Abstract: Cloud Computing is one of the most pervasive ICT changes of the last few years. Usually, Clouds offer a variety of Services, which are accessible over the Internet. These Services are regulated by some contracts called Service Level Agreements between Service providers and customers. The SLAs have already been introduced in Service Oriented Architectures in situations where some computing services need to be structured and regulated. In an SLA, the constraints of use, the duties and responsibilities of the parties involved, the charges and the service levels to guarantee, etc., are clearly stated by dedicated clauses. Despite the efforts made in systems security and the standardisation of SLAs, Cloud Services continues to suffer from various cybercriminal attacks, and unfortunately this phenomenon is likely to escalate within the next few years. It becomes urgent to take some countermeasures against these illegal practices to increase both the customer trust and quality of services of such new technologies. One of the alternatives for this phenomenon is to provide an efficient cloud Forensic Readiness System (FRS) to prevent and alert the provider and/or customer of any suspect attacks or strange behaviour. Much attention has been given to FRSs and they have certainly moved from simple log files and monitoring to very sophisticated components involving both human experts and computer analysis tools. In this paper we study the effect of SLAs on FRSs. As SLAs may be different from one jurisdiction to another we believe that FRSs should also comply with jurisdiction for more efficiency and speed of isolating and resolving forensic cases. Therefore, we propose an FRS that takes into account automatically SLAs and issue warnings and alerts to its users (providers and consumers) based on the jurisdiction and the nature of security breach and attacks. These SLAs are presented to the system as a set of rules (clauses). This will also prevent illegal data transfers and communications among different jurisdictions. Part of this paper will be dedicated to the formalisation of these SLAs and study its consequences on the FRS architecture and functioning. The rest of the paper will be dedicated to the design and development of the FRS reference architecture integrating the proposed SLA formal model.

Keywords: Cloud Forensic Readiness System, Service Level Agreements, SLA Formal Specification, Cloud Security, Cyber Crimes.

1. Introduction

Digital Forensics (DF) is probably one of the most challenging research topics of the last decades and one of the most interdisciplinary and active research fields. It has been dealing with all the aspects concerning the scientific management of digital crimes in computers and digital devices (Palmer 2001). By following the ICT progresses, different branches of such discipline have been developed, such as network (Palmer 2001), mobile (Jansen and Ayers 2007), and cloud forensics (Ruan et. al 2011), among others. For them, in most cases, some adaptations of traditional tools and procedures have been introduced for properly managing the crimes happened in different computing architectures (Garfinkel 2010; Casey 2011; Ambhrie and Meshram 2012). Instead, in cloud architectures the established forensic techniques and methodologies cannot be successfully applied due to the complex and new challenges that the cloud itself introduced from its weak points (Reilly et. al 2010 and 2011; Ruan et. al 2013; Mishra et. al 2012; Birk and Wegener 2011; Birk 2011; Dykstra and Sherman 2012). Additionally, the cloud involvement in more and more sophisticated cybercrimes is likely to increase (CSA 2013). A helpful from Digital Forensics to prevent cloud crimes can be the adding of a Digital Forensic Readiness (DFR) capability (Tan 2001; Rowlingson 2004; De Marco et. al 2013). Its main concern is about the necessity of minimizing times and costs necessary for conducting a forensic investigation while maximizing the hosting computing architecture capability of being prepared and ready for managing these events. Usually such capability is implemented through an information system that performs some readiness activities, e.g., collection and monitoring of critical and sensitive data; in this manner, the hosting system can be warned and alerted before the crimes happen. In some cases, specific actions can be automatically triggered by system components dedicated to implement DFR. In fact, its importance has been recognised (Grobler and Louwrens 200) and several Forensic Readiness System (FRS) proposals conceived for different computing architectures include this aspect (Endicott-Popovskv et. al 2007; Mouton and Venter 2011;
Valjarevic and Venter 2011; Reddy and Venter 2013; Dykstra and Sherman 2013; Trenwith and Venter 2013; De Marco et. al 2013).

Cloud Computing (CC) is delivered to the users in the form of services regulated by a Service Level Agreement (SLA) contract (Mell and Grance 2011) co-signed by the Cloud Service Provider (CSP) and the Customer. In such contract several clauses are established; they may concern data access rights, behaviour and usage, quality of service constraints, and so on, related to both parties (Baset 2012; Patel at. al 2009). They are written in natural language by using legal jargon and have legal validity in courts (Baset 2012). Due to their structure and contents the SLAs are taken into consideration by the digital forensic community. Indeed, some attempts have been made for representing them in a machine readable format such that the agreed clauses can be automatically monitored. A way for representing natural language-based documents exploits formal specification techniques (Gaudel 1994). Their strength resides in its universality and completeness. We believe that due to the importance that the SLAs own, they must be considered in Cloud Forensic Readiness (CFR); they are necessary to understand what the expected behaviour of the parties is, what the managed data flow is, and what the allowed and denied data transfer and access are, among all the stated clauses. In order to prevent cloud cybercrimes, the SLAs must be properly formalized for being automatically managed by a Cloud Forensic Readiness System (CFRS). The paper is structured as follows: Section 2 illustrates the related works about SLAs formalization; in Section 3 a reference architecture for CFRS is presented; the formalization of some SLA-related concepts necessary for CFRS is proposed in Section 4; Section 5 completes the paper with some conclusions and future works.

2. Related Works

The Service Level Agreement (SLA) contacts have been considered by the forensic community for being automatically monitored in order to guarantee the respect of the included clauses, even though not necessarily for forensic purposes. In most cases, the manner for implementing such natural language-based clauses adopted some formal specifications methods. For instance, in (Czajkowski et. al 2002) the focus is on the design of a protocol for negotiating SLAs among several actors. Different types of SLAs are defined, and some formalism is presented, such as the usage of tuple for describing an SLA. Also some definitions concerning the metrics to use for services are provided. In (Skene et. al 2007) the SLAs are formalized by using set theory for defining the concepts of actions, actors, events, parties, actions’ requirements. The purpose is to determine the possible SLAs’ degree of monitorability in the context of services provisioning through the Internet. In (Paschke and Bichler 2008) a framework called Contract Log for monitoring SLAs is presented, which uses a set of formalisms. The SLAs have been categorized on depending on the purpose they have been written for; then, the contents’ concepts have been abstracted and included in a conceptual framework. Different kinds of rules composing such contracts were identified, such as derivation rules, reaction rules, integrity rules and deontic rules; all of them were included in a homogeneous syntax and knowledge base. Finally, the conceptual framework has been evaluated by a tool running some specific test suites. In (Unger et. al 2008), the concepts of Parties, SLA Parameters, and Service Level Objectives were used for formalizing Services Oriented Architecture SLAs in order to provide a manner for aggregating more SLAs in a single Business Process. The proposal uses the formalisms of tuple, logic predicates, Boole Algebra, and Normal Forms. In (Ishakian et. al 2011) formal specifications were used for representing SLAs and rules transformation in order to address the issue of verifying efficient workload co-location of real time workloads. The approach allows transforming the SLAs whenever they don’t meet the workload efficiency requirements into an equivalent SLA that respects the same QoS. The used formalism is the tuple. The proposal includes a reasoning tool used by the transformation rules’ process that comprehends inference rules based on a database of concepts, propositions, and syntactic idioms. Instead, SLAs monitoring for a Storage as a Service facility is undertaken by (Ghosh and Ghosh 2012) where a design model for a dedicated system was provided. The tuple formalism for representing the SLAs clauses was used; they are decomposed in several services Parameters, established by defining some Service Level Objectives (SLO) describing the QoS levels to guarantee; the SLOs are measured by some Key Performance Indicators (KPI) defined as the atomic metrics to use. Additionally, roughly 30 research projects results are illustrated in a European Community Report (European Commission 2013). Such projects cover different and complementary automated aspects of the SLAs lifecycle, such as specification modelling, management, real-time and storage cloud constraints, SLA enforcement, and others.

3. SLAs in Reference Architecture for a Cloud Forensic Readiness System

A Cloud Forensic Readiness System is represented by the Reference Architecture depicted in the complementary view of Figures 1 and 2. The main components of such a system are represented as rounded rectangles modelling a set of software modules dedicated to specific operations; they communicate each other
via dedicated Open Virtualization Format (OVF) channels, which is a standard language suitable for both the design and the distribution of applications to be executed on different VMs. The CFRS communicates with the underlying Cloud architecture for real-time data collection, respecting existing laws and privacy policies. Such data comprehends both Cloud Services artifacts, outputs from some existing Cloud monitoring tools, and some logs. From the bottom white rectangles in both pictures we can understand the details of these three types of data, defined as Cloud common features by CSA (2011) and considered a “must have” requirement for a CSP that needs to add an FR capability. The collected data will be encrypted and stored in dedicated components of the CFRS thus being compliant with the widely adopted British ACPO (2007) and American National Institute of Justice (NIJ 2008) guidelines concerning the preservation of the potential digital evidence. The whole CFRS activities and modules are constantly running and collecting data for obtaining always up-to-date versions. The Data Management sub-system performs Forensic analysis and knowledge extraction in order to construct a correct and reliable events’ timeline. All this information is fed to the Intrusion Detection sub-component that is responsible to analyze when a Cloud incident is likely to happen. It implements specific policies for managing suspicious behaviors, and considers also the co-signed SLA fed to the CFRS as a set of formal rules. The Intrusion Detection module strongly collaborates with the Events Alerting one, which generates alarms related to the detected suspicious behaviors. The Data Mining module can generate the case related digital evidences; they are then managed by the Preservation of Digital Evidences module, where dedicated policies and routines are implemented. Finally, the chain of custody (CoC) report (NIJ 2008), which is necessary for investigations, is performed by the Chain of Custody sub-system, where additional data-related information, e.g., location, date, time, time zone, and system component, have to be recorded. The CFRS can be responsible to interact with the competent bodies involved in the criminal cases management; indeed, the system may need to transmit the retrieved data belonging to the arisen cases, together with the digital evidences and the chain of custody documents, to pursue with law enforcement activities. For this aim, dedicated interfaces and communication modules are included in the CFRS reference architecture.

The main purpose for needing a system component for SLAs monitoring is that they are considered as a trigger for specific Forensic Readiness actions. On depending on the closeness of the detected Readiness Events to violations, some specific alerts and warnings are launched. In some cases, some crime-preventive actions can be performed, such as data collection, interruption of specific data transmission or storage, among others. They can be automatically instantiated if already provided by specific SLAs’ clauses concerning these situations; in other cases their presence must be properly evaluated.

Figures 1 and 2: CFRS Reference Architecture (De Marco et al 2013, 2014)

4. SLA Formalization
In the Cloud most SLAs are contracts co-signed among a Cloud Service Provider (CSP) responsible of providing some computing services through the Internet, and a Customer who asks for using such services. Additional SLAs negotiations and co-signing situations there exist in service provisioning, e.g., SLAs co-signed among different CSPs for hardware and software resources outsourcing, SLAs involving Third Parties, and so on. Consequently, the customer cannot be aware of the complete flow of his data, because the chain of sub-
services necessary to accomplish his activities and the related SLAs is not disclosed to him. In Figure 3 we can see a possible Customers and Providers interactions governed by SLAs. Obviously the interactions can expand on depending on the Cloud services outsourcing chain.

Figure 3. Possible interactions involving SLAs (Qualemnins)

In this paper we are going to consider the basic case, namely SLAs co-signed among a CSP and a Customer (left hand side of Figure 3). To the best of our knowledge, a standard about the contents of a Cloud SLA does not exist, but there are some common sections identified among most CSP by Baset (2012), where the anatomy of such contracts was drawn. He reaffirms that an SLA is composed of: the provided service levels and the metrics used for guaranteeing them, the services time duration and their granularity, the billing structure, the policy about the level measurement, and the reporting manners about service guarantee violations. Despite this important features’ abstraction and due to the lack of a standard, a comparison among different SLAs is hard to perform. At the light of these considerations, we believe that the identification about what it has to be considered by our SLAs Formalization for Cloud Forensic Readiness proposal is necessary.

4.1 Forensic Readiness Events
A Forensic Readiness system for the Cloud is responsible to implement a Forensic Readiness capability. This is meant to observe and monitor the changes of status of the underlying computing architecture to render it forensically ready. Its principal purpose is to record all the events happening in such environment, namely the Cloud, in order to facilitate the subsequent criminal investigations. Such events representing the changes of status and the operations happened, include some important investigative details, such as, the sender, the recipient, the operation, the date and time. At the light of these considerations we can derive that a Forensic Readiness system is composed of a set of Subjects, Objects, and Actions, representing the set of Forensic events recording the changes of status of a Cloud environment.

4.1.1 Subject
During an FR system life cycle, the recording activity involves the presence of subjects. They are the entities performing operations into a computing architecture. They can be both humans and system processes, for instance an Internet browser session or a Skype ID. Let $s$ be a subject; it belongs to the set of subjects $S$, where $S = \{s_1, s_2, s_3, \ldots, s_n\}, h \in \mathbb{N}$

4.1.2 Object
An Object is the target of an activity performed by a subject. An Object can be both a digital file, and a software or hardware resource. Let $O$ be the set of Objects, $o \in O$ where $O = \{o_1, o_2, o_3, \ldots, o_i\}, i \in \mathbb{N}$ An object $o \in O$ is described by a set of properties and it can be defined as $o = \{p \in P_o \mid o \leftrightarrow p\}$. $P_o$ is the set of properties that can be used to describe an object; example of properties can be filename, date of creation, size and so on. $a_o$ is the relation used for connecting an object $o \in O$ to a property $p \in P_o$ that describes it. Nevertheless, an object $o \in O$ is described by a composition of one or several properties $p \in P_o$; consequently $O \subseteq P(P_o)$, where $P(P_o)$ is the set of all the subsets of $P_o$.

4.1.3 Activity
An activity is an operation executed by an entity with the effect of changing one or more statuses of a system, in our case of a CC environment. Let $A$ be the set of Activities, $a \in A$ where $A = \{a_1, a_2, a_3, \ldots, a_j\}, j \in \mathbb{N}$ In a computing system we can have three main types of activities, namely movement $m$, processing $p$, and storage $s$ (see Figure 4); all of them involve the presence of a subject who is the activity performer and an
The object that is the target of the operation. The movement activities include all the operations that move an object from a digital location to another; the storage activities instead reflect the operations dedicated to store an object into a specific digital location; all the other types of operation are generalized as processing activities. The three types of activities can be defined as a mathematical function that transforms the status of an object of the CC environment. All of them have the same domain and codomain sets, namely the set of objects \( O \) in both cases, because these functions transform an object into another one with some different properties.

| \( m: O \to P(Po) \) |
| \( p: O \to P(Po) \) |
| \( s: O \to P(Po) \) |

**Figure 4. Mathematical Representation of FR Activities**

### 4.1.4 Event

An event is a finite set of activities performed by one or more subjects targeting one or more objects, recorded at a specific time interval. Let \( e \) be an event, let \( E \) be the set of events, then \( E = \{ e_1, e_2, e_3, \ldots, e_k \} \), \( k \in \mathbb{N} \); \( e \) is defined as the tuple \( e = ( S, O, A, t_{start}, t_{end} ) \), where (i) \( S \) is the set of Subjects, (ii) \( O \) is the set of Objects, (iii) \( A \) is the set of Actions, (iv) \( t_{start} \) is the starting time, and (v) \( t_{end} \) is the ending time, where \( t_{start} \leq t_{end} \).

### 4.2 SLA and Cloud Forensic Readiness System

In a CFRS the SLAs have a big importance; they define what are the Cloud Services offered to the Customer by the Cloud Service Provider. An SLA is composed of a set of clauses, which describe all the constraints, behaviours, and duties of the co-signer parties in order to guarantee the predefined Cloud Service(s) level. For instance, some clauses concern the metrics necessary for measuring the described service(s) level attributes, such as latency or average transmission errors rate. A Customer using a Cloud Service can utilize and generate some Cloud data, usually in the form of digital files described by different computing properties, such as the date of creation, size, and type. A clause \( c \) belongs to the set of Clauses \( C \), \( C = \{ c_1, c_2, c_3, \ldots, c_l \} \), \( l \in \mathbb{N} \).

The SLA's clauses are fed as input to a CFRS and represented as formal rules, in particular as predicates of the first order logic, like \( A \to B \). In this paper we assume that such formalization is already performed and the set of formal rules is consistent. Each rule expresses a constraint to be respected by both parties in the context of the Cloud Service(s). For instance, a security constraint can be the allowance of a certain number of a Cloud Service login attempts by the same IP address. Assume that such number is three, and assume that an SLA clause affirms that at the fourth attempt the Cloud Service Provider will deny the possibility of a further login. The related formal rule can be

\[
\text{login attempts} > 3 \rightarrow \text{login denied}
\]

In the CFRS a recording of the events in the form described in Section 4.1.4 will happen: in this case three events representing the three login attempts are recorded; each of them comprehends subjects, objects, actions, start time, and end time.

\[
e_1 = ("IPaddress1", "CloudServiceHomePage", p(CloudServiceHomePage), t_{start1}, t_{end1})
e_2 = ("IPaddress2", "CloudServiceHomePage", p(CloudServiceHomePage), t_{start2}, t_{end2})
e_3 = ("IPaddress3", "CloudServiceHomePage", p(CloudServiceHomePage), t_{start3}, t_{end3})
\]

For all the three events we have that \( IPaddress_i \in S; CloudServiceHomePage \in O; login \) is a processing activity type represented as \( p(CloudServiceHomePage) \). Finally, the three events have their start and end times. In case more than three login attempts will happen, additional FR events similar to \( e_1, e_2, e_3 \) are recorded. On depending on the countermeasures designed for security violations, the system will be allowed to behave in specific manners, even though it is not required by the Forensic Readiness capability itself. In order to demonstrate the positive side effects of the presence of a CFRS, we will illustrate a simple scenario. For instance, in order to avoid the unhallowed login attempts, the system can be capable of raising warnings/alerts to both the CSP and Customer when the second login attempt is recorded, such that the limit condition of three attempts is not reached, and the recorded Cloud behaviour is compliant with the security measures.

There exists a module of the CFRS reference architecture depicted in Figures 1 and 2, namely the Events
Alerting dedicated to generate such alerts. Differently, stronger reactions can be triggered when a third login attempt is recorded; for example, they can be designed to deny a subsequent login action due to the maximum number of attempts is reached. Nevertheless, different behaviours can be implemented in case additional attempts are allowed and FR cannot interrupt the flow of actions happening into the cloud; in this particular option, CFRS is only dedicated to record the events as soon as they are run. Figure 5 depicts the CFRS modules interaction.

![Diagram](image)

**Figure 5.** Excerpt of CFRS Modules Interaction

### 4.3 SLA and FR Formalization

The SLA can be formally represented as an element of the set of SLA. Let \( sla \) be and SLA, it belongs to the set of SLA SLA, \( SLA = \{ sla_1, sla_2, sla_3, \ldots, sla_m \}, m \in \mathbb{N} \).

At the light of all the previous considerations, an SLA can be represented as the following tuple

\[
sla = (csp, csc, t_{start}, t_{end})
\]

where (i) \( csp \) is a Cloud Service Provider, (ii) \( csc \) is a Cloud Service Customer, (iii) \( C \) is the set of Clauses composing the SLA represented as predicates of the first order logic, (iv) \( t_{start} \) is the starting time of the SLA validity and (v) \( t_{end} \) is the ending time of the SLA validity.

Let \( csp \) be a Cloud Service Provider; it belongs to the set of CSPs \( CSP, CSP = \{ csp_1, csp_2, csp_3, \ldots, csp_n \}, n \in \mathbb{N} \).

Let \( csc \) be a Cloud Service Customer; it belongs to the set of CSCs \( CSC, CSC = \{ csc_1, csc_2, csc_3, \ldots, csc_o \}, o \in \mathbb{N} \).

Finally, a Cloud Forensic Readiness capability \( fr \) is an element of the set of Forensic Readiness capabilities \( FR, FR = \{ fr_1, fr_2, fr_3, \ldots, fr_p \}, p \in \mathbb{N} \). An FR capability \( fr \) can be formally represented as the following tuple

\[
fr = (S, O, A, E, sla)
\]

where (i) \( S \) is the set of Subjects, (ii) \( O \) is the set of Objects (iii) \( A \) is the set of Actions, (iv) \( E \) is the set of recorded FR events, and (v) \( sla \in SLA \) is the considered Service Level Agreement contract.

### 5. Conclusions and Future Work

Cloud SLAs management is an emerging research topic; it can be considered as a specific case of natural-language-based document management. The main purpose is to render some sections of such contract as machine-readable and manageable in order to prevent as soon as possible cyber-crimes related to Cloud data. The best manner for achieving this purpose is to exploit mathematical formalisms for abstracting the structure and the contents of SLAs, which are represented in the form of mathematical rules. Such rules are used as input of a CFRS component that is in charge of constantly monitoring the distance between the expected and the actual behaviours. As soon as a reasonable distance is detected, proper Forensic Readiness actions can be triggered. In this paper we focused on a specific case, namely the one involving the existence of an SLA co-owned among a customer and a provider, thus circumscribing the Cloud dimension. We have provided formal definitions for the basic concepts about SLA management and the formalization of some SLA’s clauses. In the future we intend to extend this proposal, in order to cover other cases, namely the ones involving a chain of CSPs, and an intersection of SLAs. One of the achievable purposes is to understand the complete users and providers data flow and paying attention on possible policies and laws violations, so that proper warnings can
be triggered as soon as one of them is likely to be violated. For these extensions, proper formal rules will be provided, in order to demonstrate also the scalability of the approach. The final part of our research work will comprehend a prototype for our set of rules in order to verify if they can be easily monitored and effectively used for preventing some Cloud crimes by witnessing the actual Cloud actors’ and data behaviour.

References


OVF - Open Virtualization Format Standard [online], http://www.dmtf.org/standards/ovf


Qualemnis, “Definitions”, [online], http://www.qualemnis.com/QPI-definitions.htm


