain its forensicability the provider should maintain the level of compliance in relation to the investigation process. First, the total number of supporting tasks (TNT) in relation to the investigation steps need to be specified. It is required to quantify the number of forensic tasks that contribute to the fulfilment of every investigation step. Given that a forensic task can participate in more than one investigation stage, the total number of forensic tasks in the investigation process is forty-five (45), as shown in Table 1. These tasks contribute to the investigation steps and Table 2 presents the quantification per investigation step.

Table 2. The supporting tasks in the investigation steps

Investigation step	Total number of supporting tasks
Confirmation of the incident	3
Incident identification	6
Collection and acquisition	6
Evidence examination & analysis	14
Presentation	3
Preservation of evidence	5
Documentation	7
Update training & planning libraries	1

Since the total number of supporting tasks is known, the second priority of an organization is to calculate the number of cloud service's forensic tasks (FT) that are already implemented, by applying the feature diagrams of each forensic constraint to cloud service's activity diagram [15]. In that way, the organization will be aware about the missing tasks of any cloud service and will be able to calculate the compliance of the investigation stage (CIS) and the total compliance level (TCL). Both type of formulas are shown in Eqs. (1) and (2).

$$CIS_i = \frac{FT_i}{TNT_i} \tag{1}$$

$$TCL = \frac{1}{nIS}\sum_{i=1}^{nIS} CIS_i \tag{2}$$

CIS is defined as the fraction of the number of forensic tasks (FT) in a cloud service divided by the total number of supporting tasks (TNT) and represents the completeness of an activity's forensicability. TCL is the sum of all the CISs divided by the number of investigation stages (nIS). TCLs possible values are $0 \leq \text{TCL} \leq 1$, where 0 means that the service does not satisfy any forensic task, and 1 that the service is forensic-enabled. Any value less than 1, means that the service is not forensic-enabled.

To define the level of compliance of a cloud service, even though it is not considered forensic-enabled, can still be very useful, since software developers can get valuable information about the investigation readiness of the services. For example, imagine a cloud service where its total compliance level is 60%. If the organization is willing to raise service's TCL to 100%, this could be succeeded, since the organization's engineers can focus on the desired investigation stages according to the unsatisfied tasks.

However, although calculating the compliance is a useful metric in identifying which tasks must be implemented, it cannot be used as a guide to decide which and in what order the remaining tasks should be implemented. Adding to that, an organization may be willing to increase its forensicability, but due to various factors (cost, complexity etc.), be unable to fully implement all the tasks required. A prioritization of the implementation of the unsatisfied tasks should be introduced, to help stakeholders and IT, decide which tasks should be implemented first. Each task may have different criteria, such as the complexity of the solution, the financial cost, or the participation of the task in a forensic investigation. In the following section an approach is presented in which the priority that should be given to each unimplemented task is quantified, by calculating necessity and effort required to implement it.

3.3 Decision Making Criteria

For the decision-making process we consider a set of indicative criteria that can assist in prioritizing the investigation steps that include unimplemented tasks in order for the stakeholders to consider which tasks should be implemented first based on the necessity and effort required. Of course, the algorithm presented can adopt additional or different variables based on stakeholder needs. In the current section the definition of the variable used are presented along with the respective equations used for their calculation.

Effort is defined as the total work done to achieve something. In our case, effort is the work an organization must produce and maintain in order to implement a solution that satisfies a specific task. A few variables need to be defined in order to understand and quantify the notion of effort. Complexity, cost, and standardization and openness are the variables that are needed. **Complexity** is defined as how hard it is to implement and maintain a specific solution in a cloud environment. This applies both to technical solutions, such as the installation of specific software/hardware, and non-technical, such as legal agreements or employee training. **Cost** is both the initial and continuous monetary cost of the implementation of a specific solution. **Standardization and Openness** is whether there are industry standards regarding the implementation of a task and whether there exist multiple and/or relatively available solutions.

In order to quantify effort, the following method is used. First, each stakeholder assigns a score (an integer between 1 and 10) to each variable, where a high score means more effort required to implement the task. It is not necessary for each stakeholder to provide a score for each variable, if his expertise is not enough. For instance, an organization's Head of IT is not expected to have an informed opinion on legal matters, so it would be pointless to ask him to score the corresponding variables. Afterwards, the average of the scores is calculated as the final score of the variable. Finally, effort is defined as the average value of the above variables. Equations (3) and (4) calculates the **effort**:

$$x = \frac{1}{s}\sum_{j=1}^{s} v_j \tag{3}$$

$$e_i = \frac{1}{n}\sum_{k=1}^{n} x_{ik} \tag{4}$$

where s is the number of stakeholders that scored the particular variable, v is the value that each stakeholder assigns to the variable x, n is the number of effort variables (in our study 3), and x is the final value of each variable. A stakeholders' score is defined as $v \in \{1, 10\}$ and therefore 1 $\leq$ e, x $\leq$ 10. x is the average value of the scores that each stakeholder gave for the specific variable. It is expected that well-informed stakeholders should assign similar scores, and in that case the mean will represent consensus. Similarly, e is the average value of the variables. There is no reason to individually weigh each variables' importance, since it varies based on the organization's capabilities and priorities. So, the stakeholders will assign a numerical value for each variable that reflects their individual goals and needs. For example, a mega-corporation will view complexity and cost as of less importance than a small cloud-based start-up and will assign scores accordingly. Assigning specific weights to each variable will assume that a variable is objectively more or less important than the others, which is false.

Necessity is defined as the state or the fact of being required or necessary. In this paper, it is defined as the organization's need to implement a specific task to make a cloud service forensic-enabled. According to our previous definition, a cloud service is either forensic-enabled or not, but it would be naive to assume that all tasks that are not yet implemented share the same importance. Our approach suggests that a specific task can be considered more important given its participation in multiple Investigation Steps and the low completion of said steps. The latter assumes that, the more an Investigation Step is completed, the greater the chance is that the forensic investigation will yield actionable results. Furthermore, if an Investigation Step is not supported by any task, it is very likely that the forensic investigation will fail. Therefore, it is preferable for a service to have tasks that support all Investigation Steps, but do not fully satisfy them, from having Investigation Steps that are fully supported and others that are not supported at all. This method assumes that all steps will yield similarly important results for the investigation process, with more specific assignments of weights to follow, after extensive incident review.

Necessity is calculated with the following method. First, the partial necessity of each investigation is calculated, i.e. how important the implementation of the task is for a specific investigation step, by multiplying the expected completion of the step if the task is implemented and the reverse of the current completion of the step. Finally, the partial necessities of each step are summed to calculate the total necessity of the task. Given that, necessity is calculated thusly, as shown in Eq. (5):

$$n_i = \sum_{k=1}^{K} \frac{1 + FT_k}{TNT_k}(1 - CIS_k) \tag{5}$$

where K is the set of Investigation Steps that the task k is in. This formula can be broken down to two coefficients: (a) $\frac{1+FT_k}{TNT_k}$, calculates the expected completion of the investigation step if the task i is implemented, and (b) $(1 - CIS_k)$, calculates the current completion of investigation step k. This means that (a) is the value of implementing i, and will have a bigger impact the less tasks there are in k. (b) will have a smaller value the larger CIS's value is. This means that for each Investigation Step, the score is higher if that step has less tasks and few of them are implemented. Finally, the score for each Investigation Step is summed, to portray the importance of participating in multiple steps. This means that the methodology is biased towards tasks that participate in multiple steps, which is desirable.

Finally, **Priority** is defined as the relative priority the implementation of a task should be given by an organization. Neither effort, nor necessity, are on their own suitable metrics to help an organization decide which task should they implement next. Necessity is a metric of a task's importance, ignoring the amount of resources required to implement it. Effort is a metric of the work and resources required to complete a task but cannot differentiate between tasks of varying importance. A combination of both is required. Therefore, priority is calculated as illustrated in Eq. (6):

$$P_i = \frac{n_i}{e_i} \tag{6}$$

Priority is defined as the fraction of the necessity of the task divided by the effort required to implement it. Based on this, priority will have maximal score when the task in question has high necessity and low effort scores, and minimal with low necessity and high effort scores.

Priority's score is relative, i.e. it is not representative of absolute states. This means that the scores assigned to each task should only be considered in the context of the specific analysis. Furthermore, by implementing a task, the necessity, and, by definition, the priority, scores of the remaining tasks will change, so this methodology should be redone. An organization should strive to implement the tasks with the highest priority scores first.

4 Case Study and Validation

In order to assess the compliance level of a cloud service, a real case study has been used. The case study concerns the University of the Aegean and it has been introduced in a previous paper [15], where two cloud services (Virtual Machine and Nextcloud storage) were chosen to be implemented as forensic-enabled. Both services concern private clouds. The Virtual Machine service is Infrastructure as a Service (IaaS), while the Nextcloud service is Software as a Service (SaaS). While it might be easier to conduct a forensic investigation on private cloud environments, all deployment models (public, private, etc.) have to satisfy the same forensic constraints. Therefore, forensicablity of the cloud service does not take under consideration the deployment model. An extended validation of this case study is used, so as to identify the level of compliance of the two services with the investigation process.

Table 3. The implemented tasks in cloud services

Investigation step	Number of forensic tasks in the VM cloud service (FTs)	Number of forensic tasks in the NextCloud service (FTs)	Total number of supporting tasks (TNTs)
Confirmation of the incident	3	3	3
Incident identification	2	2	6
Collection and acquisition	3	3	6
Evidence examination & analysis	12	11	14
Presentation	1	1	3
Preservation of evidence	3	3	5
Documentation	5	4	7
Update training & planning libraries	0	0	1

Based on the organizational analysis and the cloud forensic requirements analysis [15], both cloud services were found not to be forensic-enabled, since they do not satisfy all seven forensic constraints. Regarding the Virtual Machine (VM) service, three forensic constraints that are not satisfied were identified, involving seven tasks that are not fulfilled. Regarding the Nextcloud service, four unsatisfied forensic constraints were identified, involving eight tasks. Table 3 is produced, by applying the seven forensic constraints to the cloud services' activity diagrams. The Table presents the number of tasks that are already implemented by the University of the Aegean.

Based on Table 3 and the Eq. (2), the total level of compliance of the two cloud services is presented in Table 4.

Table 4. The compliance level of cloud services

Investigation step	Level of compliance VM cloud service (%)	Level of compliance next cloud service (%)
Confirmation of the incident	100	100
Incident identification	33.3	33.3
Collection and acquisition	50	50
Evidence examination & analysis	85.7	78.6
Presentation	33.3	33.3
Preservation of evidence	60	60
Documentation	71.4	57.1
Update training & planning libraries	0	0
Total compliance level TCL	**54.2**	**51.5**

Based on this we can calculate the necessity score of each unimplemented task using Eq. (5), as seen on Table 5.

Table 5. The necessity score of each unimplemented task

Task	Necessity	
	VM cloud service	NextCloud service
Implement discipline rules	0.67	0.67
Enable access rights	0.65	0.65
Enforce legal contracts	0.67	0.67
Ensure agreements	0.71	0.82
Provide assurance	0.38	0.49
Provide notifications on policy violations	–	0.49
Define SLAs	0.77	0.77
Maintain trained personnel	1.76	1.76

As seen here above, there are some tasks that have much higher necessity than average. This is because the tasks in question take part in multiple investigation steps with few tasks and small CIS. For example, "*Maintain trained personnel*", takes part in three investigation steps, and is the only task that satisfies one of them ("Update training & planning libraries"). This suggests that it is imperative to implement it. On the contrary, “*Provide assurance*” has smaller than average necessity. This is because it takes part in two steps (Documentation and Evidence Examination) both of which have a high CIS and a lot of tasks.

Then, the security team along with the stakeholders assigned values to the effort variable after considering the resources and priorities of the organisation, which enables the calculation of the final “*effort score*” for each task, using Eqs. (3) and (4), resulting in Table 6.

Table 6. The effort score of each unimplemented task

Task	Effort							
	VM cloud service				NextCloud service			
	Cost	Complexity	Standardization	Effort	Cost	Complexity	Standardization	Effort
Implement discipline rules	1	1	1	**1**	1	1	1	**1**
Enable access rights	2.5	2	1.5	**2**	2.5	2	1.5	**2**
Enforce legal contracts	1	2	1	**1.33**	1	2	1	**1.33**
Ensure agreements	1	2.5	6.5	**3.33**	1	2.5	6.5	**3.33**
Provide assurance	5	5	4	**4.66**	5	5	4	**4.66**
Provide notifications on policy violations	–	–	–	**–**	4	4	3	**3.66**
Define SLAs	1	1	1	**1**	1	1	1	**1**
Maintain trained personnel	5.33	7.77	3	**5.33**	5.33	7.77	3	**5.33**

Since, in our case, most unimplemented tasks concern legal issues, such as writing and enforcing agreements, the cost and complexity to implement them is low, since the University has a legal department. However, the “Ensure agreements” task, got a high standardisation score, since there is no global consensus on how to implement it. Similarly, due to the logistic difficulties “Maintain trained personnel”, got a relatively high cost and complexity score. Finally, the priority of each task can be calculated using Eq. (6), as seen on Table 7.

According to our methodology, in both cases, the University should prioritize the implementation of the “Define SLAs” task, which has the highest priority value of 0.77. It is interesting to observe that though “Maintain trained personnel” has a much higher necessity than the rest of the tasks, it still receives a fairly low priority score of 0.33, given its very high effort score. This means that although it is considered quite important for the organization to implement it, its resources are initially better spent on implementing other, easier tasks.

Table 7. The priority score of each unimplemented task

Task	Priority					
	VM cloud service			NextCloud service		
	Necessity	Effort	Priority	Necessity	Effort	Priority
Implement discipline rules	0.67	1	**0.67**	0.67	1	**0.67**
Enable access rights	0.65	2	**0.32**	0.65	2	**0.32**
Enforce legal contracts	0.67	1.33	**0.5**	0.67	1.33	**0.5**
Ensure agreements	0.71	3.33	**0.21**	0.82	3.33	**0.24**
Provide assurance	0.38	4.66	**0.08**	0.49	4.66	**0.10**
Provide notifications on policy violations	–	–	**–**	0.49	3.66	**0.13**
Define SLAs	0.77	1	**0.77**	0.77	1	**0.77**
Maintain trained personnel	1.76	5.33	**0.33**	1.76	5.33	**0.33**

5 Conclusion

The forensic readiness of an organization is of vital importance since it defines the organization's level of preparation in case of an incident. In this paper, an approach is introduced that calculates the cloud forensic readiness of an organization using an efficient algorithm. It presents a decision-making approach that calculates the level of compliance and some other values to classify the unimplemented tasks of a cloud service. The methodology is based on a decision-making algorithm that identifies the importance of all the tasks in a cloud service, so as to classify and prioritize them accordingly. The prioritization of the unimplemented tasks contains useful information for the stakeholders and software engineers since they can take a decision on which tasks they will implement based on specific criteria and methodology. The methodology was then used in a real case study to validate and assess its accuracy.

In the future, our intention is to better study the effort required to implement a specific task, by performing an empirical analysis with academic and industrial stakeholders and studying the problems associated with the currently offered solutions to each task, to assign specific values, in tandem with the values assigned by stakeholders. This will normalise the lack of experience or expertise of the scoring stakeholders regarding a specific solution. Furthermore, the necessity calculation can be improved by analysing incidents to identify the steps which are more crucial to an investigation.

References

1. Martini, B., Choo, K.K.R.: Distributed filesystem forensics: XtreemFS as a case study. Digit. Invest. **11**, 295–313 (2014)
2. Pătraşcu, A., Patriciu, V.V.: Beyond digital forensics. A cloud computing perspective over incident response and reporting. In: 2013 IEEE 8th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 455–460. IEEE, Timisoara (2013)

3. Thethi, N., Keane, A.: Digital forensics investigations in the cloud. In: 2014 IEEE International Conference on Advance Computing (IACC), pp. 1475–1480. IEEE, Gurgaon (2014)
4. Orton, I., Alva, A., Endicott-Popovsky, B.: Legal process and requirements for cloud forensic investigations. In: Ruan, K. (ed.) Cybercrime and Cloud Forensics: Applications for Investigation Processes, pp. 186–229. IGI Global, Hershey (2013)
5. Ruan, K., Carthy, J., Kechadi, T., Crosbie, M.: Cloud forensics. In: Peterson, G., Shenoi, S. (eds.) DigitalForensics 2011. IAICT, vol. 361, pp. 35–46. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24212-0_3
6. Freet, D., Agrawal, R., John, S., Walker, J.J.: Cloud forensics challenges from a service model standpoint: IaaS, PaaS and SaaS. In: Proceedings of the 7th International Conference on Management of Computational and Collective intElligence in Digital EcoSystems (MEDES 2015), pp. 148–155. ACM, Caraguatatuba (2015)
7. Simou, S., Kalloniatis, C., Gritzalis, S., Mouratidis, H.: A survey on cloud forensics challenges and solutions. Secur. Commun. Netw. **9**(18), 6285–6314 (2016)
8. RightScale, State of the Cloud Report 2018: Data to Navigate your Multi-Cloud Strategy. https://www.rightscale.com/lp/state-of-the-cloud. Accessed Mar 2018
9. Tan, J.: Forensic Readiness. Stake, Cambridge (2001)
10. Rowlingson, R.: A ten step process for forensic readiness. Int. J. Digit. Evid. **2**(3), 1–28 (2004)
11. ISO/IEC 27043:2015: Information technology – Security techniques – Incident investigation principles and processes. https://www.iso.org/standard/44407.html. Accessed Mar 2018
12. Alenezi, A., Hussein, R.K., Walters, R.J., Wills, G.B.: A framework for cloud forensic readiness in organizations. In: 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 199–204. IEEE, San Francisco (2017)
13. De Marco, L., Kechadi, M.-T., Ferrucci, F.: Cloud forensic readiness: foundations. In: Gladyshev, P., Marrington, A., Baggili, I. (eds.) ICDF2C 2013. LNICST, vol. 132, pp. 237–244. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14289-0_16
14. Kebande, V., Ntsamo, H.S., Venter, H.S.: Towards a prototype for achieving digital forensic readiness in the cloud using a distributed NMB solution. In: Rodosek, G., Koch, R. (eds.) 15th European Conference on Cyber Warfare and Security (ECCWS 2016), pp. 369–378. Academic Conferences International Limited, Munich (2016)
15. Simou, S., Kalloniatis, C., Gritzalis, S., Katos, V.: A framework for designing cloud forensic-enabled services (CFeS). Requirements Eng. (2018). https://doi.org/10.1007/s00766-018-0289-y
16. Simou, S., Kalloniatis, C., Mouratidis, H., Gritzalis, S.: Towards a model-based framework for forensic-enabled cloud information systems. In: Katsikas, S., Lambrinoudakis, C., Furnell, S. (eds.) TrustBus 2016. LNCS, vol. 9830, pp. 35–47. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44341-6_3
17. Czarnecki, K., Eisenecker, U.W.: Generative Programming: Methods, Tools, and Applications, 1st edn. Addison-Wesley, Boston (2000)