Article

Building a Reputation Attack Detector for Effective Trust Evaluation in a Cloud Services Environment

Salah T. Alshammari * and Khalid Alsubhi

Department of Computer Science, College of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; kalsubhi@kau.edu.sa
* Correspondence: salshammari0042@stu.kau.edu.sa; Tel.: +966-500-708-076

Abstract: Cloud computing is a widely used technology that has changed the way people and organizations store and access information. This technology is versatile, and extensive amounts of data can be stored in the cloud. Businesses can access various services over the cloud without having to install applications. However, cloud computing services are provided over a public domain, which means that both trusted and non-trusted users can access the services. Although there are a number of advantages to cloud computing services, especially for business owners, various challenges are posed in terms of the privacy and security of information and online services. A threat that is widely faced in the cloud environment is the on/off attack, in which entities exhibit proper behavior for a given time period to develop a positive reputation and gather trust, after which they exhibit deception. Another threat often faced by trust management services is a collusion attack, which is also known as collusive malicious feedback behavior. This is carried out when a group of people work together to make false recommendations with the intention of damaging the reputation of another party, which is referred to as a slandering attack, or to enhance their own reputation, which is referred to as a self-promoting attack. In this paper, a viable solution is provided with the given trust model for preventing these attacks. This method works by providing effective security to cloud services by identifying malicious and inappropriate behaviors through the application of trust algorithms that can identify on/off attacks and collusion attacks by applying different security criteria. Finally, the results show that the proposed trust model system can provide high security by decreasing security risk and improving the quality of decisions of data owners and cloud operators.

Keywords: trust model; reputation attacks; cloud computing; on/off attack; collusion attack; cloud services

1. Introduction

Many threats are faced by users of cloud computing services, including trust and reputation attacks. These threats are experienced due to the extremely dynamic, non-transparent, and distributed nature of cloud computing services [1,2]. Cloud service providers and customers face significant challenges in preserving and handling trust in the cloud system [3]. Threats also emerge because the provision of cloud computing services is in the public domain, which means that many users have access to it [4]. There are risks from the actions of malicious users toward other cloud service consumers, who often give feedback about their experiences.

Noor, Sheng, and Alfazi [1] asserted that the feedback of service providers was an effective means of obtaining information that helps in examining the trustworthiness of cloud service customers. Varalakshmi, Judgi, and Balagi [5] also noted that service provider feedback plays a vital part in generating trust for a cloud-based service. Cloud computing providers and consumers use trust management systems and detection systems for reputation attacks to provide improved protection and security of online data.

Although trust management and reputation attack detection systems are available, cloud computing systems continue to face targeted attacks from different sources [6,7].
Research shows that there are shortcomings in a number of detection strategies that aim to prevent on/off and collusion attacks. Hence, this study provides vital information regarding the way reputation attack detection mechanisms can be used for successful detection in cloud services. In addition, it will address the on/off and collusion attacks that decrease trust assessment in cloud computing. A major contribution is made by this study to the existing literature regarding how the security and privacy of cloud services can be improved by providers and users of such services, and can provide an enhanced experience for users on the whole.

Another major challenge emerges from authorization concerns regarding access to cloud computing. The issue is concerning because of the significant number of users and the highly sensitive big data of cloud computing [8,9]. In most cases, this has led to the application of access controls within cloud computing server application platforms. However, it has been found that access controls related to distributed systems are not reliable [10]. This is mainly due to the dynamic and complicated nature of consumers and the ineffectiveness of determining their identity beforehand [11]. To adequately handle these concerns, a better option for decentralized systems is to integrate control models with trust models [12]. This solution has been attained following various attempts by developers to create new trust models that have the ability to solve the most complex and sophisticated issues linked to authorization [13,14]. Various proposals have been presented for integrated trust models with access controls that face significant threats from attacks [15,16]. This paper aims to design a trust model system for curtailing reputation attacks on cloud services and providing high security to cloud storage systems. A major security challenge experienced by reputation systems is the on/off attacks.

1.1. Problem Statement

The on/off attack is a widely encountered threat in cloud environments. Here, a few entities show proper behavior for some time to create a positive reputation. Once they have generated trust, they start their deception. Malicious bodies win the trust of the trust management system (TMS) by exhibiting good behavior during interactions that have small impacts. However, when they come across a suitable opportunity for major interactions, their malicious intentions become evident. For example, sellers on eBay carry out various small transactions to generate a positive reputation, and then after gaining trust, cheat buyers on a more expensive product. Since these entities suddenly change their transactional behavior, it becomes difficult for other entities to sufficiently diminish the attacker’s reputation. This also consists of oscillatory transaction behavior, where an entity keeps shifting from honest to dishonest behavior so that the attacker’s reputation cannot be updated in a timely manner.

Collusion attacks are threats that emerge when a group or groups of malicious entities make attempts to destabilize the system. In most cases, when different malicious entities act together, there is a greater danger to the reliability of a reputation system than when every entity exhibits malicious behavior separately. Some specific instances of collusion attacks are described below.

Collusive slandering attack: also known as a collusive badmouthing attack, malicious users work together to propagate negative reviews about an honest user to significantly decrease their reputation. They also try to enhance their own reputations by giving positive reviews of each other.

Collusive reducing reputation: the reputation of honest users can be reduced through coordinated malicious nodes that criticize only some sections of the entities they deal with. They do so to generate conflicting views regarding the transactional behavior of the victims and the reputation of honest entities recommending the victims. There is very little effect on their own reputation, since they give honest feedback about other entities.

Collusive self-promoting attack: this kind of attack is also referred to as a ballot stuffing attack, in which negative behavior is exhibited by all entities of a group of malicious entities, but they give positive feedback about each other. This attack is carried out in different
ways. In one instance, dishonest behavior is exhibited by just one entity of the collusive group, while positive reviews are given by other entities. In another instance, unfavorable behavior is exhibited by a single entity some of the time to avoid being identified, whereas other members of this group give positive reviews for it. Often, the term “ballot stuffing” is used to refer to an attack of this category. In this kind of attack, ratings are given for fake transactions. For instance, a seller can collude in an online auction system with a group of buyers to achieve high ratings that are not reflective of actual transactions. This will increase their reputation, leading to a greater number of orders for that seller from other buyers and allowing them to sell at a greater price than justified.

1.2. Contributions

The objective of this research is to ensure that the cloud storage system is highly secure by developing a trust model system that inhibits reputation attacks of cloud services. On/off attacks are carried out by entities that present themselves as good nodes on an online platform; however, once they have acquired the trust of the system, they become malicious nodes. Nodes that carry out the on/off attacks may give false information or feedback to cause damage to the system’s reputation. Another type of security issue that threatens reputation systems are the collusion attacks. These attacks are threats that emerge when a group or groups of malicious entities make attempts to destabilize the system. In the majority of cases, when different malicious entities act together, there is greater danger to the reliability of a reputation system as when every entity exhibits malicious behavior separately. A solution for trust model systems for preventing on/off and collusion attacks will be presented in this study. In this regard, we will put forward strategies that should be considered when developing the trust evaluation process. This paper will try to determine the solution that is most appropriate for trust issues regarding access control approaches and will then present trust models that can be used to increase information security in distributed storage frameworks that use cryptographic access control techniques. It was found during the investigation that accurate results should be offered by a trust model while assessing trustworthiness, which is what our plan for the recommended trust-based distribution storage framework is based on. In the plan, trust prototypes can be organized into a framework that uses the cryptographic access control approach. To ensure its effectiveness, this paper proposes a trust model that offers the following:

✓ Ensuring that the greatest security is provided to customers of cloud services because they may be dealing with extremely sensitive data when using trust management services;
✓ Securing cloud services by effectively detecting malicious and inappropriate behaviors through the use of trust algorithms that can identify on/off and collusion attacks;
✓ Making sure that trust management service availability is adequate for the dynamic nature of cloud services;
✓ Providing dependable solutions to avoid reputation attacks and increase the precision of customers’ trust values by considering the significance of interaction.

2. Related Works

An increase has been experienced in the number of methods employed for examining and handling trust for online services. Noor, Sheng, and Alfazi [1] asserted that these methods were developed after taking into account the feedback of customers of cloud services. The drawback of this study, however, is that the methods do not stress the occasional and periodic reputation attacks that often hamper the privacy and security of cloud computing.

It is due to the dynamic nature of cloud computing that there has been very little focus on these periodic and occasional reputation attacks. The fact that multiple accounts may be held by a single user to access a single service makes the issue even more complex. The Noor, Sheng, and Alfazi [17] study offers vital information about the efficiency of occasional attack detection models in identifying occasional and periodic reputation attacks,
but the emphasis of the study was not on particular attacks, such as the on/off and collusion attacks.

A similar study was carried out by Labraoui, Gueroui, and Sekhri [18] in 2015 to examine whether trust and reputation networks were successful in preventing reputation attacks. They presented the on/off trust mitigation for trust systems employed in wireless sensor networks. The mechanism of the mitigation system punishes those with a history of misbehavior on every network node. The trust value of every network node is influenced by these penalties, so it is considered successful in preventing misbehavior. The advantage of this study is that it offers vital information about the on/off trust approach. Trust management in cloud computing was also examined by Noor, Sheng, and Bouguettaya [19] in 2014, but they did not include how on/off attacks could be prevented by trust management models. Tong, Liang, LU, and JIN [20] suggested in 2015 that the trust model simply took into account score value similarity and the collusion size score, but did not take into account the impact of feedback time.

Ghafoorian, Abbasinezhad-Mood, and Shakeri [21] carried out a study in 2018 to examine how the trust- and reputation-based RBAC model was used to provide security for data storage in the cloud. They determined that the RBAC model successfully dealt with security risks to the reputation and trust of cloud-based systems.

Other approaches have been examined to manage the trust and reputation of cloud systems. A study was carried out by Nwebonyi, Martins, and Correia [22] in 2019 to evaluate the effectiveness of various models that prevented these trust and reputation risks. However, the focus of these studies was essentially on the privacy and security of the system as a whole, and they did not deal with particular attacks, such as on/off and collusion attacks.

Many reputation-based models were developed in most of the earlier trust models for the detection of collusion attacks through different methods. A trust model defenseless against a collusion attack was presented in [3], which worked by computing the feedback density of all recommenders. Another was developed in [2] by determining the credibility of feedback and changing it in the system with alternative recommenders. An identity-based data storage mechanism was developed by the authors in [21], in which collusion attacks were inhibited, and intra-domain and inter-domain queries were supported. Mahdi et al. [22] suggested that to decrease the threat of this attack, it could be prevented by considering the least number of recommenders that should be involved in calculating the recommendation trust; otherwise, the RBAC system can easily use virtual recommenders for the compilation process of the recommendation trust.

3. Design and Methodology

The design methodology for this study is a literature review, which is a systematic research method through which findings from previous studies are collected and integrated [23–25]. When a literature review is carried out rigorously, adhering to all the rules and conditions of assessing the quality of research evidence, a sound basis is created for developing knowledge, as well as theory. There is a lot of evidence related to attacks carried out on cloud-based services and the way these attacks can be avoided [26,27]. However, this evidence is fragmented. The focus of this study will be on two types of attacks, which are similar in various ways to other attacks [28,29], despite their differences with respect to their features and how they are carried out in the cloud environment [30].

3.1. On/Off Attack

In this type of attack, malicious behavior is exhibited by the user of the cloud system in a short time frame. They begin by exhibiting good conduct so that they can deceive the trust system and gain a good reputation. This kind of attack mostly occurs when proper behavior is initially shown by a malicious cloud service user for some time to attain a good reputation and gain the trust of other users. The user then starts taking advantage of this trust. Malicious users win the trust of the TMS in most cases by behaving properly in
minor interactions. However, when they get an opportunity for a major interaction, their malicious intent becomes evident.

The opportunistic but malicious behavior of various nodes is characteristic of attacks that are made through direct trust, including the trust of the system. There is a shift in the good and bad behavior of switches, which creates a disguise, and the node is considered trustworthy despite its inappropriate behavior [4]. For instance, malicious behavior can be exhibited by a node that is considered trustworthy on an e-commerce website. This behavior may not be identified because the node was initially considered trustworthy by the system. The interaction trust (IT) will initially be computed by the trust model system, which offers an accurate result for the trust value of every cloud service consumer (CR) by calculating the service provider’s (SP’s) interaction importance (II). The feedback (F) with respect to an interaction is presented as a percentage. Equation (1) is used to compute the IT.

\[
\text{IT}(\text{CR}) = \sum_{i=1}^{n} \left( \frac{\alpha_{i}^{t}(\text{CR}) + P_{\text{CR}}}{(\alpha_{i}^{t}(\text{CR}) + P_{\text{CR}}) + (\beta_{i}^{t}(\text{CR}) + N_{\text{CR}})} \right)
\]

\[
\sum_{i=1}^{n} \alpha_{i}^{t}(\text{CR}) = \left( \begin{array}{cccc} \alpha_{1}^{t} & \alpha_{2}^{t} & \cdots & \alpha_{n-1}^{t} & \alpha_{n}^{t} \\ V_{1,1} & V_{1,2} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n-1} & V_{n,n} \end{array} \right)
\]

\[
\sum_{i=1}^{n} \beta_{i}^{t}(\text{CR}) = \left( \begin{array}{cccc} \beta_{1}^{t} & \beta_{2}^{t} & \cdots & \beta_{n-1}^{t} & \beta_{n}^{t} \\ V_{1,1} & V_{1,2} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n-1} & V_{n,n} \end{array} \right)
\]

\[
P_{\text{CR}} = \frac{\alpha_{n}^{t}(\text{CR}) \times \text{II}}{\text{NF}}
\]

\[
N_{\text{CR}} = \frac{\beta_{n}^{t}(\text{CR}) \times \text{II}}{\text{NF}}
\]

Here, positive feedback in a given time is shown by \(\alpha_{i}^{t}\); negative feedback in a given time is shown by \(\beta_{i}^{t}\); P denotes the value of the new positive recommender feedback, N represents the value of the new negative recommender feedback; the Interaction Importance value is shown with II, and the feedback numbers are denoted by NF.

The risk of on/off attacks (\(O^{2}\)) can be prevented by incorporating the penalty for the on/off attack (\(P^{O^{2}}\)). This penalty is fixed from one to n, where one refers to no risk from this user and n represents high interaction importance multiplied by the danger rate (DR). \(P^{O^{2}}\) for any customer is calculated with the trust model using a novel process in which PC\(^{O^{2}}\) is the smallest value of the greatest interactions.

\[
\begin{cases} 
\text{if } \text{II} \geq \text{PC}^{O^{2}} \text{ and } \beta_{n}^{t} < \text{II} \text{ then } & P^{O^{2}} = \text{II} \times \text{DR} \\
\text{else} & P^{O^{2}} = 1
\end{cases}
\]

\[
\text{IT}(\text{CR}) = \sum_{i=1}^{n-1} \left( \frac{\alpha_{i}^{t}(\text{CR}) + P_{\text{CR}}}{(\alpha_{i}^{t}(\text{CR}) + P_{\text{CR}}) + (\beta_{i}^{t}(\text{CR}) + N_{\text{CR}} \times P^{O^{2}})} \right)
\]

The penalty of trust decline (\(P^{TD}\)) should be added to prevent the risk of trust decline (\(TD\)), where the \(P^{TD}\) ranges from one to n, in a way that one signifies no risk from this...
customer role and \( \backslash \) signifies high risk, \( P_{CTD} \) refers to the curve of a penalty of trust decline and its integer greater than one, \( L^{II} \) denotes the limit of low interaction.

\[
\text{IT}(\text{CR}) = \frac{\sum_{i=1}^{n-1} \alpha_i^{CR} + P_{CR}}{\left(\alpha_i^{CR} + P_{CR}\right) + \left(\beta_i^{CR} + N_{CR} \times P^{O2} \times P_{TD}\right)} \tag{3}
\]

\[
\begin{cases} 
\text{if } \beta_n^{CR} < L^{II} & \text{then } P_{TD} = P_{CTD} \\
\text{else } & P_{TD} = 1 
\end{cases}
\]

\( PC_{TD} \) is used to find the value of penalty of trust decline \( (P_{TD}) \), and to prevent the attacks, the danger rate \( R \) and the penalty for on/off attack \( (P^{O2}) \) are used. Algorithm 1 is given below.

**Algorithm 1 On/off Attack Algorithm**

```
Input : F, II;
Output : Consumer Trust Value;
1: procedure Interaction Trust
2: \( \alpha_n^{CR}(CR) = F \)
3: \( \beta_n^{CR}(CR) = F - 1 \)
4: \( P_{CR} \leftarrow \left(\alpha_n^{CR}(CR) \times II\right) / NF \)
5: \( N_{CR} \leftarrow \left(\beta_n^{CR}(CR) \times II\right) / NF \)
6: if II \( \geq \) PC\(^{O2}\) and \( \beta_n^{CR} < II \) then \( P^{O2} = II \times DR \)
7: end if
8: if \( \beta_n^{CR} < II \) then \( P_{TD} = P_{CTD} \)
9: else \( P_{TD} = 1 \)
10: end if
11: for \( i = 1, i \leq n - 1; \)
12: \( \text{IT}(CR) \leftarrow \left(\alpha_i^{CR}(CR) + P_{CR}\right) / \left(\left(\alpha_i^{CR}(CR) + P_{CR}\right) + \left(\beta_i^{CR}(CR) + N_{CR} \times P^{O2} \times P_{TD}\right)\right) \)
13: end for
14: end procedure
```

### 3.2. Collusion Attack

Collusion attacks are also studied in this paper. In the literature, the word collusion refers to an agreement reached by a collection of entities to develop biased or impractical recommendations [5]. This means that the attack takes place when a group of people work together to give false recommendations with the intention of damaging another entity’s reputation. This is referred to as the slandering attack. They may also perform self-promoting attacks to enhance their own reputation [6]. The focus of this study will be on a single type of attack, which is similar to some extent to other kinds of attacks. However, they are distinct with respect to their attributes and the way they are carried out on the cloud computing platform.

In a collusion attack, malicious behavior is exhibited that involves giving fake feedback from a single node, with the aim of enhancing or reducing product ratings on an e-commerce website [25]. It is possible for the behavior to be non-collusive, with multiple misleading pieces of feedback given by the nodes to carry out self-promotion or slander another entity. The TMS will not be effective when malicious users make up more than 50% of the trust model system. The accuracy of the recommendation trust values is also threatened by the attack. There are two main types of collusion attacks: a self-promoting attack, in which malicious recommenders collude to enhance the trust value of a particular user in the TMS, and a slandering attack, in which malicious recommenders work together to reduce the trust value of a particular user in the TMS.
A novel solution is presented in this study to avoid collusion attacks; it involves computing three criteria that indicate distinct factors. Malicious recommendation detection (MRD) is the foremost criterion that identifies suspicious recommenders’ groups by determining the probability that a group will be the collusion recommender’s group. In this criterion, the time range of the collusion attack is calculated using the trust model to identify the time range of all attacks that have been carried out within a short period of time. If malicious recommenders have made any attacks on a particular user, there will be a very small time range of these attacks. The second criterion is then computed by the trust model: malicious recommenders’ behavior (MRB). This indicates the similarity of malicious recommenders’ behaviors, which are more similar when a particular user is attacked. The following Equation (4) is used to determine malicious recommendation detection (MRD) and malicious recommenders’ behavior (MRB).

\[
\text{for } i = 1 \text{ to } n - 1 \begin{cases} 
\text{if } T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR \\
\text{and } V(F_n, CR)^{FS} \geq V(F_i, CR)^{FS} \text{ and } V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \leq \max VR \\
\text{then move } (F_i, CR)^{FS} \text{ form } FS \text{ to } SS \\
\text{elseif } T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR \\
\text{and } V(F_n, CR)^{FS} < V(F_i, CR)^{FS} \text{ and } V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \geq \min VR \\
\text{then move } (F_i, CR)^{FS} \text{ form } FS \text{ to } SS \\
\end{cases} \\
\text{endif} \\
\text{endfor}
\]

\[
\text{for } i = 1 \text{ to } n \begin{cases} 
\text{if } T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR \\
\text{and } V(F_n, CR)^{FS} \geq V(F_i, CR)^{SS} \text{ and } V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \leq \max VR \\
\text{then move } (F_i, CR)^{FS} \text{ form } FS \text{ to } SS \\
\text{elseif } T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR \\
\text{and } V(F_n, CR)^{FS} < V(F_i, CR)^{SS} \text{ and } V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \geq \min VR \\
\text{then move } (F_i, CR)^{FS} \text{ form } FS \text{ to } SS \\
\end{cases} \\
\text{endif} \\
\text{endfor}
\]

\[
\text{where} \\
\begin{align*}
TR &= T(F_n, CR)^{FS} \times TC \\
\max VR &= V(F_n, CR)^{FS} \times VC \\
\min VR &= -V(F_n, CR)^{FS} \times VC
\end{align*}
\]

The time and value of all feedback in the feedback set (FS) provided to a particular user are compared in the TMS. In the first comparison, the time of the last feedback \(T(F_n, CR)^{FS}\) and the time of all \(T(F_i, CR)^{FS}\) feedback apart from the last is compared by the TMS. Following that, the value of the last feedback is compared in the TMS with the value of all \(V(F_i, CR)^{FS}\) feedback apart from the last. The trust model performs a comparison between the time \(T(F_n, CR)^{FS}\) and the value \(V(F_n, CR)^{FS}\) of the last feedback in the feedback set (FS) and the time \(T(F_i, CR)^{SS}\) and value \(V(F_i, CR)^{SS}\) of all feedback in the suspected set (SS). This is how the TMS keeps all suspected feedback in the suspected set (SS). The two parameters to identify the range of feedback time and feedback value can be determined through the two parameters TC and VC.

The third criterion, which is the collusion attack frequency (CAF) for each recommender that has feedback in the suspected set \(SS = \{SF_1, SF_2, \ldots, SF_n\}\), is determined...
to identify the malicious recommendations, also called the collusion feedback. Here, the strength of the attack is higher when the frequency of attacks is greater.

\[ CAF(SR_i, CR) = \frac{FN(SR_i, CR)}{FN(SS, CR)} \tag{5} \]

\[
\begin{cases} 
\text{if } CAF(SR_i, CR) \geq FL \text{ then move } SF(SR, CR) \text{ to } CS \\
\text{else } \text{move } SF(SR, CR) \text{ to } FS 
\end{cases}
\]

The trust model initially calculates the collusion attack frequency (CAF), where \( FN(SR, CR) \) signifies the number of feedback items that are provided by a particular recommender to a particular user in the suspected set (SS). \( FN(SS, CR) \) refers to the number of feedback items in the suspected set (SS) that are provided to the same consumer. When the collusion attack frequency (CAF) is more than the feedback limit (FL), the trust model shifts the suspected feedback to the collusion set (CS). However, if this is not the case, the suspected feedback SF(SR, CR) is shifted by the trust model from a specific recommender to a particular consumer in the feedback set (FS).

The trust model determines the size of the collusion set (CS) to determine the attack scale (AS) for a given consumer, where the malicious recommenders in a recommender’s community account for a major percentage of all recommenders who attack the trust model and cause harm to it. The following Equation (6) is used to determine the collusion attack scale (AS):

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

Here, RN(CS) refers to the number of malicious recommenders within the collusion community and FN(CS, CR) refers to the number of malicious feedback items for all consumers within the collusion set (CS).

The trust model then determines the attack target scale (ATS), which gives a malicious feedback rate for a particular user. The following Equation (7) is used to determine the attack target scale (ATS):

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

Here, the FN(CS, CR) refers to the feedback number obtained from malicious recommenders. FN(AFS, CR) is the feedback number that all feedback sets have given, in which the set of malicious recommenders belongs to all recommenders who have evaluated a particular consumer.

Finally, the strength of all collusions for a given user can be determined by calculating the collusion attack strength (CAS) using the trust model. For this, the value of all attack target scales (ATS) is considered, after which the collusion attack strength (CAS) is computed as follows (8):

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

The collusion attack algorithm given below will be employed by the trust model to identify collusion groups (Algorithm 2).
Algorithm 2 Collusion Attack Algorithm

Input: FS, TC, VC, FL;
Output: SS, CS, AS(CR), ATS(CR), CAS(CR);

1: procedure collusion attack
2: TR ← T(F_n, CR)^FS × TC
3: maxVR ← V(F_n, CR)^FS × VC
4: minVR ← − V(F_n, CR)^FS × VC
5: for i = 1 to n − 1
   if T(F_n, CR)^FS − T(F_i, CR)^FS ≤ TR
      and V(F_n, CR)^FS ≥ V(F_i, CR)^FS and V(F_n, CR)^FS − V(F_i, CR)^FS ≤ maxVR
         then move (F_i, CR)^FS from FS to SS
   elseif T(F_n, CR)^FS − T(F_i, CR)^FS ≤ TR
      and V(F_n, CR)^FS < V(F_i, CR)^FS and V(F_n, CR)^FS − V(F_i, CR)^FS ≥ minVR
         then move (F_i, CR)^FS from FS to SS
   endelseif
6: endfor
7: for i = 1 to n
   if T(F_n, CR)^FS − T(F_i, CR)^SS ≤ TR
      and V(F_n, CR)^FS ≥ V(F_i, CR)^SS and V(F_n, CR)^FS − V(F_i, CR)^SS ≤ maxVR
         then move (F_i, CR)^FS from FS to SS
   elseif T(F_n, CR)^FS − T(F_i, CR)^SS ≤ TR
      and V(F_n, CR)^FS < V(F_i, CR)^SS and V(F_n, CR)^FS − V(F_i, CR)^SS ≥ minVR
         then move (F_i, CR)^FS form FS to SS
   endelseif
8: endfor
9: end
10: end procedure

The example of the collusion attack given below can be evaluated. This example demonstrates how the trust value of the roles is influenced by the collusion attack algorithm. Here, we assume that the matrix given below consists of all feedback times.

\[
\sum_{i=1}^{n} \text{FS} \left( \begin{array}{cccc}
T(F_1) & \cdots & T(F_{n-1}) & T(F_n) \\
V_{1,1} & \cdots & V_{1,n-1} & V_{1,n} \\
V_{2,1} & \cdots & V_{2,n-1} & V_{2,n} \\
\vdots & \ddots & \vdots & \vdots \\
V_{n,1} & \cdots & V_{n,n-1} & V_{n,n} \\
\end{array} \right)
\]

\[
\sum_{i=1}^{n} \text{FS} \left( \begin{array}{cccc}
T(F_1) & \cdots & T(F_{n-1}) & T(F_n) \\
04:22 & \cdots & 05:09 & 14:22 \\
04:50 & \cdots & 05:12 & 14:50 \\
\vdots & \ddots & \vdots & \vdots \\
01:30 & \cdots & 20:01 & 22:30 \\
\end{array} \right)
\]
The different feedback times are compared by the algorithm, beginning with the final feedback time $V_{n,n}$. We make the assumption that the time period is two hours and all the feedback is obtained on the same day. Six feedback items are found in this time period, and their values are compared by the trust model. The following feedback values are assumed:

$$V(F_1) \cdots V(F_{n-1}) V(F_n)$$

$$\sum_{i=1}^{n} FS(CR) = \begin{pmatrix}
90\% & \cdots & 99\% & 98\% \\
92\% & \cdots & 95\% & 97\% \\
\vdots & \vdots & \vdots & \vdots \\
61\% & \cdots & 99\% & 82\%
\end{pmatrix}$$

Here, the system shifts the six feedback items to the suspected set (SS); the collusion attack frequency for every recommender that has feedback in the suspected set (SS) will then be determined by the trust model.

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

Using the condition given below, when the feedback limit (FL) = 10%, these feedback items will be transferred by the trust model to the collusion set (CS); if this is not the case, then they will be shifted to the FS.

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

4. Proposed Framework Architecture

The various elements of the trust model are analyzed in this section. The part played by each component in making sure the system functions effectively is also presented. The proposed system architecture is shown in Figure 1.

![Figure 1. Trust model system architecture.](image-url)
In the proposed design, TMS will be the element that examines the degree to which cloud service customers rely on cloud service providers. It is responsible for offering cloud services, with the expectation that the quality cloud service providers have promised is actually offered. There are different subsections of the TMS, each of which is assigned distinct tasks that ensure that the data available in the cloud storage system are secure. The rules identified within the level of trust are examined based on the feedback given by service providers [19]. It is important to determine the identity of the user and monitor the activities carried out to make it easier to track unauthorized customers or attackers and to present evidence of any leaked data. The registered and unregistered customers were updated using the TMS. In addition, all activities carried out by customers are identified. This monitors the authorization of all those adding feedback into the system. The TMS can recognize invalid feedback and remove it from the system.

4.1. Trust Management System (TMS)

Different layers are added to the trust model to increase the effectiveness of the overall system. There are different subsections in the TMS, which are described in the following section.

Central Repository: this functions as the interaction store. It stores all kinds of trust records and interaction histories created by interacting tasks and roles for subsequent use by the decision engine trust for assessing task and role values. The central repository cannot access elements that are not present in the TMS.

Role Behavior Analyzer: this is the component that analyzes functions and roles related to the smallest levels of trust regulations with respect to shared resources. It evaluates those rules that have been identified in the level of trust based on the feedback given by the service providers in the central repository [19]. The roles are linked by the role behavior analyzer to obtain information about them and to identify any leakage that occurs. It is imperative to determine the user’s identity and monitor all actions carried out by them so that unauthorized customers or attackers can easily be tracked, and evidence for any kind of data leakage can be presented. The accounts of registered and unregistered customers are also updated by the role behavior analyzer, and all events carried out by a customer are identified.

Task Behavior Analyzer: It is the responsibility of the task behavior analyzer to evaluate tasks and functions with respect to minimum trust level laws when accessing shared resources. The tasks identified within the trust level are analyzed in terms of the feedback of owners by calculating the trust value, and this value is then stored in the central repository. It obtains information from the channels, of which there are two here—reports from tasks regarding data leakage and reports from the role behavior analyzer—to determine the histories of customers with respect to the stored data. Customers can be identified by the task behavior analyzer, and the tasks performed should be tracked. This makes it easier to track attackers or unauthorized customers and to present proof of data leakage if it has occurred. Registered customer accounts are updated, and it is determined whether the incident was carried out by a customer account.

Feedback Collector: it is the responsibility of the feedback collector to manage feedback from the owners of the service to the repository headquarters before its automatic allocation. The user’s trustworthiness is shown by the feedback given on roles and tasks. To ensure security, the collector of tasks and role feedback secures its integrity. The authorization of the one uploading feedback into the system is ensured by this component. It has the ability to recognize invalid feedback and eliminate it from the system. In addition, information is acquired by the collector of role feedback regarding the data assignments of tasks and roles before it is uploaded to the central repository.

Trust Decision Engine: this section analyzes and identifies the value of trust of the data owners, the roles, and the values of the entity. It obtains all kinds of information regarding the interaction histories that are found in the repository center and the trust values of a specific customer before determining the kind of response the system should give.
4.2. Proposed Method

- The system is initiated by the administrator, who also establishes the system’s hierarchy of roles and tasks. Channel 1 makes it easier to upload the system’s created parameters for roles and tasks to the cloud;
- When consumers want cloud data access rights, they first submit an access request through Channel 2 based on their tasks and roles;
- If the consumer request is accepted, Channel 5 is used by the role entity to transfer the request to the task’s entity. The cloud responds by providing consumers with a cryptographic task-role based access control (T-RBAC) plan;
- Encryption and uploading of data performed through Channel 3 can only be given by the owner if they feel the role or task can be trusted. The owner also informs the feedback collector of the consumers’ identities during this stage;
- When an owner finds leakage of their data due to an untrustworthy consumer, they should report this role or task to the feedback collector through Channel 14;
- When an approved owner provides new feedback, the feedback collector sends it to the centralized database through Channel 11 to archive each confidence report and interaction record produced when the roles and tasks interact;
- The interaction logs are then stored in a centralized database, which is used by the trust decision engine to determine the trust value of roles and tasks using Channel 10;
- At any point, the roles’ entity may approve trust assessments for the roles from the TMS, whereupon the roles’ entity responds to the trust management system through Channel 13 to receive feedback from the trust decision engine;
- The role entity reviews the role parameters that decide a consumer’s cloud role membership until the feedback is obtained from the trust decision engine, and any malicious consumer membership is terminated;
- When an owner provides negative feedback regarding a role as a result of data leakage, the role entity transfers the leakage details to the role behavior analyzer through Channel 4;
- At any point, the tasks’ entity may approve trust assessments for the tasks from the TMS, whereupon the tasks entity responds to the trust management system through Channel 12 to receive feedback from the trust decision engine;
- The task entity reviews the parameters of the task that decide a consumer’s cloud task membership until the feedback is obtained from the trust decision engine, and any malicious consumer membership is terminated;
- When an owner provides negative feedback regarding a task as a result of data leakage, the task entity transfers the leakage details to the task behavior analyzer through Channel 7;
- The analyzers then use Channels 6 and 8 to keep updating the trust details for the roles and tasks in the centralized database;
- If an owner requests that their data be uploaded and encrypted in the cloud, the TMS performs a trust assessment. Once the TMS receives the request, it follows up with the owners through Channel 9;
- The trust decision engine provides the owners with details of the trust assessment for their roles and tasks. The data owners then determine whether or not to allow consumers access to their services based on the results.

5. Simulation Results

To check the results of our system, we built a C#.net Windows Forms application, and used big data to check all the criteria applied to avoid these attacks. Experiments were used to determine the trust model’s capability of resisting reputation attacks. The penalties for the on/off attack were determined by the TMS based on two conditions: whether the interaction importance was more than or equal to the danger rate (\( DR \)), and whether the feedback (\( F \)) of the recommender was less than the interaction importance (\( II \)). When the recommender’s feedback (\( F \)) was lower than the interaction importance (\( II \)), the trust
decline penalty ($P_{TD}$) was determined by the trust model. The effect of the penalty of on/off attack and trust decline in the interaction trust values are illustrated in Figure 2.

The value of interaction trust for malicious users is determined by the trust model by using the penalties of malicious behavior. The effect of new feedback on the interaction trust for malicious users and trusted users is demonstrated in Figures 3 and 4.

**Figure 2.** Penalties of on/off attack and trust decline.

**Figure 3.** Interaction trust values for 100 trusted consumers.
The trust model’s ability to endure reputation attacks is examined in this section by carrying out experiments. The collusion attack frequency (CAF) was calculated by the TMS to prevent collusion attacks. Here, the feedback frequency had a direct relationship with feedback collusion and an inverse relationship with the feedback’s credibility. The feedback frequency was determined by the number of recommender feedback items and the total number of feedback items within the suspected set (SS).

The feedback frequency of seven suspected recommenders is presented in Figure 5 and as we assume in Table 1. It can be seen that the feedback frequency of five suspected recommender was more than the feedback limit (FL), which implies that the TMS will transfer the feedback given by these recommenders to the collusion set (CS), or the suspected feedback by a particular recommender for a particular user will be shifted by the trust model to the feedback set (FS).
Table 1. Collusion attack frequency.

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

Figure 6 shows how the trust model is used to determine the attack scale (AS), which refers to the size of attack scale for various collusion sets. This is done to find out the size of the collusion set (CS) in which the malicious recommenders in a recommender’s community consist of a significant percentage of all recommenders to attack and destroy the trust model.

![Figure 6. Attack scale (AS).](image1)

The value of the attack target scale (ATS), which gives the malicious feedback rate from one collusion set (CS) for a particular user, is illustrated in Figure 7.

![Figure 7. Attack target scale (ATS).](image2)
The value of the collusion attack strength (CAS) determined by computing the rate of all malicious feedback from distinct collusion sets (CS) for a given user is illustrated in Figure 8.

![Figure 8. Collusion attack strength (CAS).](image)

**Comparison of Security and Accuracy**

Any type of trust management service is vulnerable to a variety of attacks [31]. These attacks have the potential to either enhance or destroy the reputation of a specific entity [32,33]. In order to build an accurate and secure trust model system, we focus on implementing various metrics to prevent these attacks, which enables us to create a stable, reliable, and accurate trust model framework. Table 2 shows a comparison between our proposed TMS and those discussed in related works.

**Table 2. Comparison of security and accuracy.**

| ADDRESSED METRICS           | [17] | [21] | [34] | [35] | [36] | OURS |
|-----------------------------|------|------|------|------|------|------|
| Interaction importance      | ✗    | ✓    | ✗    | ✗    | ✗    | ✓    |
| On/off attack               | ✗    | ✓    | ✓    | ✗    | ✓    | ✓    |
| Trust decline               | ✓    | ✓    | ✗    | ✓    | ✗    | ✓    |
| Collusion attack            | ✓    | ✓    | ✗    | ✓    | ✓    | ✓    |
| Collusion attack frequency  | ✗    | ✗    | ✗    | ✓    | ✓    | ✓    |

6. Conclusions

Authorization concerns regarding access to cloud computing storage are a significant challenge for users, particularly big data of cloud computing, which is often highly sensitive. A trust model was presented in this paper that provides dependable solutions for preventing on/off and collusion attacks, which are major security issues faced by cloud computing users. To adequately handle these concerns, control models should be integrated with trust models for decentralized systems through the proposed trust algorithm, which can identify on/off and collusion attacks and ensure the highest confidentiality for cloud service users.

**Author Contributions:** Conceptualization, S.T.A. and K.A.; methodology, S.T.A. and K.A.; software, S.T.A. and K.A.; validation, S.T.A. and K.A.; formal analysis, S.T.A. and K.A.; investigation, S.T.A. and K.A.; resources, S.T.A. and K.A.; data curation, S.T.A. and K.A.; writing—original draft preparation, S.T.A. and K.A.; writing—review and editing, S.T.A. and K.A.; visualization, S.T.A. and K.A.;
supervision, S.T.A. and K.A.; project administration, S.T.A. and K.A.; funding acquisition, S.T.A. and K.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia has funded this project, under grant no. (KEP-9-611-42).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia has funded this project, under grant no. (KEP-9-611-42).

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Noor, T.H.; Sheng, M.; Alfazi, A. Reputation Attacks Detection for Effective Trust Assessment among Cloud Services. In Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, Australia, 16–18 July 2013; pp. 469–476.

2. Chang, E. General Attacks and Approaches in Cloud-Scale Networks. In Proceedings of the IEEE International Conference on Computer Communications, Valencia, Spain, 29 July–1 August 2019.

3. Mahajan, S.; Mahajan, S.; Jadhav, S.; Kolate, S. Trust Management in E-commerce Websites. *Int. Res. J. Eng. Technol. (IRJET)* 2017, 4, 2934–2936.

4. Noor, T.H.; Sheng, M.; Yao, L.; Dustdar, S.; Ngui, A.H. CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services. *IEEE Trans. Parallel Distrib. Syst.* 2015, 27, 367–380. [CrossRef]

5. Varalakshmi, P.; Judgi, T.; Balaji, D. Trust Management Model Based on Malicious Filtered Feedback in Cloud. In *Data Science Analytics and Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 178–187.

6. Li, X.; Du, J. Adaptive and attribute-based trust model for service-level agreement guarantee in cloud computing, *IET Inf. Secur.* 2013, 7, 39–50. [CrossRef]

7. Huang, L.; Xiong, Z.; Wang, G. A Trust-role Access Control Model Facing Cloud Computing. In Proceedings of the 35th Chinese Control Conference, Chengdu, China, 27–29 July 2016.

8. Lin, G.; Wang, D.; Bie, Y.; Lei, M. MTBAC: A mutual trust based access control model in Cloud computing. *China Commun.* 2014, 11, 154–162.

9. Zhu, C.; Nicanfar, H.; Leung, V.; Yang, L.T. An Authenticated Trust and Reputation Calculation and Management System for Cloud and Sensor Networks Integration. *IEEE Trans. Inf. Forensics Secur.* 2014, 10, 118–131.

10. Chen, X.; Ding, J.; Lu, Z. A decentralized trust management system for intelligent transportation environments. *IEEE Trans. Intell. Transp. Syst.* 2020, 1–14. [CrossRef]

11. Zhang, P.; Kong, Y.; Zhou, M. A Domain Partition-Based Trust Model for Unreliable Clouds. *IEEE Trans. Inf. Forensics Secur.* 2018, 13, 2167–2178. [CrossRef]

12. Tan, Z.; Tang, Z.; Li, R.; Sallam, A.; Yang, L. Research of Workflow Access Control Strategy based on Trust. In Proceedings of the 11th Web Information System and Application Conference, Tianjin, China, 12–14 September 2014.

13. Li, X.; Ma, H.; Zhou, F.; Yao, W.; Yao, W. T-Broker: A Trust-Aware Service Brokering Scheme for Multiple Cloud Collaborative Services. *IEEE Trans. Inf. Forensics Secur.* 2015, 10, 1402–1415. [CrossRef]

14. Varsha, M.; Patil, P. A Survey on Authentication and Access Control for Cloud Computing using RBDAC Mechanism. *Int. J. Innov. Res. Comput. Commun. Eng.* 2015, 3, 12125–12129.

15. Li, X.; Ma, H.; Zhou, F.; Gui, X. Service Operator-Aware Trust Scheme for Resource Matchmaking across Multiple Clouds. *IEEE Trans. Parallel Distrib. Syst.* 2014, 26, 1419–1429. [CrossRef]

16. Bhattachali, T.; Chaki, R.; Chaki, N.; Saeed, K. An Adaptation of Context and Trust Aware Workflow Oriented Access Control for Remote Healthcare. *Int. J. Softw. Eng. Knowl. Eng.* 2018, 28, 781–810. [CrossRef]

17. Noor, T.H.; Sheng, Q.Z.; Alfazi, A. Detecting Occasional Reputation Attacks on Cloud Services. In *Web Engineering*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 416–423.

18. Labraoui, N.; Geraoui, M.; Sekhri, L. On-Off Attacks Mitigation against Trust Systems in Wireless Sensor Networks. In Proceedings of the 5th International Conference on Computer Science and Its Applications (CIIA), Saida, Algeria, 20–21 May 2015; pp. 406–415.

19. Noor, T.H.; Sheng, Q.Z.; Bouguettaya, A. *Trust Management in Cloud Services*; Springer: Berlin/Heidelberg, Germany, 2014.

20. Tong, W.M.; Liang, J.Q.; Lu, L.; Jin, X.J. Intrusion detection scheme based node trust value in WSNs. *Syst. Eng. Electron.* 2015, 37, 1644–1649.

21. Ghafoorian, M.; Abbasinezhad-Mood, D.; Shakheri, H. A Thorough Trust and Reputation Based RBAC Model for Secure Data Storage in the Cloud. *IEEE Trans. Parallel Distrib. Syst.* 2018, 30, 778–788. [CrossRef]
22. Nwebonyi, F.N.; Martins, R.; Correia, M.E. Reputation based approach for improved fairness and robustness in P2P protocols. *Peer-to-Peer Netw. Appl.* 2018, 12, 951–968. [CrossRef]

23. Deng, W.; Zhou, Z. A Flexible RBAC Model Based on Trust in Open System. In Proceedings of the 2012 Third Global Congress on Intelligent Systems, Wuhan, China, 6–8 November 2012; pp. 400–404.

24. Liang, J.; Zhang, M.; Leung, V.C. A reliable trust computing mechanism based on multisource feedback and fog computing in social sensor cloud. *IEEE Internet Things J.* 2020, 7, 5481–5490. [CrossRef]

25. Zhou, L.; Varadharajan, V.; Hitchens, M. Integrating Trust with Cryptographic Role-Based Access Control for Secure Cloud Data Storage. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, Australia, 16–18 July 2013; pp. 1–12.

26. Chang, W.; Xu, F.; Dou, J. A Trust and Unauthorized Operation Based RBAC (TUORBAC) Model. In Proceedings of the International Conference on Control Engineering and Communication Technology, Shenyang, China, 7–9 December 2012.

27. Marudhadevi, D.; Dhatchayani, V.N.; Sriram, V.S. A Trust Evaluation Model for Cloud Computing Using Service Level Agreement. *Comput. J.* 2014, 58, 2225–2232. [CrossRef]

28. Tsai, W.T.; Zhong, P.; Bai, X.; Elston, J. Role-Based Trust Model for Community of Interest. In Proceedings of the 2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA), Taipei, Taiwan, 14–15 January 2009.

29. Fan, Y.; Zhang, Y. Trusted Access Control Model Based on Role and Task in Cloud Computing. In Proceedings of the 7th International Conference on Information Technology in Medicine and Education, Huangshan, China, 13–15 November 2015.

30. Bhatt, S.; Sandhu, R.; Patwa, F. An Access Control Framework for Cloud-Enabled Wearable Internet of Things. In Proceedings of the 3rd International Conference on Collaboration and Internet Computing (CIC), San Jose, CA, USA, 15–17 October 2017; pp. 213–233.

31. Alshammari, S.; Telaihan, S.; Eassa, F. Designing a Flexible Architecture based on mobile agents for Executing Query in Cloud Databases. In Proceedings of the 21st Saudi Computer Society National Computer Conference (NCC), Riyadh, Saudi Arabia, 25–26 April 2018; pp. 1–6.

32. Alshammari, S.; Albeshri, A.; Alsuhbi, K. Integrating a High-Reliability Multicriteria Trust Evaluation Model with Task Role-Based Access Control for Cloud Services. *Symmetry* 2021, 13, 492. [CrossRef]

33. Alshammari, S.T.; Albeshri, A.; Alsuhbi, K. Building a trust model system to avoid cloud services reputation attacks. *Egypt. Inform. J.* 2021. [CrossRef]

34. Uikey, C.; Bhilare, D.S. TrustRBAC: Trust role based access control model in multi-domain cloud environments. In Proceedings of the 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), Indore, India, 17–19 August 2017; pp. 1–7.

35. Fortino, G.; Fotia, L.; Messina, F.; Rosaci, D.; Sarne, G.M.L. Trust and Reputation in the Internet of Things: State-of-the-Art and Research Challenges. *IEEE Access* 2020, 8, 60117–60125. [CrossRef]

36. Barsoum, A.; Hasan, A. Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems. *IEEE Trans. Parallel Distrib. Syst.* 2012, 24, 2375–2385. [CrossRef]