2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications

Reputation Attacks Detection for Effective Trust Assessment Among Cloud Services

Talal H. Noor, Quan Z. Sheng, and Abdullah Alfazi
School of Computer Science, The University of Adelaide, Australia
{*talal, qsheng, alfazi*}*@cs.adelaide.edu.au*

***Abstract*—Consumers' feedback is a good source to help assess overall trustworthiness of cloud services. However, it is not unusual that a trust management system experiences malicious behaviors from its users (i.e., collusion or Sybil attacks). In this paper, we propose techniques for the detection of reputation attacks to allow consumers to effectively identify trustworthy cloud services. We introduce a credibility model that not only identifies misleading trust feedbacks from collusion attacks but also detects Sybil attacks, either strategic (in a long period of time) or occasional (in a short period of time). We have collected a large collection of consumer's trust feedbacks given on real-world cloud services (over 10,000 records) to evaluate and demonstrate the applicability of our approach and show the capability of detecting such malicious behaviors.**

***Keywords*-Trust management, cloud computing, credentials, credibility, reputation, attacks detection, privacy**

I. Introduction

Due to highly dynamic, distributed and non-transparent nature of cloud services, trust management is considered as a challenging problem in cloud environments [1], [2], [3], [4]. SLAs alone are inadequate to establish trust between cloud consumers and providers because of its unclear and inconsistent clauses [5]. For instance, in a recent survey [6], 46.6% of consumers agree that SLA's legal contents are unclear. Many researchers agree that consumers' feedback is a good source to assess trust and proposed trust management techniques based on feedback [3], [7], [8], [9], [10], [11]. However, in reality, it is not unusual that a trust management service experiences malicious behaviors (e.g., collusion or Sybil attacks) from its users [11], [5], [2], [7]. Attackers can trick users into trusting untrustworthy cloud services through creating several accounts, producing numerous transactions (e.g., creating multiple virtual machines for a short period of time), and leaving misleading trust feedbacks. Such manipulation makes it hard for consumers to identify trustworthy providers.

This paper focuses on the detection of reputation attacks to allow consumers to effectively identify trustworthy cloud services. The detection of reputation attacks involves several issues including i) *Consumers Dynamism* where new users join the cloud environment and old users leave around the clock which makes the detection of feedback collusion a significant challenge, ii) *Multiplicity of Identities* where users may have multiple accounts for a particular cloud service[1] which makes it difficult to detect whether a Sybil attack is performed because multiple identities can be used to give misleading information [12], iii) *Attackers Behaviors* where it is difficult to predict when such malicious behaviors take place either in a long or short period of time (i.e., strategic vs. occasional behaviors) [13], and iv) *Consumers' Privacy* where the detection of attacks can make users subject to privacy breaches especially when the interactions involve sensitive information.

In this paper, we overview the design and the implementation of a credibility model that allows consumers to effectively identify trustworthy cloud services. Our model exploits novel techniques that help detect collusion and Sybil attacks without breaching consumers' privacy. In a nutshell, the salient features of our model are i) *Zero-Knowledge Credibility Proof Protocol* that enables the trust management service to prove the credibility of consumers' feedback without breaching consumers' privacy (i.e., without the use of sensitive information); ii) *Collusion Attacks Detection* where we propose several detection metrics including the *Feedback Density* and *Occasional Feedback Collusion* to distinguish between misleading and credible feedbacks no matter attacks occur in a strategic or occasional behavior (i.e., attackers who intend to manipulate the trust results by giving multiple feedbacks to a certain cloud service in a long or short period of time); iii) *Sybil Attacks Detection* where we propose several metrics for the Sybil attacks detection including the *Multi-Identity Recognition* and *Occasional Sybil Attacks* to identify misleading feedbacks from Sybil attacks that occur strategically or occasionally (i.e., attackers who create multiple identities and give feedbacks in a short or long period of time to trick consumers into trusting cloud services that are not trustworthy).

The remainder of the paper is organized as follows. The design of the Zero-Knowledge Credibility Proof Protocol, assumptions and attack models are described in Section II. Section III describes the details of our credibility model. Section IV reports the architectural support for the trust management service and the implementation. Section V reports several experimental evaluations for the proposed techniques. Finally, Section VI overviews the related work and provides some concluding remarks.

[1]It is not uncommon nowadays that a user may have multiple accounts for a particular service such as owning multiple email accounts in Gmail.

978-0-7695-5022-0/13 $26.00 © 2013 IEEE
DOI 10.1109/TrustCom.2013.59

II. Zero-Knowledge Credibility Proof Protocol (ZKC2P)

Since there is a strong relation between trust and identification [14], we propose that the *Identity Management Service* (IdM) can help the *Trust Management Service* (TMS) in measuring the credibility of a cloud service consumer's feedback. However, processing IdM's information can breach the privacy of consumers. Thus, we propose a *Zero-Knowledge Credibility Proof Protocol* (ZKC2P) to allow TMS to process IdM's information (i.e., credentials) using the *Multi-Identity Recognition* factor (explained in detail in Section III-B). TMS processes credentials without including the sensitive information. Instead, anonymized information is used via consistent hashing (e.g., sha-256). The anonymization process covers all the credentials' attributes except the *Timestamps* attribute. The various credentials' attributes are explained in detail in Section II-A.

A. Identity Management Service (IdM)

When consumers attempt to use TMS for the first time, TMS requires them to register their credentials at the trust identity registry in IdM to establish their identities. The trust identity registry stores an identity record represented by a tuple $\mathcal{I} = (\mathcal{C}, \mathcal{C}_a, \mathcal{T}_i)$ for each consumer. $\mathcal{C}$ is the consumer's primary identity. $\mathcal{C}_a$ represents a set of credentials' attributes (e.g., passwords, postal address, IP address, etc.) and $\mathcal{T}_i$ represents the consumer's registration time in TMS. More details on how IdM facilitates TMS in the detection of Sybil attacks can be found in Section III-B.

B. Trust Management Service (TMS)

In a typical reputation-based TMS, consumers either give feedback or request trust assessment for a particular cloud service[2]. From consumers' feedback, the trust behavior of a cloud service is actually a collection of invocation history records, represented by a tuple $\mathcal{H}$ = ($\mathcal{C}$, $\mathcal{S}$, $\mathcal{F}$, $\mathcal{T}_f$), where $\mathcal{C}$ is the consumer's primary identity, $\mathcal{S}$ is the cloud service's identity, and $\mathcal{F}$ is a set of Quality of Service (QoS) feedbacks (i.e., the feedback represent several QoS parameters including availability, security, response time, etc.). Each feedback in $\mathcal{F}$ is represented in numerical form with the range of [0, 1], where 0, 1, and 0.5 means *negative*, *positive*, and *neutral* feedbacks respectively. $\mathcal{T}_f$ is the timestamps when feedbacks are given. Whenever consumer c requests a trust assessment for cloud service s, TMS calculates the trust result, denoted as $\mathcal{T}_r(s)$, from the collected feedbacks as follows:

$$\mathcal{T}_r(s) = \frac{\sum_{c=1}^{|\mathcal{V}(s)|} \mathcal{F}(c,s) * \mathcal{C}_r(c,s,t_0,t)}{|\mathcal{V}(s)|} * (\chi * \mathcal{C}_t(s,t_0,t)) \quad (1)$$

[2]We assume a transaction-based feedback where all feedbacks are held in the trust management service.

where $\mathcal{V}(s)$ denotes all feedbacks given to cloud service s and $|\mathcal{V}(s)|$ represents the total number of feedbacks. $\mathcal{F}(c,s)$ are feedbacks from the c^{th} consumer weighted by the credibility aggregated weights $\mathcal{C}_r(c,s,t_0,t)$ to allow TMS to dilute the influence of misleading feedbacks from attacks. $\mathcal{F}(c,s)$ is held in the invocation history record h and updated in TMS. $\mathcal{C}_t(s,t_0,t)$ is the change rate of trust results in a period of time that allows TMS to adjust trust results for cloud services that have been affected by malicious behaviors. χ is the normalized weight factor for the change rate of trust results which increase the adaptability of the model. More details on how to calculate $\mathcal{C}_r(c,s,t_0,t)$ and $\mathcal{C}_t(s,t_0,t)$ are described in Section III.

C. Assumptions and Attack Models

In this paper, we consider the following two types of attacks i) *Collusion Attacks* also known as *collusive* malicious feedback behaviors, such attacks occur when several vicious users collaborate together to give numerous misleading feedbacks to increase the trust result of cloud services (i.e., a self-promoting attack [15]) or to decrease the trust result of cloud services (i.e., a slandering attack [16]); ii) *Sybil Attacks* that arise when malicious users exploits multiple identities [12], [15] to give numerous misleading feedbacks (e.g., producing numerous transactions by creating multiple virtual machines for a short period of time to leave fake trust feedbacks) for a self-promoting or a slandering attack.

III. Credibility Model

We propose a credibility model which considers several factors for i) the *Feedback Collusion Detection* including the feedback density and occasional feedback collusion, and ii) the *Sybil Attacks Detection* including the multi-identity recognition and occasional Sybil attacks.

A. Feedback Collusion Detection

1) Feedback Density: Malicious users may give numerous misleading feedbacks to manipulate trust results for cloud services (i.e., *Self-promoting* and *Slandering* attacks). For instance, suppose there are two different cloud services x and y and the aggregated trust feedbacks of both cloud services are high (i.e., x has 89% positive feedbacks from 150 feedbacks, y has 92% positive feedbacks from 150 feedbacks). Intuitively, consumers should proceed with cloud service y because it has the highest trust result. However, a *Self-promoting* attack might have been performed on cloud service y, which means x should have been selected instead.

In order to overcome this problem, we introduce the concept of *Feedback Density* to support the determination of credible feedbacks. Specifically, we consider the total number of consumers who gave feedbacks to a particular cloud service as the *Feedback Mass*, the total number of feedbacks given as the *Feedback Volume*. The feedback volume is influenced by the *Feedback Volume Collusion*

factor which is controlled by a specified volume collusion threshold. This factor regulates the multiple feedbacks extent that could collude the overall trust feedback volume. For instance, if the volume collusion threshold is set to 5 feedbacks, any consumer c who gives more than 5 feedbacks is considered to be suspicious of involving in a feedback volume collusion. The feedback density of a certain cloud service s, $\mathcal{D}(s)$, is calculated as follows:

$$\mathcal{D}(s) = \frac{\mathcal{M}(s)}{|\mathcal{V}(s)| * \mathcal{L}(s)} \tag{2}$$

where $\mathcal{M}(s)$ denotes the total number of consumers who gave feedbacks to cloud service s (i.e., *Feedback Mass*). $|\mathcal{V}(s)|$ represents the total number of feedbacks given to cloud service s (i.e., *Feedback Volume*). $\mathcal{L}(s)$ represents the *Feedback Volume Collusion* factor, calculated as follows:

$$\mathcal{L}(s) = 1 + \left(\sum_{h \in \mathcal{V}(s)} \left(\sum_{c=1}^{|\mathcal{V}_c(c,s)|} \frac{\sum_{|\mathcal{V}_c(c,s)| > e_v(s)} |\mathcal{V}_c(c,s)|}{|\mathcal{V}(s)|} \right) \right) \tag{3}$$

$\mathcal{L}(s)$ is calculated as the ratio of the number of feedbacks given by consumers $|\mathcal{V}_c(c,s)|$ who give feedbacks more than the specified volume collusion threshold $e_v(s)$ over the total number of feedbacks received by the cloud service $|\mathcal{V}(s)|$. The idea is to reduce the value of the multiple feedbacks which are given diversely from the same consumer.

2) Occasional Feedback Collusion: Since collusion attacks against cloud services occur occasionally [13], we consider *time* as a factor in detecting occasional collusion attacks (i.e., periodicity). In other words, we consider the total number of feedbacks $|\mathcal{V}(s)|$ given to a particular cloud service s during a period of time $[t_0, t]$. A sudden change in the feedback behavior indicates an occasional feedback collusion because the change of the number of feedbacks given to a cloud service happened abruptly in a short period of time. To detect such behavior, we measure the percentage of occasional change in the total number of feedbacks among the whole feedback behavior (i.e., consumers' behavior in giving feedbacks for a certain cloud service). The occasional feedback collusion factor $\mathcal{O}_f(s, t_0, t)$ of cloud service s in a period of time $[t_0, t]$, is calculated as follows:

$$\mathcal{O}_f(s,t_0,t) = 1 - \left(\frac{\left(\int_{t_0}^{t} |\mathcal{V}(s,t)| \, \mathrm{d}t \right) - \left(\int_{t_0}^{t} \Delta_f(s,t) \mathrm{d}t \right)}{\int_{t_0}^{t} |\mathcal{V}(s,t)| \, \mathrm{d}t} \right)$$

$$where \Delta_f(s,t) = \begin{cases} \mathcal{C}\mu\left(|\mathcal{V}(s,t)|\right) & if\ |\mathcal{V}(s,t)| \geq \\ & \mathcal{C}\mu\left(|\mathcal{V}(s,t)|\right) \\ |\mathcal{V}(s,t)| & otherwise \end{cases} \tag{4}$$

where the first part of the numerator represents the whole area under the curve which represents the feedback behavior for cloud service s. The second part of the numerator represents the intersection between the area under the curve and the area under the cumulative mean of the total number of trust feedbacks. $\mathcal{C}\mu\left(|\mathcal{V}(s,t)|\right)$ represents the mean of all points in the total number of trust feedbacks and up to the last element because the mean is dynamic and changes from time to time. The denominator represents the whole area under the curve. As a result, the occasional collusion attacks detection is based on measuring the occasional change in the total number of trust feedbacks in a period of time. The higher the occasional change in the total number of trust feedbacks, the more likely that the cloud service has been affected by an occasional collusion attack.

B. Sybil Attacks Detection

1) Multi-Identity Recognition: Since consumers have to register their credentials at the *Trust Identity Registry*, we believe that *Multi-Identity Recognition* is applicable by comparing consumers' credentials attributes values from the identity records $\mathcal{I}$. The main goal in this factor is to protect cloud services from malicious consumers who use multiple identities (i.e., *Sybil* attacks) to manipulate trust results. In a typical *Trust Identity Registry*, the entire identity records $\mathcal{I}$ are represented as a list of m consumers' primary identities $\mathcal{C}_p = \{p_1, p_2, ..., p_m\}$ and a list of n credentials' attributes $\mathcal{C}_a = \{a_1, a_2, ..., a_n\}$ (e.g., passwords, IP address, etc.). In other words, the entire $\mathcal{C}_p \times \mathcal{C}_a$ (Consumer's Primary Identity-Credentials' Attributes) Matrix, denoted as IM, covers all consumers who registered their credentials in TMS. The credential attribute value for a particular consumer $v_{c,t}$ is stored in TMS without including credentials with sensitive information using the ZKC2P (see Section II).

We argue that TMS can identify patterns in consumers' anonymous credentials. Malicious users can use similar credentials in different identity records $\mathcal{I}$. Thus, we translate IM to the *Multi-Identity Recognition Matrix*, denoted as $MIRM$, which similarly covers the entire identity records $\mathcal{I}$ represented as the entire $\mathcal{C}_p \times \mathcal{C}_a$. However, the value for a particular consumer $q_{c,t}$ in the new matrix represents the frequency of the credential attribute value for the same particular consumer $v_{c,t}$ in the same credential attribute (i.e., attribute a_t). The frequency of a particular credential attribute value $v_{c,t}$, denoted as $q_{c,t}$, is calculated as the times of appearance (denoted as $\mathcal{A}_p$) that the credential value appears in the t^{th} credential attribute normalized by the total number of identity records (i.e., the length of a_t) as follows:

$$q_{c,t} = \frac{\sum_{c=1}^{c=m} \left(\mathcal{A}_p(v_{c,t}) \right)}{|a_t|} \tag{5}$$

Then, the *Multi-Identity Recognition* factor $\mathcal{M}_{id}$ is calculated as the sum of frequencies of each credential attribute value for a particular consumer normalized by the total number of identity record as follows:

$$\mathcal{M}_{id}(c) = 1 - \left(\sum_{t=1}^{t=n} q_{c,t} \right) \tag{6}$$

where the sum of $q_{c,t}$ represents the similar credentials distributed over different identity records $\mathcal{I}$ and $\mathcal{M}_{id}(c)$ represents the opposite (i.e., at least that the consumer has fairly unique credentials).

2) Occasional Sybil Attacks: Malicious users may manipulate trust results to disadvantage particular cloud services by creating multiple accounts and giving misleading feedbacks in a short period of time (i.e., Sybil attacks). To overcome the occasional Sybil attacks, we consider the total number of established identities $|\mathcal{I}(s)|$ for consumers who gave feedbacks to cloud service s during a period of time $[t_0, t]$. The sudden changes in the total number of established identities is an indicator for an occasional Sybil attack. To detect such behavior, we measure the percentage of occasional and periodic change in the total number of established identities among the whole identity behavior (i.e., all established identities for consumers who gave feedbacks to a particular cloud service). Similarly, the occasional Sybil attacks factor $\mathcal{O}_i(s, t_0, t)$ of a certain cloud service s in a period of time $[t_0, t]$, is calculated as follows:

$$\mathcal{O}_i(s,t_0,t) = 1 - \left(\frac{\left(\int_{t_0}^{t} |\mathcal{I}(s,t)| \, \mathrm{d}t \right) - \left(\int_{t_0}^{t} \Delta_i(s,t) \mathrm{d}t \right)}{\int_{t_0}^{t} |\mathcal{I}(s,t)| \, \mathrm{d}t} \right)$$
$$where \Delta_i(s,t) = \begin{cases} \mathcal{C}\mu\left(|\mathcal{I}(s,t)|\right) & if\ |\mathcal{I}(s,t)| \geq \\ & \mathcal{C}\mu\left(|\mathcal{I}(s,t)|\right) \\ |\mathcal{I}(s,t)| & otherwise \end{cases} \tag{7}$$

C. Feedback Credibility

Based on the proposed credibility metrics, TMS dilutes the influence of misleading feedbacks by assigning the credibility weights $\mathcal{C}_r(c, s, t_0, t)$ to each feedback as shown in Equation 1. $\mathcal{C}_r(c, s, t_0, t)$ is calculated as follows:

$$C_r(c,s,t_0,t) = \frac{1}{\lambda} * (\rho * \mathcal{D}(s) + \phi * \mathcal{O}_f(s,t_0,t) + \Omega * \mathcal{M}_{id}(c) + \iota * \mathcal{O}_i(s,t_0,t)) \tag{8}$$

where ρ and $\mathcal{D}(s)$ denote the *Feedback Density* factor's normalized weight (i.e., parameter) and the factor's value respectively. ϕ and $\mathcal{O}_f(s, t_0, t)$ denote the normalized weight of the occasional feedback collusion factor and the factor's value respectively. Ω denotes the *Multi-identity Recognition* normalized weight and $\mathcal{M}_{id}(c)$ denotes the factor's value. ι denotes the occasional Sybil attacks' normalized weight and $\mathcal{O}_i(s, t_0, t)$ denotes the factor's value. λ represents the number of factors used to calculate $\mathcal{C}_r(c, s, t_0, t)$. For example, if we only consider feedback density, λ will be 1; if we consider all credibility factors, λ will be 4.

D. Change Rate of Trust Results

To allow TMS to adjust trust results for cloud services that have been affected by malicious behaviors, we introduce an additional factor on the change rate of trust results. The idea behind this factor is to compensate the affected cloud services by the same percentage of damage in the trust results. Given $\mathcal{C}on(s, t_0)$ the conventional model (i.e., calculating the trust results without considering the proposed approach by turning $\mathcal{C}_r(c, s, t_0, t)$ to 1 for all trust feedbacks) for cloud service s in a previous time instance, $\mathcal{C}on(s, t)$ the conventional model for the same cloud service calculated in a more recent time instance, the credibility aggregated weights $\mathcal{C}_r(c, s, t_0, t)$, and $e_t(s)$ the attacks percentage threshold. The change rate of trust results factor $\mathcal{C}_t(s, t_0, t)$ is calculated as follows:

$$\mathcal{C}_t(s,t_0,t) = \begin{cases} \left(\frac{\mathcal{C}on(s,t_0)}{\mathcal{C}on(s,t)} \right) + 1 & if\ \mathcal{C}on(s,t) < \mathcal{C}on(s,t_0) \\ & and\ 1 - \mathcal{C}_r(c,s,t_0,t) \geq e_t(s) \\ 0 & otherwise \end{cases} \tag{9}$$

where $\left(\frac{\mathcal{C}on(s,t_0)}{\mathcal{C}on(s,t)} \right)$ represents the change rate of trust results for cloud service s during a period of time $[t_0, t]$. The idea behind adding the number 1 to this ratio is to increase the trust result for the affected cloud services. The change rate of trust results will only be used if the conventional model in the more recent time instance is less than the conventional model in the previous time instance and the attacks percentage during the same period of time $[t_0, t]$ (i.e., $1 - \mathcal{C}_r(c, s, t_0, t)$) is larger or equal to the attacks percentage threshold. For instance, even if the conventional model in the current time for cloud service a is less than the conventional model 10 days ago, cloud service a will not be rewarded because the attacks percentage is less than the attacks percentage threshold (e.g., $1 - \mathcal{C}_r(c, a, t_0, t) = 20\%$ and $e_t(a) = 30\%$). The change rate of trust results is designed to limit the rewards to cloud services that are affected by slandering attacks (i.e., cloud services that have decreased trust results) because TMS can dilute the increased trust results from self-promoting attacks using the credibility factors (i.e., $\mathcal{C}_r(c, a, t_0, t)$). The adaptive change rate of trust results factor can be used to assign different weights using χ the normalized weight factor as shown in Equation 1.

IV. System Implementation

TMS implementation is part of our ongoing project (Cloud Armor[3]), which aims at developing a platform for a reputation-based trust management of cloud services [17], [8]. The platform provides an environment where consumers can give feedback and request trust assessment for a particular cloud service. TMS consists of several components

[3] http://cs.adelaide.edu.au/∼cloudarmor

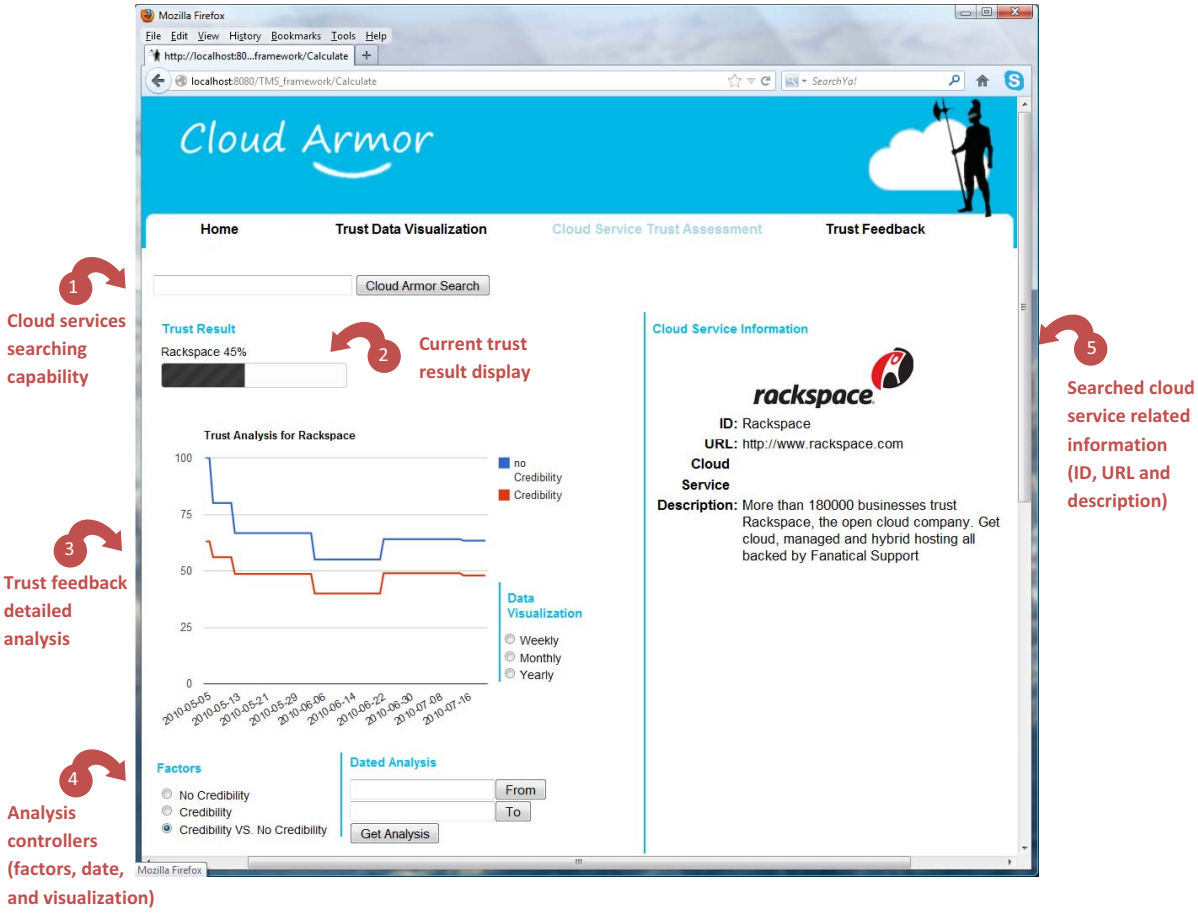


Figure 1. The Interface of the Trust Management Service

namely the *Trust Data Provisioning* and *Trust Assessment Function*.

The *Trust Data Provisioning* is responsible for collecting cloud services and trust information (i.e., the cloud services information includes the cloud service's ID, URL and description which is stored in the Cloud Services Repository to be displayed when consumers search for cloud services (Figure 1, area 5)). We developed the *Cloud Services Crawler* module based on the Open Source Web Crawler for Java (crawler4j[4]) and extended it to allow the platform to automatically discover cloud services on the Internet. In addition, we developed the *Trust Feedbacks Collector* module to collect feedbacks directly from consumers in the form of history records and stores them in the *Trust Feedbacks Database* (i.e., which is held in TMS). Indeed, consumers typically have to establish their identities for the first time they attempt to use the platform through registering their credentials at IdM which stores the credentials in the *Trust Identity Registry*. Moreover, we developed the *Zero-Knowledge Credibility Proof Protocol* (ZKC2P) module to anonymize credentials and store them in the *Identity Recognition Storage* held in TMS.

The *Trust Assessment Function* is responsible for handling trust assessment requests from users (Figure 1, area 1) where the trustworthiness of cloud services are compared and the credibility of trust feedbacks are calculated. We developed the *Trust Assessor* to calculate the trustworthiness of cloud services (i.e., the *Trust Assessor* is designed to request and translate anonymized credentials and recognizes multiple identities using the *Multi-ID Recognition* factor) through requesting the aggregated credibility weights from the *Credibility Calculator* to weigh the feedbacks (Figure 1, area 2 and 3). The trust results for each cloud service and the credibility weights for trust feedbacks are stored in the databases (i.e., *Trust Results and Credibility Weights Storage* within TMS). Furthermore, several analysis controllers are provided for users such as credibility factors in calculating the trust result and the ability to visualize trust results for the cloud service based on different time periods (e.g., in day, month, or year) (Figure 1, area 4).

[4]http://code.google.com/p/crawler4j/

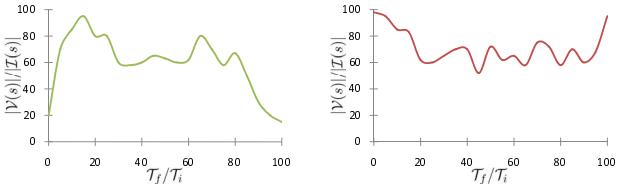
V. Experimental Evaluation

We particularly focused on validating and studying the robustness of the proposed credibility model against different malicious behaviors namely: collusion and Sybil attacks under several behaviors.

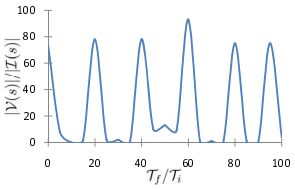
A. Experimental Design and Setup

We validated our credibility model using real-world trust feedbacks on cloud services. We crawled review websites such as *CloudHostingReviewer.com*, *cloud-computing.findthebest.com* and *cloudstorageprovidersreviews.com* where consumers usually give their feedback on cloud services that they have used. The collected data is represented in a tuple $\mathcal{H}$ where the feedback represents several QoS parameters as mentioned earlier in Section II-B and augmented a set of credentials for each corresponding consumer. We managed to collect 10,076 feedbacks given by 6,982 consumers to 113 real-world cloud services. The collected dataset will be release to the research community in the project website.

For experimental purposes, the collected data is divided into 6 groups of cloud services, 3 of which are used to validate the credibility model against collusion attacks, and the other 3 groups are used to validate the model against Sybil attacks where each group consists of 100 consumers. Each cloud service group is used to represent a different attacking behavior model, namely: *Waves*, *Uniform* and *Peaks* as shown in Figure 2. The behavior models represent the total number of malicious feedbacks introduced in a curtain time instance (e.g., $|\mathcal{V}(s)| = 60$ malicious feedbacks when $\mathcal{T}_f = 40$, Figure 2(a)) when experimenting against collusion attacks. The behavior models also represent the total number of identities established by attackers in a period of time (e.g., $|\mathcal{I}(s)| = 78$ malicious identities when $\mathcal{T}_i = 20$, Figure 2(c)) where one malicious feedback is introduced per identity when experimenting against Sybil attacks. In collusion attacks, we simulated malicious feedback to increase trust results of cloud services (i.e., self-promoting attack) while in Sybil attacks we simulated malicious feedback to decrease trust results (i.e., slandering attack). To evaluate the robustness of our credibility model with respect to malicious behaviors (i.e., collusion and Sybil attacks), we use two experimental settings: I) measuring the robustness of the credibility model with a conventional model $\mathcal{C}on(s, t_0, t)$ (i.e., turning $\mathcal{C}_r(c, s, t_0, t)$ to 1 for all trust feedbacks), and II) measuring the performance of our model using two measures namely *precision* (i.e., to know how well TMS did in detecting attacks) and *recall* (i.e., to know how many detected attacks are actual attacks). In our experiments, TMS starts rewarding cloud services that have been affected by malicious behaviors when the attacks percentage reaches 25% (i.e., $e_t(s) = 25\%$), so the rewarding process will occur only when there is a significant damage in the trust result.



(a) Waves (b) Uniform



(c) Peaks

Figure 2. Attacking Behavior Models

We have conducted 12 experiments where 6 of which are conducted to evaluate the robustness of our credibility model against collusion attacks and the other 6 for Sybil attacks. Each experiment is denoted by a letter (e.g., A, B, C, etc.) as shown in Table I.

Table I
Behavior Experimental Design

Malicious Behaviors	Experimental Setting	Waves	Uniform	Peaks
Collusion Attacks	I	A	B	C
	II	A'	B'	C'
Sybil Attacks	I	D	E	F
	II	D'	E'	F'

B. Robustness Against Collusion Attacks

For the collusion attacks experiments, we simulated malicious consumers to increase trust results of cloud services (i.e., self-promoting attack) by giving malicious feedback with the range of [0.8, 1.0]. Figure 3 depicts the analysis of 6 experiments which are conducted to evaluate the robustness of our model with respect to collusion attacks. In Figure 3, A, B, and C show the trust result for experimental setting I, while A', B', and C' depict the result for experimental setting II (experimental settings can be found in Section V-A).

We note that the closer to 100 the time instance is, the higher the trust results are when considering to calculate the trust based on the conventional model. This happens because malicious users are giving misleading feedback to increase the trust result for the cloud service. On the other hand, the trust results show nearly no change when considering to calculate the trust based on the credibility model (Figure 3 A, B and C). This demonstrates that our credibility model is sensitive to collusion attacks and is able to detect such malicious behaviors. In addition, we can make an interesting

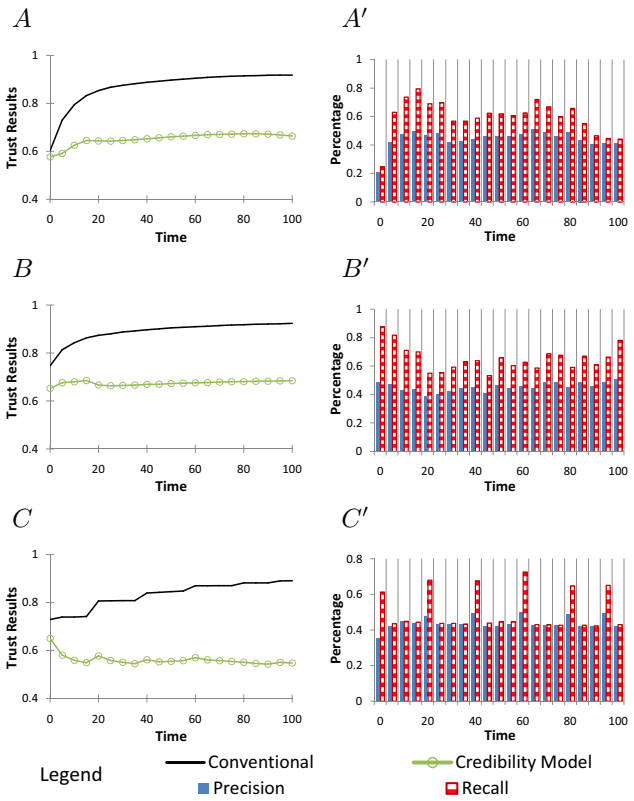


Figure 3. Robustness Against Collusion Attacks Experiments

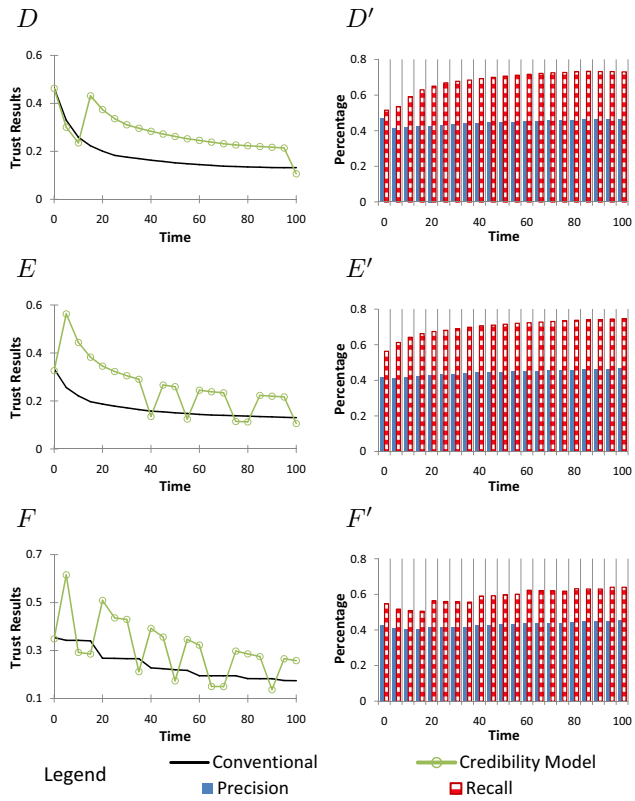


Figure 4. Robustness Against Sybil Attacks Experiments

observation that our credibility model gives the best results in precision when the $Uniform$ behavior model is used (i.e., 0.51, see Figure 3 B'), while the highest recall score is recorded when the $Waves$ behavior model is used (i.e., merely 0.9, see Figure 3 A'). Overall there is a fair degree in recall scores when all behavior models are used which indicate that most of the detected attacks are actual attacks. This means that our model can successfully detect collusion attacks (i.e., whether the attack is strategic such as in *Waves* and *Uniform* behavior models or occasional such as in the *Peaks* behavior model) and TMS manged to dilute the increased trust results from self-promoting attacks using the proposed credibility factors.

C. Robustness Against Sybil Attacks

For the Sybil attacks experiments, we simulated malicious consumers to decrease trust results of cloud services (i.e., slandering attack) by establishing multiple identities and giving one malicious feedback with the range of [0, 0.2] per identity. Figure 4 depicts the analysis of 6 experiments which are conducted to evaluate the robustness of our model with respect to Sybil attacks. In Figure 4, D, E, and F show the trust result for experimental setting I, while D', E', and F' depict the result for experimental setting II (experimental settings details are described in Section V-A).

From Figure 4, we can observe that trust results obtained by considering the conventional model decrease when the time instance becomes closer to 100. This is because of malicious users who are giving misleading feedback to decrease the trust result for the cloud service. On the other hand, trust results obtained by considering our credibility model are fairly higher than the ones obtained by considering the conventional model (Figure 4 D, E and F). This is because the cloud service was rewarded when the attacks occurred. We also can see some sharp drops in trust results obtained by considering our credibility model where the highest number of drops is recorded when the $Peaks$ behavior model is used (i.e., we can see 5 drops in Figure 4 F which actually matches the drops in the *Peaks* behavior model in Figure 2(c)). This happens because TMS will only reward the affected cloud services if the attacks percentage during the same period of time has reached the attacks percentage threshold (i.e., which is set to 25% in this case). This means that TMS has rewarded the affected cloud service using the change rate of trust results factor. Moreover, from Figure 4 D', E' and F', we can see that our credibility model gives the best results in precision when the $Waves$ behavior model is used (i.e., 0.47, see Figure 3 D'), while the highest recall score is recorded when the $Uniform$ behavior model is used (i.e., 0.75, see Figure 3 A'). This indicates that our model can successfully detect Sybil attacks (i.e., either strategic attacks such as in *Waves*

and *Uniform* behavior models or occasional attacks such as in the *Peaks* behavior model) and TMS is able to reward the affected cloud service using the change rate of trust results factor.

VI. Discussions and Conclusion

Over the past few years, trust management has been one of the hot topics especially in the area of cloud computing. Some of the research works use policy-based trust management techniques. For example, Ko et al. [18] proposed TrustCloud framework for accountability and trust in cloud computing which consists of five layers including workflow, data, system, policies and laws, and regulations layers to address accountability in the cloud environment from all aspects. Brandic et al. [19] proposed a novel approach for compliance management in cloud environments to establish trust where the approach is developed using a centralized architecture and uses compliant management technique to establish trust. Unlike previous works that use policy-based trust management techniques, we evaluate the trustworthiness of a cloud service using reputation-based trust management techniques. Reputation represents a high influence that consumers have over a cloud service.

Other research works use reputation-based trust management techniques. For instance, Habib et al. [11] proposed a multi-faceted Trust Management (TM) system architecture which models uncertainty of trust information collected from multiple sources using a set of Quality of Service (QoS) attributes such as security, latency, availability, and customer support. Hwang et al. [3] proposed a security-aware cloud architecture where trust negotiation and data coloring techniques are used to support cloud providers and the trust-overlay networks to support consumers. Unlike previous works which did not consider the problem of unpredictable reputation attacks against cloud services, we present a credibility model that not only detects misleading feedbacks from collusion and Sybil attacks, but also has the ability to adaptively adjust trust results for cloud services that have been affected by malicious behaviors.

In this paper, we have presented novel techniques that help in detecting reputation attacks to allow consumers to effectively identify trustworthy cloud services. We introduce a credibility model that not only identifies misleading trust feedbacks from collusion attacks but also detects Sybil attacks no matter these attacks take place in a long or short period of time (i.e., strategic or occasional attacks respectively). We have collected a large collection of consumer's trust feedbacks given on real-world cloud services (i.e., over 10,000 records) to evaluate and demonstrate the applicability of our approach and show the capability of detecting such malicious behaviors. In the future, we plan to combine different trust management techniques such as reputation and recommendation to increase the trust results accuracy. Performance optimization of TMS is another focus of our future research work.

Acknowledgment

Talal H. Noor and Abdullah Alfazi's work has been supported by King Abdullah's Postgraduate Scholarships, the Ministry of Higher Education, Kingdom of Saudi Arabia.

References

[1] M. Armbrust and et al., "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] T. H. Noor and et al., "Trust Management of Services in Cloud Environments: Obstacles and Solutions," *ACM Computing Surveys (CSUR)*, 2013, in press.

[3] K. Hwang and D. Li, "Trusted Cloud Computing with Secure Resources and Data Coloring," *IEEE Internet Computing*, vol. 14, no. 5, pp. 14–22, 2010.

[4] S. Pearson and A. Benameur, "Privacy, Security and Trust Issues Arising From Cloud Computing," in *Proc. of CloudCom'2010*, 2010.

[5] S. Habib and et al., "Fusion of Opinions under Uncertainty and Conflict - Application to Trust Assessment for Cloud Marketplaces," in *Proc. of TrustCom'2012*, 2012.

[6] F. Dickmann and et al., "Technology Transfer of Dynamic it Outsourcing Requires Security Measures in SLAs," in *Proc. of GECON'2010*, 2010.

[7] Z. Malik and A. Bouguettaya, "Rater Credibility Assessment in Web Services Interactions," *World Wide Web*, vol. 12, no. 1, pp. 3–25, 2009.

[8] T. H. Noor and Q. Z. Sheng, "Credibility-Based Trust Management for Services in Cloud Environments," in *Proc. of ICSOC'2011*, 2011.

[9] F. Skopik and et al., "Trustworthy Interaction Balancing in Mixed Service-Oriented Systems," in *Proc. of SAC'2010*, 2010.

[10] W. Conner and et al., "A Trust Management Framework for Service-Oriented Environments," in *Proc. of WWW'2009*, 2009.

[11] S. Habib and et al., "Towards a Trust Management System for Cloud Computing," in *Proc. of TrustCom'2011*, 2011.

[12] E. Friedman and et al., *Algorithmic Game Theory*. New York, USA: Cambridge University Press, 2007, ch. Manipulation-Resistant Reputation Systems, pp. 677–697.

[13] K. Ren and et al., "Security Challenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[14] O. David and C. Jaquet, "Trust and Identification in the Light of Virtual Persons," pp. 1–103, Jun 2009, accessed 10/3/2011, Available at: http://www.fidis.net/resources/deliverables/identity-of-identity/.

[15] J. R. Douceur, "The Sybil Attack," in *Proc. of IPTPS'2002*, 2002.

[16] S. Ba and P. Pavlou, "Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior," *MIS Quarterly*, vol. 26, no. 3, pp. 243–268, 2002.

[17] T. H. Noor and Q. Z. Sheng, "Trust as a Service: A Framework for Trust Management in Cloud Environments," in *Proc. of WISE'2011*, 2011.

[18] R. Ko and et al., "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," in *Proc. of SERVICES'2011*, 2011.

[19] I. Brandic and et al., "Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds," in *Proc. of CLOUD'2010*, 2010.