Egyptian Informatics Journal 22 (2021) 493–503

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Building a trust model system to avoid cloud services reputation attacks

Salah T. Alshammari *, Aiiad Albeshri, Khalid Alsubhi

Department of Computer Science, College of Computing and Information Technology, King Abdul-Aziz University, Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 8 August 2020
Revised 4 January 2021
Accepted 6 April 2021
Available online 1 June 2021

Keywords:
Trust model
Reputation attacks
Cloud computing
On–off attack
Collusion attack
Sybil attack

ABSTRACT

The safety of cloud services within a Trust Model System (TMS) is certainly compromised by a lack of defense against security threats as well as by inaccuracy of the trust results. Our proposed model addresses well-known security threats to the reputation trust model system, and is shown to deal with all possible potential attack threats, such as Sybil, on–off, and collusion attacks, by specifying the identity of users and tracking activities undertaken by them in order to easily track unauthorized consumers or attackers and to provide proof of any kind of data leakage. The TMS can also oversee the authorization of whoever uploads feedback into the system. It can also identify invalid feedback and discard it from the system. The algorithms of the TMS first establish a variety of trust criteria in which trustworthiness is calculated. Then, feedback from the cloud service provider nodes is accepted only according to the rules of the TMS. A consumer's trust value is finally computed using a flexible system capable of guaranteeing a good balance of consumer trust and owners' feedback. Furthermore, a majority of the existing TMS models do not take full account of interaction importance, thus impeding the accuracy of the trust values, a shortcoming that has been rectified in our proposed model.

1. Introduction

Consumers of cloud computing services are constantly exposed to a variety of threats, including trust and reputation attacks. These threats arise from the highly dynamic, distributed, and non-transparent nature of cloud computing services [1], which makes it extremely challenging for cloud service providers and consumers to maintain and manage trust within the cloud system [2]. Risks can also be attributed to the fact that cloud computing services are offered in a public domain, which gives access to many users. Threats are evident from the behavior of malicious users of cloud services who frequently submit feedback on their experiences with other cloud service consumers.

According to Noor, Sheng, and Alfazi [1], service providers' feedback on their clients is a good source of information that can be used to assess the trustworthiness of cloud service consumers. Varalakshmi, Judgi, and Balagi [3] also highlight the importance of service providers' feedback in computing trust for cloud-based service consumers. Trust management systems and detection systems for reputation attacks have been extensively used by cloud computing providers and consumers to enhance the security and privacy of online data.

Despite the availability of trust management and reputation attack detection systems, cloud computing systems still experience targeted attacks from a variety of sources [4,5,6]. The literature reveals a gap in the availability of detection strategies targeted against reputation attacks [7,8]. This study will therefore provide essential insights on how reputation attack detection can be effectively used for effective assessment in cloud services, and particularly in addressing the reputation attacks that undermine trust assessment in cloud computing. This research will contribute significantly to the existing literature on how consumers and providers of cloud services can enhance the security and privacy of the services, hence improving the overall user experience.

The fact that cloud services are often provided in the public domain makes security and privacy crucial and ought to be main-

* Corresponding author.
E-mail addresses: salshammari0042@stu.kau.edu.sa (S.T. Alshammari), aaalbeshri@kau.edu.sa (A. Albeshri), kalsubhi@kau.edu.sa (K. Alsubhi).

Peer review under responsibility of Faculty of Computers and Artificial Intelligence, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2021.04.001
1110-8665/

tained as public platforms that can be accessed by a wide range of users [9,10,11]. The invention of cloud data storage is certainly revolutionary, as it has eliminated the costliness, inconvenience, and need for more space that additional hardware brings. It allows significantly large amounts of data to be stored at a low cost [12,13]. This has dramatically increased the number of cloud services offered online as well as the number of consumers of such services. However, privacy and security have always been a challenge for digitally available data, as most of the data resides in the public domain, where every model of trust management service is endangered by some security threats [14]. These threats may either elevate the reputation of a specific entity with malicious intentions or entirely ruin it. Security threats that endanger a trust management service may sometimes arise from the consumers of the service themselves. However, in such cases, it can be very difficult to determine which kind of activities are malicious.

1.1. Problem statement

Authorization issues concerning access to cloud computing storage are of serious concern when there are a large number of consumers using the big data of cloud computing for sensitive data [15,16,17]. In most cases this has resulted in the use of access controls in cloud computing server application platforms. Nevertheless, a serious lack of reliability has been discovered on access controls associated with distributed systems. This has been largely caused by failure to establish consumers' identity in advance and their dynamic and complex population [18,19]. Such concerns can be properly attended to through the integration of control models with trust models as a better alternative for decentralized systems [20,21]. These models have resulted from a series of attempts by developers to design new trust models capable of resolving the most intricate and advanced issues of authorization [22,23]. There have been several proposals on integrated trust models with access controls that are still vulnerable to some attacks [24,25]. These threats may either elevate the reputation of any entity with a distinct extent with malicious intentions or entirely ruin it [26]. The security threats endangering a trust management service may sometimes arise from the consumers of the service themselves.

One of the common types of threats in cloud environments is an on–off attack, where the user of a cloud service acts maliciously within a small time period but resumes the proper code of conduct afterward with the aim of deceiving the trust system and maintaining his/her own reputation. Another type of threat is a collusion attack, also called collusive malicious feedback behaviors, in which a group of people acting in concert gives false feedback aimed at ruining the reputation of someone else (a slandering attack) or increasing their own position (a self-promoting attack). Finally, there is the Sybil attack, where the attacker begins by creating multiple memberships within the trust management service. Afterward, he/she make use of these fake identities in giving particular people/entities regular feedback, which could either be positive or negative. As such, the adversary is capable of boosting or ruining the reputation of those entities. This paper focuses on building a Trust Model System (TMS) that ensures the security of a cloud storage system by preventing reputation attacks.

1.2. Contribution

Among the key security requirements in cloud computing is access control. This is due to the fact that user access occurs remotely when an organization transfers its applications to the cloud, which calls for the need to establish access control policies. Several studies have addressed a wide range of access control policies that seem suitable for implementation in cloud computing, but it is still vulnerable to some threats. In order to cover all threats in the proposed trust model, we will propose strategies that must be taken into consideration when we design our trust evaluation process. This paper will endeavor to decide the best possible solution to the trust issues with access control approaches and will propose trust models that can improve the security of information in distributed storage frameworks that utilize cryptographic access control approaches. In the investigation, it was determined that a trust model should provide accurate results in the assessment of trustworthiness, which is the basis of our plan for the proposed trust-based distributed storage framework. The plan permits trust prototypes to be coordinated into a framework that utilizes the cryptographic access control approach. To make this effective, we propose a trust model that contributes to the following:

- Upholding the utmost privacy of the consumers of cloud services, as their association with the trust management services may involve highly sensitive data.
- Effectively protecting cloud services through effective detection of malicious as well as unbecoming behaviors by applying trust algorithms that detect on–off attacks, collusion attacks, and Sybil attacks, where each trust algorithm employs different criteria to avoid reputation attacks.
- Ensuring trust management service availability sufficient to cloud services' dynamic nature.
- Offering reliable solutions for preventing reputation attacks and making consumers' trust values more accurate by taking interaction importance into consideration.

1.3. Related works

The number of techniques used to assess and manage trust for online services has been growing recently. The techniques, according to Noor, Sheng, and Alfazi [1] in 2013, are informed by feedback from consumers of cloud services. The advantage of this study is that the techniques focus on detecting the occasional and periodic reputation attacks that frequently hinder the security and privacy of cloud computing. However, the authors did not focus on how to prevent these attacks.

The evident lack of focus on solutions for periodic and occasional reputation attacks on cloud services can be attributed to the dynamic nature of cloud computing, which is further complicated by the fact that a single consumer may own multiple accounts for accessing a single service. The study by Noor, Sheng, and Alfazi [12] in 2013 provided essential information regarding the efficiency of occasional attack detection models in detecting occasional and period reputation attacks but did not focus on specific attacks such as on–off attacks. In 2014, Noor, Sheng, and Bouguettaya [27] also explored trust management in cloud computing, but their work not discuss how trust management models could avoid the three mentioned attacks. In 2015, Tong, Liang, Lu, and Jin [28] proposed a trust model that considers score value similarity and collusion size score but does not consider the effect of scoring time, nor include provisions to avoid all reputation attacks.

Similarly, in 2015 Labraoui, Gueroui, and Sekhri [29] carried out a study to assess the effectiveness of trust and reputation networks in hindering reputation attacks. The study proposed the on–off (O^2)

trust mitigation for trust systems, which is used in wireless sensor networks. The mitigation system works by penalizing the history of misbehavior on each network node. The penalties affect the trust value of each network node, hence averting the consequences of misbehavior. The main contribution of this study is that it provides essential information regarding the O^2 Trust approach. However, it does not explore whether the same approach would be effective in averting collusion and Sybil Attacks.

In 2018, Ghafoorian, Abbasinezhad-Mood, and Shakeri [30] also explored the use of the role-based access control (RBAC) model as a trust and reputation-based model used to secure data storage in the cloud. The study found that the RBAC model is effective in addressing security threats related to the trust and reputation of cloud-based systems. In another study in 2018 [31], the authors tried to decrease the overhead of the trust model while improving the detection rate of malicious nodes. However, they did not considered all security requirements for the metrics of a reputation-based system.

Other approaches for managing trust and reputation in cloud systems have been explored. In 2019, Nwebonyi, Martins, and Correia [32] and Chang [33] investigated the effectiveness of different models in averting reputation and trust threats in cloud-based systems. However, these studies primarily focus on the overall security and privacy of the system, thereby failing to address some of the specific attacks.

In a 2020 study [34], where the authors proposed a QoS-based model for trust evaluation of cloud service providers by calculating accumulative trust value. However, the authors did not focus on how to avoid all reputation attacks.

1.4. Organization

The rest of this paper is organized as follows. In section 2, we present the material and methods for a solution that prevents on–off, collusion, and Sybil attacks. In section 3, we analyze the components of the trust model. In section 4, we present the simulation results for reputation attacks. Finally, in sections 5 and 6 we discuss the results, along with plans for future works, and provide a conclusion.

2. Material and methods

This study will focus on three types of attacks, which have several similarities, although they differ in how they are perpetrated in a cloud computing environment.

2.1. On-off attacks

In a direct trust model, on–off attacks are characterized by opportunistic malicious behavior at different nodes, which compromises the trust of the system. The behavior switches between good/bad, creating a facade that makes the node pass as trustworthy even when it is behaving badly [29]. For example, a node that passes as trustworthy on an e-commerce site can initiate malicious behavior, remaining undetected because the system initially considered the node trustworthy. First, the trust model system will calculate the interaction trust (IT), which provides an accurate result for the trust value of each cloud service consumer (CR) by computing the service providers ($SP's$) interaction importance (II). The feedback (F) is regarding an interaction is expressed as a percentage. Interaction Trust (IT) is calculated as (1).

$$IT(CR) = \sum_{i=1}^{n-1} \frac{\alpha_i^t(CR) + P_{CR}}{(\alpha_i^t(CR) + P_{CR}) + (\beta_i^t(CR) + N_{CR})} \tag{1}$$

$$\sum_{i=1}^{n} \alpha_i^t(CR) = \begin{array}{c} \\ CR_1 \\ CR_2 \\ \vdots \\ CR_n \end{array} \begin{array}{c} \begin{array}{cccccc} \alpha_1^t & \alpha_2^t & \cdots & \alpha_{n-1}^t & \alpha_n^t \end{array} \\ \begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n-1} & V_{n,n} \end{pmatrix} \end{array}$$

$$\sum i = 1_n \beta i_t(CR) = \begin{array}{c} \\ CR_1 \\ CR_2 \\ \vdots \\ CR_n \end{array} \begin{array}{c} \begin{array}{ccccc} \beta 1_t & \beta 2_t & \cdots & \beta n-1_t & \beta n_t \end{array} \\ \begin{pmatrix} V1,1 & V1,2 & \cdots & V1,n-1 & V1,n \\ V2,1 & V2,2 & \cdots & V2,n-1 & V2,n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Vn,1 & Vn,2 & \cdots & Vn,n-1 & Vn,n \end{pmatrix} \end{array}$$

$P_{CR} = \frac{\alpha_n^t(CR) \times II}{NF}$

$N_{CR} = \frac{\beta_n^t(CR) \times II}{NF}$

Where α^t represents positive feedback at a specific time, β^t represents negative feedback at a specific time, P represents the value of new positive recommender feedback, N represents the value of new negative recommender feedback, II represents the interaction importance value, and NF represents the feedback number.

To avoid the danger of on–off attacks (O^2), we need to add the penalty for an on–off attack (P^{O^2}), which is set from 1 to n, such that 1 represents no danger from a consumer role, and n represents high interaction importance multiplied by the danger rate (DR). The trust model will compute P^{O^2} for any consumer by applying a new procedure, where PC^{O^2} represents the limit of a minimum of the interactions with the greatest value.

$$\begin{cases} \textbf{if } II \geq PC^{O^2} \textbf{ and } \alpha_n^t < II \textbf{ then } P^{O^2} = II \times DR \\ \textbf{else } P^{O^2} = 1 \end{cases}$$

$$IT(CR) = \sum_{i=1}^{n-1} \frac{\alpha_i^t(CR) + P_{CR}}{(\alpha_i^t(CR) + P_{CR}) + (\beta_i^t(CR) + N_{CR} \times P^{O^2})} \tag{2}$$

We also need to add penalty of trust decline (P^{TD}) to avoid dangers of trust decline (TD), where P^{TD} is set from 1 to n, such that 1 represents no danger from this consumer role and ν represents high danger, PC^{TD} represents the curve of a penalty of trust decline with an integer greater than 1, and L^{II} represents the limit of low interaction.

$$IT(CR) = \sum_{i=1}^{n-1} \frac{\alpha_i^t(CR) + P_{CR}}{(\alpha_i^t(CR) + P_{CR}) + (\beta_i^t(CR) + N_{CR} \times P^{O^2} \times P^{TD})} \tag{3}$$

$$\begin{cases} \textbf{if } \alpha_n^t < II \textbf{ then } P^{TD} = PC^{TD} \\ \textbf{else } P^{TD} = 1 \end{cases}$$

$PC^{TD} = \sum_{P^{TD} > L^{II}} P^{TD}$

To avoid this attack, we use PC^{TD} to determine the value of penalty of trust decline (P^{TD}) and we use the danger rate (DR) and the penalty for on–off attacks (P^{O^2}). Algorithm 1 is as follows:

Algorithm 1: On/off Attack Algorithm

Input: F, II;
Output: *Consumer Trust Value*;
procedure *Interaction Trust*
$\alpha_n^t(CR) = F$
$\beta_n^t(CR) = F - 1$
$P_{CR} \leftarrow (\alpha_n^t(CR) \times II)/NF$
$N_{CR} \leftarrow (\beta_n^t(CR) \times II)/NF$
if $II \geq PC^{O^2}$ **and** $\alpha_n^t < II$ **then** $P^{O^2} = II \times DR$
else $P^{O^2} = 1$
endif
if $\alpha_n^t < II$ **then** $P^{TD} = PC^{TD}$
else $P^{TD} = 1$
for $i = 1, i \leq n-1$;
$IT(CR) \leftarrow \left(\alpha_i^t(CR) + P_{CR}\right)/\left(\left(\alpha_i^t(CR) + P_{CR}\right) + \left(\beta_i^t(CR) + N_{CR} \times P^{O^2} \times P^{TD}\right)\right)$
end for
endif
end procedure

We can examine an example of an on–off attack detailed below. The example should exhibit how the penalty of on–off attack and penalty of trust decline impacts the roles' trust values.

Let us assume the last value of $IT(CR) = 0.75$ and $PC^{O^2} = 0.61$, which represents the limit of a minimum of the interactions with the greatest value, and this value should be more than 0.5 to 1.00, which is determined by the cloud administer. We have two conditions to apply this penalty in our formula. The first condition is if the interaction importance (II) is greater than or equal to PC^{O^2}, and the second condition is the new feedback α_n^t is less than the interaction importance (II) of last interaction. If these two conditions hold, then the trust model will compute the penalty of an on–off attack as $P^{O^2} = II \times DR$. Now, let us assume that the value of the new feedback is $\alpha_n^t = 0.60$ and the interaction importance of these interactions is $II = 0.90$and the danger rate is $DR = 3$. In this case, the trust model will compute the penalty of the on–off attack $P^{O^2} = 0.90 \times 3 = 2.7$. Then, the TMS will calculate the new trust value of this consumer as follows:

$$IT(CR) = \sum_{i=1}^{n-1} \frac{\alpha_i^t(CR) + P_{CR}}{(\alpha_i^t(CR) + P_{CR}) + (\beta_i^t(CR) + N_{CR} \times P^{O^2} \times P^{TD})}$$

$$IT(CR) = \sum_{i=1}^{n-1} \frac{0.75 + 0.27}{(0.75 + 0.27) + (0.25 + 0.18 \times 2.7 \times 2)} = 0.45$$

$$P_{CR} = \frac{0.60 \times 0.90}{2} = 0.27$$

$$N_{CR} = \frac{0.40 \times 0.90}{2} = 0.18$$

2.2. Collusion attacks

A collusion Attack is a type of malicious behavior that is characterized by fake feedback from a single node, which is targeted at decreasing or increasing the productivity of ratings of products on an e-commerce site [35]. The behavior can also be non-collusive, whereby the node gives multiple misleading feedback items as a way of self-promoting or slandering another entity. If malicious users comprise greater than 50% in the trust model system, the TMS will be ineffective. This attack also threatens the correctness of recommendation trust values. There are two main types of collusion attacks: a self-promoting attack, where the malicious recommenders cooperate to increase the trust value of a specific consumer in the TMS, and a slandering attack, where the malicious recommenders cooperate to decrease the trust value of a specific consumer in the TMS.

To prevent collusion attacks, we propose a new solution by calculating three criteria representing different factors. The first criterion is malicious recommendations detection (MRD), which detects suspicious recommenders' groups by measuring the probability of a suspicious recommender's group being a collusion recommender's group. In this criterion, the trust model will compute the time range of the collusion attack to determine the time range of all attacks that have happened in a short time. If there are any attacks from malicious recommenders to a specific consumer, the time range of these attacks will be very small. After that, the trust model will calculate the second criterion, malicious recommenders' behavior (MRB), which represents the similarity of malicious recommenders' behaviors, which will be more similar when the malicious recommenders attack a specific consumer. The malicious recommendations detection (MRD) and malicious recommenders' behavior (MRB) are calculated as (4).

$$\begin{cases} \textbf{for } i = 1 \textbf{ t } on - 1 \\ \quad \begin{cases} \textbf{if } T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR \\ \textbf{and } V(F_n, CR)^{FS} \geq V(F_i, CR)^{FS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \leq maxVR \\ \quad \textbf{then move}(F_i, CR)^{FS} \textbf{from FS to SS} \\ \quad \textbf{else if } T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR \\ \textbf{and } V(F_n, CR)^{FS} < V(F_i, CR)^{FS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \geq minVR \\ \quad \textbf{then move}(F_i, CR)^{FS} \textbf{from FS to SS} \\ \textbf{end else if} \\ \textbf{end if} \end{cases} \\ \textbf{endfor} \end{cases}$$

$$\begin{cases} \textbf{for } i = 1 \textbf{ t } on \\ \quad \begin{cases} \textbf{if } T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR \\ \textbf{and } V(F_n, CR)^{FS} \geq V(F_i, CR)^{SS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \leq maxVR \\ \quad \textbf{then move}(F_n, CR)^{FS} \textbf{from FS to SS} \\ \quad \textbf{else if } T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR \\ \textbf{and } V(F_n, CR)^{FS} < V(F_i, CR)^{SS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \geq minVR \\ \quad \textbf{then move}(F_n, CR)^{FS} \textbf{from FS to SS} \\ \textbf{end else if} \\ \textbf{end if} \end{cases} \\ \textbf{endfor} \end{cases}$$

$$\textbf{where} \begin{cases} TR = T(F_n, CR)^{FS} \times TC \\ maxVR = V(F_n, CR)^{FS} \times VC \\ minVR = -V(F_n, CR)^{FS} \times VC \end{cases} \quad (4)$$

The TMS will compare the time and the value of all feedback in the feedback set (FS) that are given to a specific consumer. In the first comparison, the TMS will compare the time of last feedback $T(F_n, CR)^{FS}$ and the time of all $T(F_i, CR)^{FS}$ except for the last feedback. After that, the TMS will compare between the value of the last feedback $V(F_n, CR)^{FS}$ and the value of all $V(F_i, CR)^{FS}$ except the last feedback. Then the trust model will compare between the time $T(F_n, CR)^{FS}$ and the value $V(F_n, CR)^{FS}$ of the last feedback in feedback set (FS) and the time $T(F_i, CR)^{SS}$ and the value $V(F_i, CR)^{SS}$ of all feedback in the suspected set (SS). In this way, the TMS will place all suspected feedback into the suspected set (SS). TC and VC are two parameters to determine the range of feedback time and feedback value.

To detect the malicious recommendations, or collusion feedback, the trust model will compute the third criterion, the collusion attack frequency (CAF) for each recommender who has feedback in the suspected set $SS = \{SF_1, SF_2, \cdots, SF_n\}$, where the higher the frequency of attacks, the greater the strength of the attack.

$$CAF(SR, CR) = \frac{FN(SR, CR)}{FN(SS, CR)} \quad (5)$$

$$\begin{cases} \textbf{if } CAF(SR, CR) \geq FL \textbf{ then move } SF\ (SR, CR) \textbf{ to } CS \\ \textbf{else} \qquad\qquad\qquad\qquad \textbf{move } SF\ (SR, CR) \textbf{ to } FS \\ \textbf{end if} \end{cases}$$

First, the trust model will compute the collusion attack frequency (CAF), where $FN(SR, CR)$ is the number of feedback items that are given from a specific recommender to a specific consumer in the suspected set (SS). $FN(SS, CR)$ is the number of all feedback items in the suspected set (SS) that are given to the same consumer. If the collusion attack frequency (CAF) is greater than the feedback limit (FL), the trust model will move the suspected feedback to the collusion set (CS), or else the trust model will move the suspected feedback $SF(SR, CR)$ from a specific recommender to a specific consumer to the feedback set (FS).

To measure the attack scale (AS) for a specific consumer, the trust model will measure the size of the collusion set (CS), where the malicious recommenders in a recommender's community must account for a large proportion of all recommenders to attack the trust model and cause damage to it. The collusion attack scale (AS) is calculated as (6):

$$AS(CR) = 1 - \frac{RN(CS)}{FN(CS, CR)} \tag{6}$$

where $RN(CS)$ is the number of malicious recommenders in the collusion community and $FN(CS, CR)$ is the number of malicious feedback items for all consumers in the collusion set (CS).

After that, the trust model will compute the attack target scale (ATS), which provides the malicious feedback rate for a specific consumer. The attack target scale (ATS) is calculated as (7):

$$ATS(CR) = \frac{FN(CS, CR)}{FN(AFS, CR)} \tag{7}$$

Where the $FN(CS, CR)$ is the feedback number that has been provided from malicious recommenders. $FN(AFS, CR)$ represents the feedback number that has been provided from all feedback sets in same community, where the set of malicious recommenders is part of all recommenders who evaluated a specific consumer.

Finally, to measure the strength of all collusion attacks for a specific consumer, the trust model will compute the collusion attack strength (CAS) by taking the number of all attack scale from different communities. Collusion attack strength (CAS) is then calculated as (8):

$$CAS(CR) = \frac{\sum_{i=1}^{n} FN(CS_i, CR)}{\sum_{i=1}^{n} FN(AFS_i, CR)} \tag{8}$$

To detect collusion groups, the trust model will use the collusion attack algorithm below:

Algorithm 2: Collusion Attack Algorithm

Input : FS, TC, VC, FL;
Output : $SS, CS, AS(CR), ATS(CR), CAS(CR)$;
procedure *collusion attack*
$TR \leftarrow T(F_n, CR)^{FS} \times TC$
$maxVR \leftarrow V(F_n, CR)^{FS} \times VC$
$minVR \leftarrow -V(F_n, CR)^{FS} \times VC$
for $i = 1$ ***to*** $n - 1$
if $T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR$ ***and*** $V(F_n, CR)^{FS} \geq V(F_i, CR)^{FS}$ ***and*** $V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \leq maxVR$ ***then move*** $(F_i, CR)^{FS}$ ***from*** FS ***to*** SS
else if $T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR$ ***and*** $V(F_n, CR)^{FS} < V(F_i, CR)^{FS}$ ***and*** $V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \geq minVR$ ***then move*** $(F_i, CR)^{FS}$ ***from*** FS ***to*** SS
endelse if
endif

* (*continued*)

Algorithm 2: Collusion Attack Algorithm

endfor
for $i = 1$ ***to*** n
if $T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR$ ***and*** $V(F_n, CR)^{FS} \geq V(F_i, CR)^{SS}$ ***and*** $V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \leq maxVR$ ***then move*** $(F_n, CR)^{FS}$ ***from*** FS ***to*** SS
else if $T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR$ ***and*** $V(F_n, CR)^{FS} < V(F_i, CR)^{SS}$ ***and*** $V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \geq minVR$ ***then move*** $(F_n, CR)^{FS}$ ***from*** FS ***to*** SS
endelseif
endif
endfor
for $i = 1$ *to* n
$CAF(SR_i, CR) \leftarrow FN(SR_i, CR)/FN(SS, CR)$
if $CAF(SR_i, CR) \geq FL$ ***then move*** $SF(SR_i, CR)$ ***to*** CS
else
move $SF(SR_i, CR)$ ***to*** FS
endif
endfor
$AS(CR) \leftarrow 1 - RN(CS)/FN(CS, CR)$
$ATS(CR) \leftarrow FN(CS, CR)/FN(AFS, CR)$
for $i = 1$ *ton*
$CAS(CR) \leftarrow FN(CS_i, CR)/FN(AFS_i, CR)$
endfor
endprocedure

We can examine the example of the collusion attack detailed below. The example will show how the collusion attack algorithm impacts the roles' trust values. Let us assume the below matrix contains all feedback times.

$$\sum_{i=1}^{n} FS(CR) = \begin{array}{c} \\ SP_1 \\ SP_2 \\ \vdots \\ SP_n \end{array} \begin{array}{c} \begin{array}{cccc} T(F_1) & \cdots & T(F_{n-1}) & T(F_n) \end{array} \\ \begin{pmatrix} V_{1,1} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & \cdots & V_{n,n-1} & V_{n,n} \end{pmatrix} \end{array}$$

$$\sum_{i=1}^{n} FS(CR) = \begin{array}{c} \\ SP_1 \\ SP_2 \\ \vdots \\ SP_n \end{array} \begin{array}{c} \begin{array}{cccc} T(F_1) & \cdots & T(F_{n-1}) & T(F_n) \end{array} \\ \begin{pmatrix} 04:22 & \cdots & 05:09 & 14:22 \\ 04:50 & \cdots & 05:12 & 14:50 \\ \vdots & \ddots & \vdots & \vdots \\ 01:30 & \cdots & 20:01 & 22:30 \end{pmatrix} \end{array}$$

The algorithm will compare all feedback time, starting with the last feedback time $V_{n,n}$. Let us assume the time range is two hours and all feedback is on same day. In this case, there are six feedback items in this time range, and the trust model will compare their values. Let us assume the following feedback values:

$$\sum_{i=1}^{n} FS(CR) = \begin{array}{c} \\ SP_1 \\ SP_2 \\ \vdots \\ SP_n \end{array} \begin{array}{c} \begin{array}{cccc} V(F_1) & \cdots & V(F_{n-1}) & V(F_n) \end{array} \\ \begin{pmatrix} 90\% & \cdots & 99\% & 98\% \\ 92\% & \cdots & 95\% & 97\% \\ \vdots & \ddots & \vdots & \vdots \\ 61\% & \cdots & 99\% & 82\% \end{pmatrix} \end{array}$$

In this case, the system will move the six feedback items to the suspected set (SS); then, the trust model will compute the collusion attack frequency for each recommender who has feedback in the suspected set (SS).

$CAF(SR, CR) = \frac{FN(SR,CR)}{FN(SS,CR)}$
$CAF(SR, CR) = \frac{3}{6} = 50\%$

With the below condition, if the feedback limit $FL = 10\%$, then the trust model will move these feedback items to the collusion set (CS), or else will move the feedback to the (FS).

$$\begin{cases} \textit{if } CAF\ (SR, CR) \geq FL \textit{ then move } SF(SR, CR) \textit{to } CS \\ \textit{else} \qquad\qquad\qquad \textit{move } SF\ (SR, CR) \textit{ to } FS \\ \textit{end if} \end{cases}$$

2.3. Sybil attacks

According to Mahajan, Mahajan, Jadhav, and Kolate [35], a Sybil attack involves the emulation of multiple behaviors by a single entity [36], typically by multiple accounts using a single entity/node. The multiple identities can be stolen or fake, or even obtained through a Sybil attack. Sybil attackers can use multiple entities to either damage or build the reputation of an online system by giving either negative or positive feedback, respectively. The literature will be systematically reviewed for the effectiveness of detecting reputation attacks so as to prevent Sybil Attacks.

To avoid this attack, we need to apply two kinds of attack detection, multi-identity detection (M_{id}) and Sybil attack detection(SA). We have all users' credentials in the trust identity registry, and by comparing all credentials attributes for all users we can avoid this attack, where R represents a list of identity records for all users' primary identities $UP = \{PI_1, PI_2, \cdots, PI_m\}$ and the credentials attributes for all users $UC = \{CA_1, CA_2, \cdots, CA_n\}$, UM is $UP \times UC$ matrix which contains the credentials (CA) of all users who are registered in the TMS. The attacker will use the same credentials in various identity records, and the TMS will identify patterns in anonymous consumers' credentials.

The TMS will detect this attack by calculating the value of multi-identity detection (M_{id}) as the appearance times (Q) of credential attribute value (CA), where the value of the credential is shown by the credentials attribute divided by the number of all records with the user's identity (R_{id}). The frequency of any credentials attribute (CA) is represented by the number of similar credentials. (M_{id}) is calculated as in (9):

$$M_{id} = \sum_{t=1}^{t=n} \left(\sum_{CA=1}^{CA=n} \frac{Q(CA_n)}{R_{id}} \right) \tag{9}$$

To make the TMS ignore all fake trust results, we need to apply a new procedure, where we identify the recommendation trust value at a previous time as $RT(F(t))$ and that at the present time as $RT(L(t))$. The TMS will foil this attack by using the limit of a maximum number of records (RL) that has the same value of the credential attribute; for example, if there are many records that have the same credential attribute registered within a specific time frame, the TMS will check whether the number of these records is greater than RL and ignore all feedback that comes from these recommenders. Finally, the change rate factor of the trust result $RT(CR, F(t), L(t))$ measures the consumer trust value by taking into consideration the behavior of all recommenders in a specific time, calculated as (10):

$$RT(CR, F(t), L(t)) = \tag{10}$$

$$\begin{cases} \textbf{\textit{if}} \ \ M_{id} \geq RL \ \textbf{\textit{then}}\ RT(F(t)) = RT(F(t)) \\ \textbf{\textit{else}} \qquad\qquad\quad RT(F(t)) = RT(L(t)) \end{cases}$$

We can examine the example of the Sybil attack detailed as follows:

$$M = \begin{array}{c} \\ PI_1 \\ PI_2 \\ \vdots \\ PI_m \end{array} \begin{array}{c} \begin{array}{cccc} CA_1 & CA_2 & \cdots & CA_n \end{array} \\ \begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m,1} & V_{m,2} & \cdots & V_{m,n} \end{pmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ PI_1 \\ PI_2 \\ \vdots \\ PI_m \end{array} \begin{array}{c} \begin{array}{cccc} CA_1 & CA_2 & \cdots & CA_n \end{array} \\ \begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m,1} & V_{m,2} & \cdots & V_{m,n} \end{pmatrix} \end{array}$$

where PI_m represents the primary identities for all consumers (for example, the consumer's user name) and CA_n represents the credential's attributes (the consumer's IP address, passwords, computer name, etc.). $Q(CA_n)$ is the frequency of any credential attribute CA represented as similar for all consumers. If M_{id} is greater than or equal to RL, then the TMS will ignore all fake trust results that come from different accounts that have the same credential attribute (CA).

Secondly, we need to use the time factor by calculating the number of identities $|N_{id}(t)|$ within a specific time frame$[F(t), L(t)]$ for all recommenders who send their feedback to the TMS. If there is any change in the recommender's behavior at a specific time, this situation indicates a Sybil attack. The TMS will calculate the percentage of occasional and periodic changes in the number of identities in the behavior of all identities (the identities for all recommenders that provided feedback to any consumer). SA is calculated as (11):

$$C(t) = \lfloor N_{id}(F(t)) \times SC \rfloor$$

$$\sum_{t=1}^{t=n} \begin{pmatrix} \textbf{\textit{if}}\ N_{id}(L(t)) > N_{id}(F(t)) + C(t)\ \textbf{\textit{then}} \\ SA(F(t), L(t)) = \frac{N_{id}(L(t)) - N_{id}(F(t)) + C(t)}{|N_{id}(F(t))|} \\ \textbf{\textit{else}} \qquad\qquad\qquad \textbf{\textit{do nothing}} \end{pmatrix} \tag{11}$$

where$C(t)$ represents the identities curve, SC represents the Sybil attack curve, $N_{id}(F(t))$ represents the number of identities at the beginning of the time frame, and $N_{id}(L(t))$ represents the number of identities at the end.

As an example, let us assume the number of identities at the beginning of the time frame $N_{id}(F(t)) = 421$ identities, $N_{id}(L(t)) = 487$, and $SC = 5\%$. Then, $SA(R, F(t), L(t))$ is calculated as follows:

$$C(t) = \lfloor 421 \times 0.05 \rfloor = 21$$

$$\sum_{t=1}^{t=n} \begin{pmatrix} \textbf{\textit{if}} \qquad 487 > 421 + 21 \qquad \textbf{\textit{then}} \\ SA(F(t), L(t)) = \dfrac{487 - 421 + 21}{421} \\ SA(F(t), L(t)) = 10\%\ \textit{of identities number} \\ \textit{more than identities curve} \\ \textbf{\textit{else}} \qquad\qquad\qquad \textbf{\textit{do nothing}} \end{pmatrix}$$

Finally, the trust model will compute the recommender importance value, which means SPs are often influential depending on their trustworthiness as recommenders and their experience $W(R_i, CR)$ by adding a weight function of the exchange transaction (W^{EX}) and exchange transaction value (EX_i) in the final equation, which measures the value of the number of exchange transactions for any recommender with a specific consumer. Another two factors will be associated in this equation to measure the impact of time (t) in the recommender interaction, where the last time the service provider interacted with a specific consumer $last(t)$ is multiplied by a weight function for the last time (W^L). The trust value of any consumer $RT(CR)$ can be calculated as below, where the feedback value is $F(R_i, CR)$ for all recommenders who have enough experience to give feedback. To eliminate the challenge of all attacks from the posited trust model, we need to add the time factor (12).

$$W(R_i, CR) = \sum_{i=1}^{n} \frac{\left(EX_i \times W^{EX} + L_i(\prime) \times W^L\right)}{100}$$

$$\textit{where } W^{EX} + W^L = 100$$

$$\begin{cases} \textbf{if } W(R_i, CR) \geq II \textbf{ then } RT(CR, F(t), L(t)) = \frac{\sum_1^n F(R_i, CR)}{N^R} \\ \textbf{else do nothing} \end{cases} \quad (12)$$

As an example, consider the interaction importance $II = 0.80$, $W(R_i, CR) = \{0.88, 0.72, 0.90, 0.96\}$, and recommender feedback is $F(R_i, CR) = \{0.90, 0.89, 0.30, 0.67\}$. With this scenario, the consumer trust value is calculated as below.

$W(R_i, CR) = \sum_{i=1}^{n} \frac{(0.91 \times 80 + 0.77 \times 20)}{100} = 0.88$

if $0.88 \geq 0.70$ **then** $RT(CR, F(t), L(t)) = \frac{0.90+0.30+0.67}{3} = 0.62$ *or* 62%

Algorithm 3: Sybil Attacks Algorithm

Input : RL, W^{EX}, W^{L};
Output : RT;
procedure *Sybil Attacks*
for $t = 1$ *ton*
for $CA = 1$ *ton*
$M_{id} \leftarrow Q(CA_n)/R_{id}$
if $M_{id} \geq RL$ ***then*** $RT(F(t)) = RT(F(t))$
else
$RT(F(t)) = RT(L(t))$
end if
end for
endfor
$C(t) = \lfloor N_{id}(F(t)) \times SC \rfloor$
for $t = 1$ *ton*
if $N_{id}(L(t)) > N_{id}(F(t)) + C(t)$ ***then***
$SA(F(t), L(t))$
$\leftarrow (N_{id}(L(t))$
$-N_{id}(F(t)) + C(t))$
$/|N_{id}(F(t))|$
endif
endfor
for $i = 1$ ***to*** n
for $t = 1$ ***to*** n
$W(R_i, CR) = \left(EX_i \times W^{EX} + L_i(\checkmark) \times W^{L}\right)/100$
if $W(R_i, CR) \geq II$ ***then*** $RT(CR, F(t), L(t)) = F(R_i, CR)^{SP}/N^R$
end if
endfor
endfor
end procedure

3. Proposed framework architecture

This section provides an analysis of the different components of the trust model and the role it plays in ensuring that the system works effectively. Fig. 1 illustrates the proposed system architecture.

TMS will be the component of the proposed design that evaluates the extent to which providers of the cloud services are willing to depend on the consumers of the cloud services. It will also be entitled to the provision of certain cloud services as promised by the cloud service providers. The trust management system is composed of various subsections entitled with different tasks, all of which are aimed at ensuring the security and the privacy of the data in the cloud storage systems. It evaluates the rules that are determined within the level of trust depending on feedback from service providers. There is a need to specify the identity of the user and track the activities of this entity for easy tracking of unauthorized consumers or attackers and issuance of proof of any form of data leakage. Through the TMS, the accounts of registered as well as recognized consumers are updated and all the incidents in which a consumer is involved are also determined. This component will also oversee the authorization of whoever uploads the feedback into the system. It is capable of identifying invalid feedback as well as discarding such invalid feedback from the system.

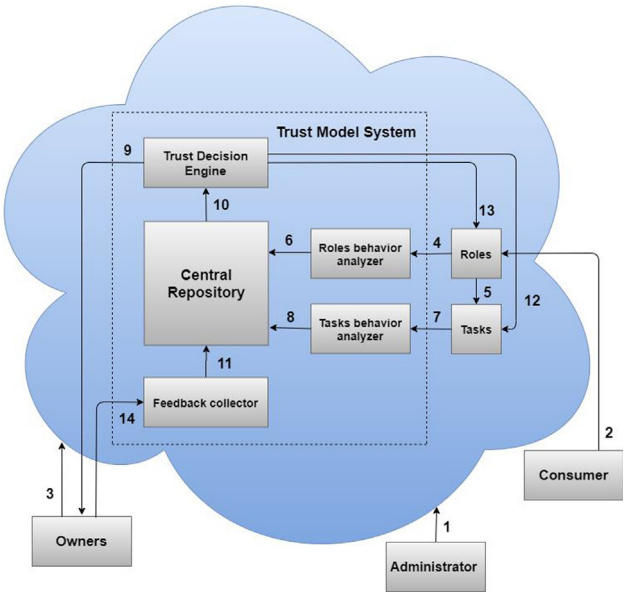

Fig. 1. Trust model system architecture.

Trust Management System (TMS): This adds to the layers of the trust model, whose main aim is to make the entire system operate effectively and securely. The trust management system is composed of various subsections as elaborated below.

Central Repository: This assumes the role of interaction store, mainly operating to store all types of trust records and interaction histories generated by interacting tasks and roles for later utilization by the trust decision engine in the evaluation of the tasks and roles values.

Role Behavior Analyzer: This serves to analyze the functions and roles associated with minimum levels of trust regulations in relation to shared resources. Its role is to evaluate rules that are determined within the level of trust depending on the feedback that arises from service providers in the central repository. The role behavior analyzer entity links the roles in order to collect information concerning them and report any form of leakage if it arises. This entity fills the need to specify the identity of the user and track the activities undertaken for easy tracking of unauthorized consumers or attackers and issuance of proof of any form of data leakage. The role behavior analyzer also updates the accounts of registered as well as recognized consumers and assesses all of the incidents in which a consumer is involved.

Task Behavior Analyzer: This is responsible for analyzing tasks and functions concerning minimum trust level regulations upon accessing shared resources. It evaluates tasks determined within the trust level according to the feedback from owners through computing trust value and stores this value in the central repository. It listens to channels to collect information, the two channels in this case, including reports from the tasks regarding leakage of data and reports from the role behavior analyzer to identify the histories of the consumers regarding the stored data. The task behavior analyzer needs to specify identifying consumers and tracking the tasks performed. It can easily track attackers or unauthorized consumers and issues proof of any data leakage. It will also update registered and recognized consumers' accounts and determine whether a consumer account has been involved in the incident.

Feedback Collector: This is an agent tasked with the collection of feedback from the service owners to the repository headquarters before it is automatically allocated. Consumers' trustworthiness is represented by the feedback on tasks and roles. For the purposes of security, the collector of roles and task feedback protects its integrity. This component serves the purpose of ensuring the authorization of whoever uploads the feedback into that system. It is capable of identifying invalid feedback as well as discarding such invalid feedback from the system. Furthermore, the collector of feedback gathers information concerning data assignments on tasks and roles before updating it to the central repository on the nature of the assignment.

Trust Decision Engine: This is a component that evaluates and determines the value of trust of the data owners, the roles, and the task entities. It gathers every form of information concerning the interaction histories that arise from the repository center and a particular consumer's trust values output before making a decision on the nature of the response required by the system.

4. Simulation results

We built a C#.net Windows Forms application to compare our architecture with several related works. In this section, we will evaluate through experiments the ability of the trust model to withstand reputation attacks. The TMS will compute penalties of an on–off attack by using two conditions. The first conditions, if the interaction importance greater than or equal to the PC^{O^2}, where PC^{O^2} represents the limit of a minimum of the interactions with the greatest value. The second condition if the feedback value (F) is less than interaction importance (II) then the trust model will compute the penalty of on–off attack (P^{O^2}). If the feedback (F) of the recommender is less than the interaction importance (II) then the trust model will compute the trust decline penalty (P^{TD}). Fig. 2 shows the impact of the penalty of on–off attack and trust decline in the interaction trust values Table 1.

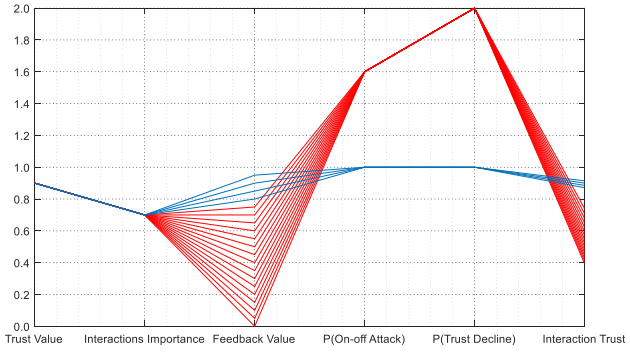

Fig. 2. Penalties of on–off attack and trust decline.

We tested big data to check the standard NIST metrics before and after adding the penalties to equations (1), (2), and (3) to check the five security functions. As we see in Fig. 3, due to the differences between these three equations, attacks will occur if the penalties are not added. The impact on the trust value of users should be significant in the event of malicious behaviors in important transactions.

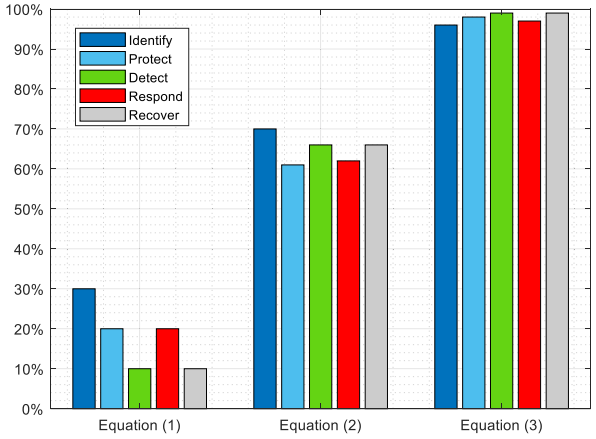

Fig. 3. Penalties of on–off attack and trust decline.

The trust model will compute the Interaction Trust value for malicious consumers by applying the penalties of malicious behavior. Fig. 4 and Fig. 5 show the impact of new feedback on the value of interaction trust (IT) for malicious consumers and trusted consumers.

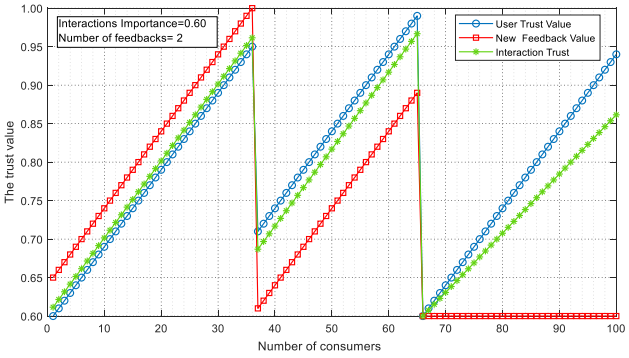

Fig. 4. Interaction trust values for 100 trusted consumers.

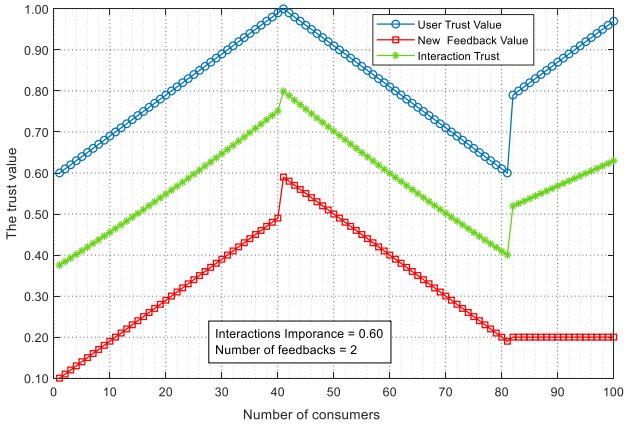

Fig. 5. Interaction trust values for 100 malicious consumers.

To avoid collusion attacks, the TMS will compute the collusion attack frequency (*CAF*), where the value of feedback frequency is directly proportional to the feedback collusion and inversely proportional to the feedback's credibility. The frequency of feedback depends on the number of recommender feedback items and the number of all feedback items in the suspected set (*SS*).

Fig. 6 shows the feedback frequency of seven suspected recommenders. We can see that five suspected recommenders have feedback frequency greater than the feedback limit (*FL*), which means the TMS will move the feedback of these recommenders to the collusion set (*CS*), or else the trust model will move the suspected feedback by a specific recommender to a specific consumer to the feedback set (*FS*).

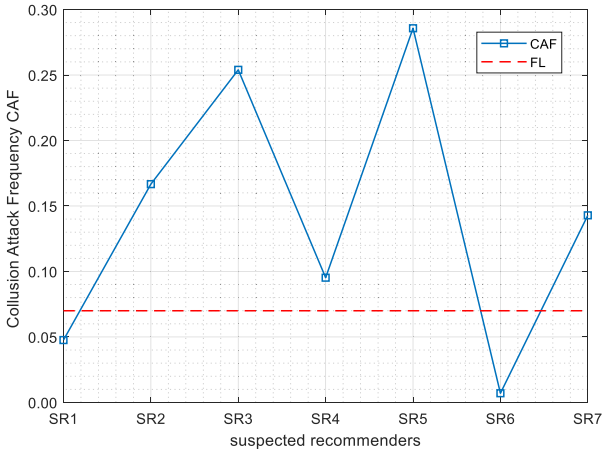

Fig. 6. Collusion attack frequency.

To measure the size of the collusion set (*CS*) where the malicious recommenders in a recommender's community must comprise a large proportion of all recommenders to attack and cause damage to the trust model, as we see in Fig. 7 the trust model will compute the attack scale (*AS*), which represents the size of attack scale for different collusion sets.

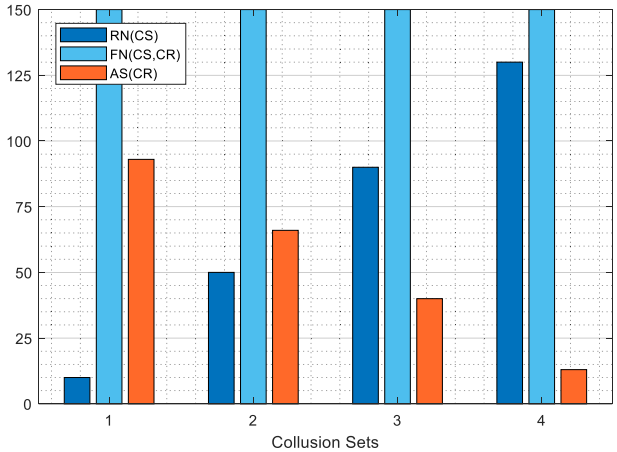

Fig. 7. Attack scale.(*AS*)

Fig. 8 shows the value of the attack target scale (*ATS*), which provides the malicious feedback rate from one collusion set (*CS*) for a specific consumer.

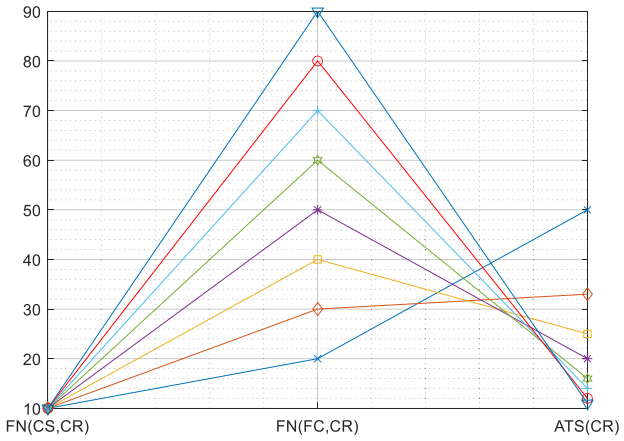

Fig. 8. Attack target scale.(*ATS*)

Table 1
Collusion attacks frequency.

$FN(SR_i, CR)$	6	21	32	12	36	1	18
$FN(SS, CR)$	126	126	126	126	126	126	126
$CAF(SR_i, CR)$	0.05	0.17	0.25	0.10	0.29	0.01	0.14

Fig. 9 shows the value of the collusion attack strength (*CAS*) by calculating the rate of all malicious feedback from different communities for a specific consumer.

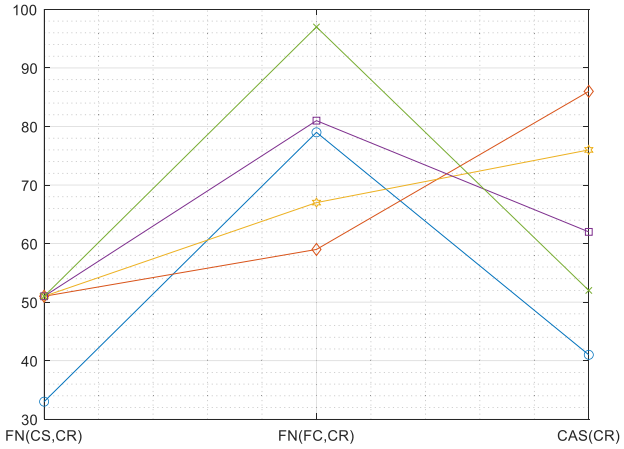

Fig. 9. Collusion attack strength.(*CAS*)

5. Discussion

Every model of trust management service is endangered by some security threats. These threats may either elevate the reputation of a certain unit with malicious intentions or entirely ruin it. In order to build a robust, secure, and accurate trust model system,

Table 2
Comparison of security and accuracy.

Addressed Metrics	[12]	[30]	[37]	[38]	[39]	OURS
Interaction Importance	✗	✓	✗	✗	✗	✓
On/off Attack	✗	✓	✓	✓	✗	✓
Trust Decline	✓	✓	✗	✗	✗	✓
Recommender importance	✗	✓	✗	✗	✗	✓
Collusion Attack	✓	✓	✓	✗	✓	✓
Collusion Attack Frequency	✗	✗	✗	✗	✓	✓
Sybil Attack	✓	✓	✓	✗	✗	✓

we focus on all reputation attacks of cloud computing by applying different criteria to avoid these attacks. Table 2 shows the comparison between our proposed TMS with those given in related works.

5.1. Future work

The issue of cloud service trust has attracted many researchers, but there are still many concerns that must be addressed. In future work, we will introduce additional criteria that increase the security of the trust model. In addition, we will seek other types of reputation attacks that threaten the security of the cloud computing environment and propose solutions to avoid these attacks.

6. Conclusion

Authorization issues concerning access to cloud computing storage are of serious concern when there are a large number of consumers using the big data of cloud computing for sensitive data. Such concerns can be properly attended to through the integration of control models with trust models as an improved security strategy for decentralized systems. We can draw the conclusion that cloud computing threats may either elevate the reputation of a unit with malicious intentions or entirely ruin it. This paper focuses on ensuring the security of cloud storage systems by building a trust model system to avoid reputation attacks of cloud services, effectively protecting cloud services through efficient detection of malicious as well as unbecoming behaviors by applying trust algorithms tailored to prevent on–off, collusion, and Sybil attacks, where each trust algorithm contains different criteria to avoid reputation attacks. Overall, our results demonstrate a robust ability to uphold the security of cloud service consumers. Their association with trust management services may involve highly sensitive data, so assurance of trust management services is crucial owing to cloud services' dynamic nature. Our solution offers reliable solutions to avoid reputation attacks and makes for very accurate consumer trust values by taking interaction importance into consideration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (D-066-611-1442). The authors, therefore, gratefully acknowledge DSR technical and financial support.

We would like to express our appreciation to the editor and reviewers for their time and effort in helping us to improve the quality of this manuscript.

References

[1] Noor TH, Sheng QZ, Alfazi A. Reputation Attacks Detection for Effective Trust Assessment Among Cloud Services. In: in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. p. 469–76.
[2] Mehraj S, Banday MT. Establishing a Zero Trust Strategy in Cloud Computing Environment. in *International Conference on Computer Communication and Informatics*, 2020.
[3] Varalakshmi P, Judgi T, Balaji D. "Trust Management Model Based on Malicious Filtered Feedback in Cloud. In: In International Conference on Data Science Analytics and Applications", in *International Conference on Data Science Analytics and Applications*. p. 178–87.
[4] S. Bhatt, R. Sandhu, and F. Patwa, "An Access Control Framework for Cloud-Enabled Wearable Internet of Things," in IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), pp. 213–233, Oct. 2017.
[5] Zhou L, Varadharajan V, Hitchens M. Integrating Trust with Cryptographic Role-based Access Control for Secure Cloud Data Storage. in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2013.
[6] Deng W, Zhou Z. A Flexible RBAC Model Based on Trust in Open System. Third Global Congress on Intelligent Systems 2012.
[7] Ismail R, Jøsang A. In: in *Proceedings of the 15th Bled Electronic Commerce Conference*. p. 2502–11.
[8] Chang W, Xu F, Dou J. A Trust and Unauthorized Operation Based RBAC (TUORBAC) Model. in *International Conference on Control Engineering and Communication Technology*, 2012.
[9] Tan Z, Tang Z, Li R, Sallam A, Yang L. Research on Trust-based Access Control Model in Cloud Computing. in *6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, 2011.
[10] L. Huang, Z. Xiong, and G. Wang, "A Trust-role Access Control Model Facing Cloud Computing," Proceedings of the 35th Chinese Control Conference, July 27-29, 2016.
[11] Li X, Du J. Adaptive and Attribute-based Trust Model for Service Level Agreement Guarantee in Cloud Computing. IET Inf Secur 2013;7 (1):39–50.
[12] Noor TH, Sheng QZ, Alfazi A. Detecting occasional reputation attacks on cloud services. In: in *International Conference on Web Engineering*. p. 416–23.
[13] M. Varsha and P. Patil, "A Survey on Authentication and Access Control for Cloud Computing using RBDAC Mechanism," International Journal of Innovative Research in Computer and Communication Engineering, 12125–12129, 2015.
[14] Lin G, Wang D, Bie Y, Lei M. MTBAC: A Mutual Trust Based Access Control Model in Cloud Computing. China Commun 2014;11(4):154–62.
[15] Zhu C, Nicanfar H, Leung VCM, Yang LT. An Authenticated Trust and Reputation Calculation and Management System for Cloud and Sensor Networks Integration. IEEE Trans Inf Forensics Secur 2014;10 (1):118–31.
[16] X. Li, H. Ma, F. Zhou, and X. Gui, "Service Operator-Aware Trust Scheme for Resource Matchmaking across Multiple Clouds," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 5, pp. 1419–1429, May. 2014.
[17] S. Chakraborty, and I. Ray, "TrustBAC: Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems," in SACMAT '06 Proceedings of the eleventh ACM symposium on Access control models and technologies, 2006, DOI:10.1145/1133058.1133067.
[18] T.H. Noor Z. Quan Z. Sheng Maamar, and Sherali, Zeadally. Managing trust in the cloud: State of the art and research challenges Computer 49 2016 34 45
[19] Zhao L, Liu S, Li J, Xu H, Zhao L. A Dynamic Access Control model Based on Trust. Information Application Technology; 2010.
[20] Li X, Ma H, Zhou F, Yao W. T-Broker: A Trust-Aware Service Brokering Scheme for Multiple Cloud Collaborative Services. IEEE Trans Inf Forensics Secur 2015;10(7):1402–15.
[21] Bhattasali T, Chaki R, Chaki N, Saeed K. An Adaptation of Context and Trust Aware Workflow Oriented Access Control for Remote Healthcare. Int J Software Eng Knowl Eng 2018;28(6):781–810.
[22] Marudhadevi D, Neelaya Dhatchayani V, Shankar Sriram VS. A Trust Evaluation Model for Cloud Computing Using Service Level Agreement. The Computer Journal, Nov. 2014;58:2225–32.
[23] Tsai WT, Zhong P, Bai X, Elston J. Role-Based Trust Model for Community of Interest. in *IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 2009.
[24] Zhang L, Yin N, Liu J, Wang R. Collusion detector based on G-N algorithm for trust model. J Syst Eng Electron 2016;27(4):926–35.
[25] V. Oleshchuk, "Trust-Aware RBAC," in Lecture Notes in Computer Science, Springer, pp. 97–107, 2012.
[26] Fan Y, Zhang Y. Trusted Access Control Model Based on Role and Task in Cloud Computing. in *7th International Conference on Information Technology in Medicine and Education*, 2012.
[27] Noor TH, Sheng QZ, Bouguettaya A. Trust management in cloud services. Springer; 2014.
[28] W. Tong , J. Liang, L. Lu, and X. Jin, "Intrusion detection scheme based node trust value in WSNs," in Systems Engineering and Electronics, pp. 1644–1649, 2015.
[29] Labraoui Nabila, Gueroui Mourad, Sekhri Larbi. On-off Attacks mitigation against trust systems in wireless sensor networks. In: in IFIP International Conference on Computer Science and its Applications. p. 406–15.
[30] Ghafoorian M, Abbasinezhad-Mood D, Shakeri H. A Thorough Trust and Reputation Based RBAC Model for Secure Data Storage in the Cloud. IEEE Trans Parallel Distrib Syst Apr. 2019;30(4).
[31] Zhang P, Kong Y, Zhou M. A Domain Partition-Based Trust Model for Unreliable Clouds. IEEE Trans Inf Forensics Secur Sept. 2018;13(9):2167–78.
[32] F. N. Nwebonyi, R. Martins, and M. E. Correia, "Reputation based approach for improved fairness and robustness in P2P protocols," in Peer-to-Peer Networking and Applications, pp. 951–968, 2019.
[33] Eric Chang, "General Attacks and Approaches in Cloud-Scale Networks," in IEEE International Conference on Computer Communications, 2019.
[34] Hassan H, El-Desouky AI, Ibrahim A, El-Kenawy EM. "Enhanced QoS-Based Model for Trust Assessment in Cloud Computing Environment", in *IEEE*. Access 2020.

[35] S. Mahajan, S. Mahajan, S. Jadhav, and Sangita K., "Trust Management in E-commerce Websites," in International Research Journal of Engineering and Technology (IRJET), pp. 2934–2936, 2017.
[36] Noor TH, Sheng QZ, Yao L, Dustdar S, Ngu AHH. CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services. IEEE Trans Parallel Distrib Syst 2015;27(2):367–80.
[37] Barsoum A, Hasan A. Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems. IEEE Trans Parallel Distrib Syst 2012;24 (12):2375–85.
[38] C. Uikey and D. S. Bhilare, "TrustRBAC: Trust Role Based Access Control Model in Multi-domain Cloud Environments," in International Conference on Information, Communication, Instrumentation and Control (ICICIC), p. 978-1-5090-6314-7, 2017.
[39] Fortino G, Fotia L, Messina F, Rosaci D, Sarné GML. "Trust and Reputation in the Internet of Things: State-of-the-Art and Research Challenges", in *IEEE*. Access 2020.