Effective Third Party Auditing in Cloud Computing

Mohammed Hussain and Mohamed Basel Al-Mourad
Department of Interdisciplinary Studies
Zayed University
Dubai, UAE
{mohammed.hussain,basel.almourad}@zu.ac.ae

Abstract—Cloud computing is the next phase in the Internet's evolution, providing the means through which computing power, computing infrastructure, applications, business processes can be delivered to businesses and individual as a service wherever and whenever they need. Businesses who use cloud service providers (CSP) tend to have service level agreements (SLA) that act as contract and define the level of expected service from the CSP, including availability, performance and security. Recent research suggests the use of third party auditors (TPA) as a mean to monitor CSPs. This paper shows how CSP may deceive the SLA to reduce their cost and become more competitive, while avoiding detection by TPAs. The paper presents a crowdsourced TPA model to monitor the CSPs and consequently detect any deception.

Keywords-component: cloud computing; web-service auditing; quality of service

I. INTRODUCTION

Cloud computing is an attractive solution for businesses that require on-demand computation and storage capabilities [1]. Businesses need not to devote time and invest in software and hardware upfront. They may, instead, start their operations immediately. An organization interested in moving its data and services to the cloud needs to negotiate a service level agreement (SLA), with a cloud service provider (CSP). An SLA is considered as a contract between an organization (also known as a cloud user) and a CSP. An SLA specifies the quality of service and security requirements that the CSP needs to guarantee for the user. To ensure that the CSPs adhere the SLAs, cloud users need to audit CSPs. Users may find it more convenient to delegate auditing to a more specialized auditors, called third party auditors (TPA) [2-5]. A TPA periodically checks whether a CSP is adhering or deceiving the agreed SLAs.

Cloud computing is a very competitive industry, where cloud providers are always pressured to provide higher computation and storage capabilities, while reducing costs. Cloud providers are therefore tempted to lower the computational and storage space allocated to their users, without their knowledge. A deceptive CSP may violate the SLA by moving some of its users data and services to a lower quality of service level, saving a great deal of money. Unfortunately, a deceptive CSP may avoid TPA detection by respecting SLAs only when auditors are monitoring that CSP. Avoiding detection is possible due to the fact that a CSP may provide auditors with a preferential access to cloud users data and services, which prove to TPAs that the CSP is adhering the SLA. In reality, however, the services of a cloud user are not allocated with resources, specified by the SLA.

Consider this scenario, where a hotel uploads its booking web-service to a CSP. Based on an SLA the hotel delegates the auditing process to a TPA. The CSP makes a copy of the booking service on a resourceful virtual server, and another copy at a basic virtual server. When the TPA audits the SLA the CSP directs their requests to the resourceful server. Hotel clients are directed to the basic server. In this scenario, the CSP cheats the hotel, while avoiding detection by TPA.

CSPs are also pressured to maintain their reputation to stay competitive in the market. CSPs are therefore, tempted to hide security incidents, which affected their clients’ data and web-services.

Research on cloud service auditing is still in an early stage, and to best of our knowledge the current research doesn't address the above-mentioned problem.

A. Contribution

The contributions of this paper are the following.

- The paper describes a process that allows some CSP to avoid the detection of third party auditors in cloud computing.
- The paper presents a model for auditing CSP, which solves the CSP deception problem.

The paper is organized as follows. Section II defines the problem of auditing cloud service providers, reviews the related work and shows its limitation. Section III presents a model for auditing cloud service providers. Section IV discusses the effectiveness of our model. Section V concludes the paper.

II. AUDITING CLOUD SERVICE PROVIDERS

Security and quality of service are among the top cited concerns over cloud computing [6, 7]. Cloud service providers emphasize the role of security in their solutions [8-10]. Extensive research has been dedicated to ensure cloud users that their data and applications are secure and that quality of service is maintained [11-14]. The following is a brief review of related work on security and quality of service in cloud computing.

Microsoft, Amazon and IBM secure their cloud solutions and monitor the quality of service using the following principals [8-10].
• Risk management is used to assess risks to cloud computing assets, prioritize risks, and select security and quality controls.
• Security controls, as well as quality controls, are implemented. User virtual machines are well separated from each other. Physical machines are protected from being directly controlled by virtual machines.
• A compliance framework is used to monitor and evaluate controls in order to ensure that the controls are operating as required, and that the controls meet industrial and governmental standards.

Lee [11] present a quality model for cloud computing, which defines how quality attributes are calculated and mapped to SLAs. Liu [12] present an agent-based architecture to measure quality attributes at cloud providers and compare them to SLAs. Other research explores different architectures for the cloud. Nallur [13] describe a self-optimizing architecture for quality control. The architecture optimizes resource allocation to increase quality of service.

Since monitoring CPSs is a burden for many types of cloud users, many researchers suggest the use of third party auditors [2-5]. Third party auditors ensure that CSPs are conforming to the security and quality of service requirements, specified by SLAs. Patel [2] propose a framework for managing SLAs, where monitoring quality attributes are delegated to third parties. Wang [3] and Xu [4] introduce a compliance framework to monitor and ensure auditing, while preserving the privacy of the cloud user.

A. Cloud Model and Problem Definition

Figure 1 illustrates the model of cloud computing addressed in this paper. We denote a cloud service provider as $P$, which hosts a set of $n$ business clients $\{C_i\}_{i=1}^n$. For simplicity, $C_i$ is defined as a triplet $(d_i, w_i, s_i)$, where $d_i$ and $w_i$ are the data and web-services of $C_i$ and $s_i$ is the SLA between $C_i$ and $P$. The users of $C_i$ web-services are unaware that $C_i$ has moved its web-service to the cloud. An SLA $s_i$ is defined as a set of $m$ guarantees $\{g_j\}_{j=1}^m$. Each $g_j$ represents a guarantee with respect to various $C_i$ requirements, for example, availability, performance, security and data-location. $C_i$ subscribes to an auditing service $A$. The responsibility of $A$ is to execute a function $f(s_i)$ at $P$, which returns true if $s_i$ holds. The function $g_j(s_i, t)$ returns what $P$ actually guarantees, with respect to $g_j$, at a given time $t$.

**Definition 1. Deceptive CSP:** A CSP is deceptive if the following holds at any time $t$.

$$\exists \{g_j\}_{j=1}^k \subseteq s_i \mid g_j > q(g_j, t) \& \text{true} \leftarrow f(s_i) \quad (1)$$

The following subsection describes two techniques in which $P$ may deceive $C_i$, while avoiding the detection of $A$.

B. How to Deceive a TPA?

Figure 2 illustrates two deception scenarios. The first scenario is based on using two virtual servers to host $C_i$ web-services and data - Figure 2(a). The second scenario is based on the dynamic assignment of SLAs - Figure 2(b).

1) In Figure 2(a), $P$ initializes two virtual servers for $C_i$. The first server is activated only when $A$ is executing $f$. The server operates according to $s_i$; thus, whenever $A$ executes $f(s_i)$, the returned value is always true. The users of $C_i$ web-services are directed, however, to the second virtual server. To save resources, $P$ runs the second server with accordance to $s_i$, an inferior SLA to $s_i$. This is achieved based on detecting the IP addresses that $A$ uses to connect to $C_i$ web services. $P$ uses a function $v(i)$, defined below, which takes the IP address of the requester and determines whether the requester be connected to $IP_1$ (first virtual server) or to $IP_2$ (second virtual server). According to Definition 1, since $P$ runs $C_i$ virtual server with $s_i$, while proving to $A$ that $s_i$ is being upheld, $P$ is a deceptive CSP. This deception reduces the amount of computational and storage resources, which leads $P$ to become more competitive in the cloud computing market.

$$v(i) = \begin{cases} IP_1, i \in A_{IP} \\ IP_2, i \notin A_{IP} \end{cases} \quad (2)$$

2) In Figure 2(b), $P$ initializes one virtual server for $C_i$ based on $s_i$. However, $P$ analyzes the pattern that $A$ follows to execute $f(s_i)$. $P$ may predict a set of time instances $T$, where $A$ will execute $f$. Finally, $P$ uses a function $v(i)$ to decide which SLA to maintain for the virtual server. The function, defined below, allows $P$ to uphold $s_i$ only when $A$ is about to execute $f$. According to Definition 1, $P$ is a deceptive CSP.
Current TPA models [2-5] check the conformance of non-deceptive CSPs to SLAs. Benjamin [14] presents a model to audit SLAs that is immune to deception in grid computing. The model, however, requires grid computing infrastructure; and thus, cannot be used for a cloud computing environment. To the best of our knowledge, detecting deceptive CSPs is still an open problem.

III. A CROWDSOURCED TPA MODEL

The weakness of current TPA models stems from the fact that the TPA directly accesses CSPs for auditing, which makes it feasible for a deceptive CSP to control when to uphold the agreed SLA, and when to uphold an inferior SLA.

The TPA cannot detect the CSP. Therefore, an effective TPA model should be based on TPA auditing CSPs indirectly.

We present a TPA model that is inspired by crowdsourcing [15]. Crowdsourcing is a problem-solving model where an online community of members, of varying knowledge and heterogeneity known as the crowd, participates in a task of variable complexity [16]. The task owner and the crowd share mutual benefit from undertaking the task. Figure 3 illustrates our TPA model that allows A to detect P deception, even if P follows the deception techniques mentioned in Section II. The main property of the model is that the TPA uses intermediate entities to monitor Ci data and web-services.

A. Model Description

In our model, A does not audit P directly; instead, it relies on a crowd of volunteers. A volunteer can be a cloud client whose web-services are hosted at P or a client of another

\[ y(t) = \begin{cases} 
  s_i, & t \in T \\
  x_i, & t \not\in T 
\end{cases} \quad (3) \]
CSP. Further, a volunteer may simply be a user of $C_i$ web-services. $A$ may provide incentives for $C_i$ users to join as volunteers. To simplify model description, we delay the discussion of such volunteers to Section IV. Nevertheless, the main reason why other cloud clients may join as volunteers is to get discounted auditing rates from $A$.

Figure 3 shows our model in action, where two clients, $C_1$ and $C_2$, have subscribed to cloud computing services provided by $P$, according to $s_1$ and $s_2$. Both clients arranged for auditing service from $A$. To lower operation costs, $P$ decided to substitute $s_1$ and $s_2$ with inferior SLAs, namely, $x_1$ and $x_2$. $A$ uses a set of volunteers $V$ to audit $P$. In the first scenario, since $P$ entertains $C_i$ web-data and services with $x_1$, if and only if the IP address of the user does not belong to a TPA. $V$ traffic will be directed to $C_i$ virtual server entertained by $x_1$. When $V$ reports back to $A$, discrepancies between SLAs may be discovered. In the second scenario, since $P$ replaces $x_1$ by $s_2$, if and only if requests are coming from $A$, $P$ will allow $V$ members to access $C_2$ data and web-services while $x_2$ is active. This helps $V$ to detect discrepancies between SLAs and report back to $A$. Thus, in both scenarios, $V$ has a high chance detecting $P$ deception of the SLA.

IV. DISCUSSION

By utilizing the presented model, there is a very high probability that TPAs will detect deceptive CSPs.

A. Model Advantages

The model discourages a CSP from deceiving its clients. This is true because any user request to access client data and web-services maybe in fact originating from an auditing volunteer, rather than just a user. The model also allows TPAs to convince clients to subscribe for TPA services. This is justified by the discount that the clients will enjoy if they volunteer to audit other clients.

B. Model Limitations

The limitations of this model is listed and discussed below.

1) The first limitation is the recruitment of a crowd of volunteers to participate in auditing targeted CSPs.

2) The second limitation is the communication overhead between $A$ and the volunteers, compared to current TPA models in which the TPA communicates directly with the CSP.

3) The third limitation is the reliability of volunteer reports. As volunteers vary in knowledge and experience, it is expected that the reliability of some reports will be questionable.

Creating a mutual benefit for participation solves the first limitation. The TPA may offer volunteers discounts on auditing services. The benefit provided for a volunteer is computed based on the level of participation contributed by that volunteer. Other forms of incentives maybe used too, such as monetary incentives.

The second limitation is reduced if the TPA groups requests before sending it to the volunteers. Allowing the $A$ to send several auditing requests in one message to $V$, rather than sending requests individually can easily alleviate this limitation.

The third limitation can be tackled through several strategies. The first strategy is to ensure reliability by using a number of volunteers to perform the same auditing task, and finding the average results. The second strategy is to inject auditing tasks with assertions to ensure that volunteers are executing the tasks and collecting results according to the agreed protocol.

V. CONCLUSION

Cloud computing is a solution that relieves businesses from the task of managing their IT. Providers of cloud computing services (CSP) manage client data and web-services in accordance to a service level agreement (SLA). CSPs, however, are always under pressure to remain competitive, and may cheat SLAs to lower cost. CSPs are also tempted to hide security incidents to maintain reputation. Businesses may subscribe to a third party auditor TPA to ensure SLAs are fulfilled.

This paper shows how current TPA models are susceptible to CSP deception. The paper presents a crowdsource-based TPA model for cloud computing, which overcomes the CSP deception problem. The feasibility and limitation of the presented TPA model are discussed.

ACKNOWLEDGMENT

This research was supported by Zayed University (RIF Grant R14014).

REFERENCES


