Book Chapter

A High-Reliability Trust Management System in Cloud Computing

Salah T Alshammari*, Aiiad Albeshri and Khalid Alsubhi

Department of Computer Science, College of Computing and Information Technology, King Abdul-Aziz University, KSA

***Corresponding Author:** Salah T Alshammari, Department of Computer Science, College of Computing and Information Technology, King Abdul-Aziz University, Jeddah 21589, Saudi Arabia

Published **June 28, 2021**

This Book Chapter is a republication of an article published by Salah T Alshammari, et al. at Symmetry in March 2021. (Alshammari, S.T.; Albeshri, A.; Alsubhi, K. Integrating a High-Reliability Multicriteria Trust Evaluation Model with Task Role-Based Access Control for Cloud Services. Symmetry 2021, 13, 492. https://doi.org/10.3390/sym13030492)

How to cite this book chapter: Salah T Alshammari, Aiiad Albeshri, Khalid Alsubhi. A High-Reliability Trust Management System in Cloud Computing. In: Konan-Marcelin Kouamé, editor. Prime Archives in Symmetry. Hyderabad, India: Vide Leaf. 2021.

Author Contributions: Conceptualization, S.A. , A.A. and K.A.; methodology, S.A. , A.A. and K.A.; software, S.A. , A.A. and K.A.; validation, S.A. , A.A. and K.A.; formal analysis,

S.A. , A.A. and K.A.; investigation, S.A. , A.A. and K.A.; resources, S.A. , A.A. and K.A.; data curation, S.A. , A.A. and K.A.; writing—original draft preparation, S.A. , A.A. and K.A.; writing—review and editing, S.A. , A.A. and K.A.; visualization, S.A. , A.A. and K.A.; supervision, S.A. , A.A. and K.A.; project administration, S.A. , A.A. and K.A.; funding acquisition, S.A. , A.A. and K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (D-067-611-1442). The authors, therefore, gratefully acknowledge the DSR for technical and financial support.

Acknowledgments: This project was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under grant No. (D-067-611-1442). The authors, therefore, gratefully acknowledge the DSR for technical and financial support.

All praises to Allah and His blessing for the completion of this work. I thank God for all the opportunities, trials and strength that have been showered on me to finish writing this manuscript. I would like to thank my parents who made all of this possible, for their endless love, support and patience. They have been always there for me whenever I needed them.

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

Abstract

Cloud data storage is revolutionary because it eliminates the need for additional hardware, which is often costly, inconvenient, and requires additional space. Cloud data storage allows data owners to store large amounts of data in a flexible way and at low cost. The number of online cloud storage services and their consumers has therefore increased dramatically. However, ensuring the privacy and security of data on a digital platform is often a challenge. A cryptographic task-role-based access control (T-RBAC) approach can be used to

protect data privacy. This approach ensures the accessibility of data for authorized consumers and keeps it safe from unauthorized consumers. However, this type of cryptographic approach does not address the issue of trust. In this paper, we propose a comprehensive trust model integrated with a cryptographic T-RBAC to enhance the privacy and security of data stored in cloud storage systems, and suggests that trust models involve inheritance and hierarchy in the roles and tasks of trustworthiness evaluation, where this study aims to identify the most feasible solution for the trust issue in T-RBAC approaches. Risk evaluations regarding other possible flaws of the design are also performed. The proposed design can decrease risk by providing high security for cloud storage systems and improve the quality of decisions of cloud operators and data owners.

Keywords

Access Control; Authorization; Cryptography; Databases; Encryption; Information Security; Online Services; Security; Security Management; Web Services

Introduction

Cloud computing technology is an internet-based computing modality that offers on-demand resources, online storage, and access to resources that are not located on local servers. The use of online cloud computing services has increased with the development of internet technology [1–3]. Cloud computing is a diverse utility that involves clients who are both insiders and outsiders, as well as servers for storage, internet systems, and cloud providers [4]. This computing platform has gained dominance in the business sector as a result of its minimal operational and maintenance costs. Service providers use cloud computing to provide their customers with reliable, quick, and flexible services [5]. While offering these services to cloud consumers, service providers need to prioritize the security and privacy of cloud data. Privacy and security have always been a challenge for digitally available data as most of the data is offered in the public domain. To curb this situation, we need to

use a cryptographic approach, T-RBAC, to provide protection for the privacy of data available on digital platforms of cloud computing. T-RBAC enables authorized consumers to access data but denies unauthorized users the opportunity to reach the data. Nonetheless, such a cryptographic approach is not capable of addressing issues pertaining to trust. In this paper, we propose a comprehensive trust model integrated with a cryptographic T-RBAC to enhance the privacy and security of data stored in cloud storage systems, and suggest that trust models involve inheritance and hierarchy in the roles and tasks of trustworthiness evaluation, where this study aims to identify the most feasible solution for the trust issue in T-RBAC approaches. The proposed design can decrease risk by providing high security for cloud storage systems and improve the quality of the decisions of cloud operators and data owners.

Different providers of cloud services may offer a wide range of cloud services, which may include IaaS, PaaS, SaaS, either singly or as a combination of all of them in a public domain by use of the internet. The providers of cloud services can make advertisements for their cloud services by using the internet, e.g., via search engines. Other consumers of such cloud services, on the other hand, may use the available cloud services to host their own services, which is common for new startups with limited funding. A cloud service provider may give feedback or make inquiries concerning the trust values of cloud consumers by invoking a management trust service which is constituted of many distributed nodes. The nodes of a trust management system (TMS) expose interfaces to receive feedback or inquiries concerning trust results from the consumers in a decentralized manner [6–8]. The TMS discovers the available cloud services via the internet and allows consumers to access the new cloud service's trust by searching. The TMS may also market the trust to consumers as a service through the internet.

Cloud Access Control Systems

Security is crucial to cloud services primarily because these services are mostly offered in the public domain [9,10]. Public platforms are accessible to a variety of users, both insiders and

outsiders; hence, the security of such services needs to be maintained. In some cases, the owners of the data stored in the cloud require a high level of privacy from the public and the cloud providers themselves. To offer privacy and security in such a situation, access controls are used [11–13]. Several factors must be considered when selecting an appropriate access control model; these include cost-effectiveness, affordability, security, efficiency in curbing misconduct, and the trust built as a result of using the model. Trust is a critical element to secure cloud computing services [14,15]. The authors in [1] define trust as "an implicit property that exists in the background rather than being an explicit property, which is well-defined and quantifiable". These models include attribute-based access control (ABAC), role-based access control (RBAC), discretion access control (DAC), and mandatory access control (MAC) [16–18]. Integrating trust with cryptographic task-role-based access control (T-RBAC) to secure data stored in the cloud, on the other hand, is more convenient and enhances cloud users' interaction in the roles and tasks associated with the owners' data. There is also a need for a control method that will develop trust regarding the tasks carried out by individual users. In this research, we propose a design that allows a trust model to be integrated with a cloud storage system that uses a cryptographic T-RBAC approach.

Several studies have been performed regarding access control policies and system designs deemed suitable for the implementation of cloud computing frameworks. In this study, we introduce the T-RBAC as a new access control model integrated with a comprehensive trust model by applying different criteria to provide high security for cloud storage systems while taking flexibility into consideration. T-RBAC functions in such a way that all users in the system are assigned to one role or different roles, and each role has many tasks within an enterprise setting. Although T-RBAC provides a complex security mechanism that supports the security structure of a network with physical servers, it is not suitable for distributed environments. To improve security in cloud computing environments, it is imperative to employ trust models

to enhance the privacy and security of the data stored in cloud storage systems.

In cloud computing, many existing access control models have trust-based systems, but these trust models do not offer precise trust value computations because they may be exposed to some security threats. Thus, an accurate trust-based T-RBAC model is proposed to resolve this vulnerability [19]. The T-RBAC model is based on the RBAC model, but the T-RBAC model takes the idea of tasks into consideration, thereby providing more security and flexibility than the RBAC model [20,21]. Therefore, we propose a trust model that uses the T-RBAC model to manage access to cloud storage. T-RBAC supports dynamic real-time security management and applies the concept of access control over tasks, which are the minimum units for cloud computing activities.

In Figure 1, the permissions in the RBAC model are given to the role, and the permissions of these roles are assigned to specific consumers [22–24]. Figure 2 shows task-based access control (TBAC), where the permissions are assigned to tasks, and the tasks are assigned to specific consumers [20]. Because of this, the TBAC model creates a direct relationship between the consumer and the task, which makes maintenance management more complex. In Figure 3, the permissions in the T-RBAC model are assigned either to roles or tasks [20,25]. If permissions are assigned to tasks, then tasks are assigned to roles, and so a consumer can obtain permission to access the cloud storage through either roles or tasks. Therefore, tasks are the minimum units in our architecture. In our trust model, the roles represent the consumers' roles for all consumers that are given permission to access the service data and a task represents one operation of the service data.

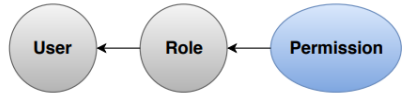


Figure 1: Role-based access control (RBAC) model.

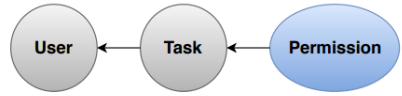


Figure 2: Task-based access control (TBAC) model.

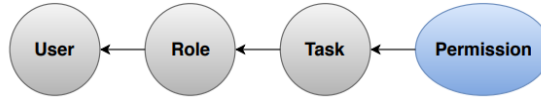


Figure 3: Cryptographic task-role-based access control (T-RBAC) model.

System Motivations

Figure 4 demonstrates a trust-based T-RBAC model in the cloud environment. A T-RBAC model has four main parties. The first are the data owners who have a tendency to share and store their information or resources within the cloud. Then, there are the users who need to use the shared resources of the owners. The third party refers to the roles that provide access levels to the registered cloud users, and the fourth are the tasks that are allocated for each role. Three possible models emanate from the integration of the trust model with the T-RBAC model in the enhancement of the cloud computing storage system: recommender owner-role, consumer-role, and consumer-task based T-RBACs. The data owners can evaluate the trust value of the roles and tasks, which in turn evaluates the user's trustworthiness in the three models.

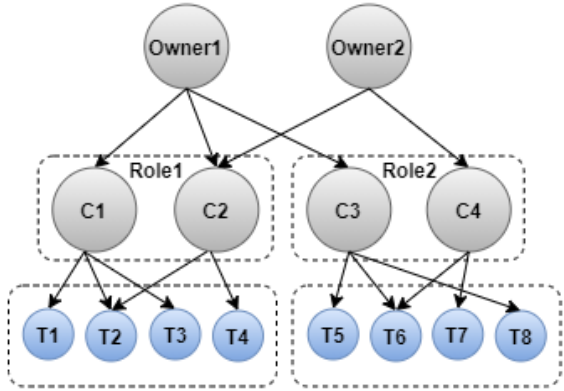


Figure 4: Trust based T-RBAC model.

The T-RBAC system model works as follows. First, resources/services are shared by the owners, i.e., the service providers (SPs), with the roles or tasks based on roles' and tasks' trust level and reputation.. The registered users in each rolereceive access to the shared resources/services when their trust level is more than the minimum trust acceptance level of the roles, while the registered users in tasks receive access to the shared services or resources when their trust level is more than the minimum trust acceptance level of the tasks. It is clear that the reputation of the role or task that has been registered by the user will change with modifications in the trust level of users, which will consequently alter the users' privileges. There are five kinds of trust in this system: the owner's trust in the recommendations of the other owners; the owner's trust in the functionality of the tasks; the owner's trust regarding the functionality of the roles; the role's trust regarding the users' functionalities; and task's trust in the users' functionality. The motivation behind choosing this T-RBAC approach is as follows:

- It provides the highest degree of flexibility, which is the basis for interactions in the cloud environment, by enhancing cloud users' interaction in the roles and tasks associated with the owners' data.

- It develops trust regarding the tasks carried out by individual users.
- Many existing access control models have trust-based systems, but these trust models do not offer precise trust value computations because they may be exposed to some security threats. Thus, an accurate trust-based T-RBAC model is proposed to resolve this vulnerability.
- T-RBAC supports dynamic real-time security management and applies the concept of access control over tasks, which are the minimum units for cloud computing activities.

Problem Statement

Issues of authorization or authorized access to computers or computing devices owned by organizations, whether profit or nonprofit, are serious concerns that can also occur in open environment systems such as a cloud computing system [26,27]. Hence, access controls are commonly used in server applications for cloud computing platforms [28,29]. However, all previous studies have found that access controls are not reliable overall for distributed systems, the primary reason being that the population of consumers is dynamic and complex and their identities are not established in advance [30,31]. With such concerns, integrated trust models with access control models are the best option for decentralized systems with complex consumer bases [23]. This particular model is the result of many attempts made by trust model developers to design new trust models that can resolve the most complicated and advanced authorization issues. However, many access controls that are integrated with trust models that have been already proposed [5,32,33] are defenseless against some attacks. In light of this, we will use a T-RBAC model as a new access control model for a cloud storage system. However, despite T-RBAC's status as a new access control in cloud computing, we must take the evolution of security threats over time into consideration.

Contribution

This research focuses on the T-RBAC model and its application to access control to provide secure cloud data storage. The main

contribution is that T-RBAC uses cryptography to ensure the safety of data within a cloud storage system. There already exist several studies of trust management systems with the RBAC model, but we are introducing the task-role-based access control or T-RBAC as new access control integrated with a comprehensive trust model by applying different criteria to provide flexibility and high security for cloud storage systems. This research proposes the use of tasks and roles together with cryptography to provide better protection for the stored data.

The trust in a consumer depends on the historical behavior of the tasks and the roles. Therefore, in this research, the impacts of task hierarchy and inheritance on the trust in the task and consumer are evaluated. Additionally, we identify the criteria that need to be applied when using trust-based systems and suggest an acceptable comprehensive trust-based T-RBAC model that provides privacy and security of data storage in cloud systems to fulfill the desired security requirements, and that provides an efficient and reliable architecture. Finally, we used T-RBAC to simplify security management on a large scale. Tasks are used to give consumers permission to access an available resource, assigned according to the tasks performed by the consumer. The T-RBAC model is suitable for distributed computing activities with multiple access points and controls and lays the foundation for active security models. Active security models are models that approach security modeling from the perspective of tasks; these tasks provide an active work mechanism and security management as the task proceeds to completion. To make this effective, we propose a trust model that accomplishes the following:

- Protects cloud data from various attacks such as on/off attacks, collusion attacks, and sybil attacks.
- Upholds the utmost privacy of the cloud consumers' data, as their use of the cloud services may involve highly sensitive data.
- Offers comprehensive solutions to make consumers' trust values very precise by applying the interaction importance criterion.

- Assures trust management service availability owing to cloud services' dynamic nature.
- Applies several criteria to ensure accurate assessment and data protection.

Organization

The rest of this paper is organized as follows. In section 2, we review related works. In section 3, we present the design and methodology as a solution to protect cloud storage from any attack. In section 4, we present the design of our approach. In section 5, we present the simulation results for our system. Finally, the conclusion of this work is presented in section 6.

Related Works

By utilizing a cryptographic RBAC model with a trust model in cloud systems, the owner of the data can provide specific consumers with permission to access the data, where only the consumer who is allowed to access the role can decode the information (Zhou, Vijay, and Hitchens, [5]). The consumer should have a high trust value in various roles to allow the service provider to decide the trustworthiness of a consumer role in the trust model and give permission for future data access, enabling the owners of the data to utilize the trust evaluations to decide whether the data should be stored in the cloud for a specific consumer role. Another concept employed in that study is that the proposed trust models consider the roles of hierarchy and legacy in the evaluation of the trustworthiness of the consumer's role. The authors proposed the RBAC model integrated with the trust model, and show two probable models for trust in cloud storage, Role–User and Owner–Role RBACs, to enhance data security in cloud computing. In these two models, service providers (SPs) can evaluate the trust value of roles, and the roles evaluate the trust in the user. In any case, there are some security challenges with these two models: they are defenseless against collusion attacks and on/off attacks. To improve the quality and proficiency of cloud-based frameworks, the authors of another study advanced "Trust RBAC," a new trust-based RBAC prototype [31]. As opposed to [5], the authors

proposed a trust model that can protect the cloud system from an on/off attack, but which is still vulnerable to some security threats such as collusion attacks. In 2009, Hasan et al. [34] proposed a new solution to remove the element of subjectivity from trust recommendations. In 2011, Tan et al. [12] put forward an innovative trust model based on the RBAC model to secure cloud-based storage, but careful evaluation of their system shows that it is unable to satisfy pertinent security problems. In 2012, Barsoum and Hasan [13] introduced a storage model for the cloud environment with several features. A crucial highlight of their trust model is the ability to empower common trust for both proprietors and cloud service providers. In 2013, Noor et al. [35] built a new system to detect collusion attacks and sybil attacks in cloud computing systems. In 2018, Mahdi et al. [32] proposed a trust model based on RBAC that provides different criteria but that is still defenseless against sybil attacks.

In a 2020 study [36], 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. In another study in 2021 [37] the author proposes a new trust model integrated with the access control model based on privacy and permission, which aims to protect user privacy and solve the privacy disclosure problem in cloud environment, but which is still defenseless against some attacks.

Design and Methodology

This section examines the trust- and reputation-based model's ability to meet the requirements of an exact trust model system to defeat the previously mentioned security problems. We will propose strategies that must be taken into consideration when designing a trust evaluation process. We will seek to determine the best possible answer to the problem of trust in T-RBAC. In the investigation, it was determined that the proposed trust models provide hierarchy and inheritance 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 a

cryptographic T-RBAC approach. There are two kinds of trust: interaction trust (IT), which we could also call consumer trust, which refers to a service provider's trust in consumers' use of their service, and recommendation trust (RT), which refers to the service provider's trust in other service providers' recommendations.

Interaction Trust

Interaction trust (IT) represents the point of view of the service providers toward consumers depending on their own interaction experiences.

Interaction Importance

To make the interaction value very accurate, SPs' interactions are distinguished based on their sensitivity and importance. Interaction importance (II) is the most required element in the procedure of evaluating trust. Our trust evaluation model has a favorable interaction importance structure since it enables a service provider to give a recommendation to other service providers based on the trust value information of their consumers as a percentage [32]. To calculate interaction trust (IT), most researchers have utilized likely prototype models. Among those that utilize aggregate negative and positive feedback to evaluate interaction trust (IT), such as [38] and [39], the beta distribution function stands out. It includes α and β in evaluating trust interactions. All of the existing prototypes that utilize a beta distribution function to calculate the value of the consumer trust increment α by 1 if there is any positive feedback and β by 1 if there is any negative feedback. However, this methodology is not precise, as it does not account for the importance of interactions. Accordingly, to address these cases, we will propose a novel methodology that calculates IT based on precise feedback by taking interaction importance into consideration. In our proposed TMS, feedback (F) regarding an interaction is expressed as a number. In other words, as an ordered feedback set from 1 to n, 1 represents "untrusted" and n represents "highly trusted." In other words, a feedback set contains n satisfaction levels that a service provider decides for

any interaction feedback. The n rate is a whole number controlled by the administrator of the system depending on the complexity of the trust value. This approach registers IT as well as incorporating an SP′s interaction importance (II) in the process of trust evaluation. The posited IT model is given as follows.

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

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

$$\sum_{i=1}^{n} \beta_i^t(CR) = \begin{array}{c} \\ CR_1 \\ CR_2 \\ \vdots \\ CR_n \end{array} \begin{array}{c} \begin{array}{ccccc} \beta_1^t & \beta_2^t & \cdots & \beta_{n-1}^t & \beta_n^t \end{array} \\ \begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n-1} & V_{n,n} \end{pmatrix} \end{array}$$

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

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

where t represents time, α^t represents positive feedback given for a specific consumer role (CR), β^t represents negative feedback given for a specific consumer role (CR), P represents the value of new positive feedback, N represents the value of new negative feedback, II represents the value of interaction importance, and NF represents the number of feedback inputs.

On/off Attack

In many cases, malicious consumers can increase their trust value if they act well in unimportant interactions, after which

they can use this trust value to act maliciously in important interactions. This is known as an on/off attack. To avoid the dangers of on/off attacks (O^2), we need an on/off attack penalty (P^{O^2}), where P^{O^2} is set from 1 to n, such that 1 represents no danger from this role and n represents high danger. We need to apply a new procedure to allow the trust model to calculate P^{O^2} for any specific consumer, where PC^{O^2} is > 0.5, which represents the curve of a minimum of high interactions.

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

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

Trust Decline

In some cases, the functionality of the system depends on the trust in users in one another. Therefore, if a system flaw in the trust management system drives its users to doubt one another, all system functions will break down. A gradual decline in trust can happen if data incidents caused by the malignant actions of different users are leaked, in which case doubt is cast on every user in the system. In this way, vindictive users can incapacitate the entire trust model. To defeat this problem, rather than placing all clients in doubt, we need a way to diminish malignant users' influence immediately. To alleviate the dangers of trust decline (TD) we need a penalty of trust decline (P^{TD}), where P^{TD} is set from 1 to n, to such that 1 represents no danger from this role and n represents high danger, PC^{TD} represents the curve of a penalty of trust decline and L^{II} is an integer greater than one, represents the limit of low interaction.

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

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

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

Task Trust

The owners of the resources can give permissions for consumers' roles and tasks, and if there is a leak of data that can stop the role or task, each task represents one operation in any cloud computing service. In order to have a more flexible system that is not interrupted, the system identifies the tasks in which data leaks have occurred and then messages the owners to prevent the users of a role from accessing these tasks. Therefore, if a user of a role has caused a data leak involving one task, then the trust model will send feedback to the other owners to stop this task from accessing the resources. The trust in the task can be calculated as (4).

$$IT(T) = \sum_{i=1}^{n-1} \frac{\alpha_i^t(T) \times P_T}{\left(\alpha_i^t(T) + P_T\right) + \left(\beta_i^t(T) + N_T\right)} \tag{1}$$

$$\sum_{i=1}^{n} \alpha_i^t(T) = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ T_n \end{array} \begin{array}{c} \begin{array}{cccccc} \alpha_1^t & \alpha_2^t & \cdots & \alpha_{n-1}^t & \alpha_n^t \end{array} \\ \begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n-1} & V_{n,n} \end{pmatrix} \end{array}$$

$$\sum_{i=1}^{n} \beta_i^t(T) = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ T_n \end{array} \begin{array}{c} \begin{array}{cccccc} \beta_1^t & \beta_2^t & \cdots & \beta_{n-1}^t & \beta_n^t \end{array} \\ \begin{pmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,n-1} & V_{1,n} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,n-1} & V_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ V_{n,1} & V_{n,2} & \cdots & V_{n,n-1} & V_{n,n} \end{pmatrix} \end{array}$$

$$P_T = \frac{\alpha_n^t(T) \times II}{NF}$$

$$N_T = \frac{\beta_n^t(T) \times II}{NF}$$

Inheritance Interaction Trust

Inheritance interaction trust (IIT) represents credence that arises due to a history of interaction between certain roles that possess inheritance relations with a consumer role. Firstly, we delve into the specific inheritance trust in which only the descendant roles' interaction history is considered. Once the owner senses a data leak through a subrole of the role, the owner's feedback needs to be applied to that subrule and also influence R's trust (since role users equally access the owner's data allotted to the subrole, and are thus suspicious of triggering a fruitless interaction). Hence, while examining the role's trust, the interaction history belonging to its subroles is considered.

Assuming that consumer role CR bears n instant descendant roles, we shall have $\{CR_1, CR_2, .., CR_m\}$, and if there are different consumer tasks T assigned to this role, we shall have $\{T_1, T_2, .., T_n\}$. The trust value can be elucidated as $IIT(CR)$, which represents the inheritance interaction trust value of the role. Further, the aggregate count of the roles inherited from the consumer role (CR) can be elucidated as $NCSR$ and the tasks inherited from the consumer role (CR) can be elucidated as NT. IS^{CSR} represents the percentage impact size of the subroles on the trust value, IS^{T} represents the percentage impact size of the consumer tasks on the trust value, and IS^{CR} represents the percentage impact size of the roles on the trust value, with all impact size factors determined by the cloud administrator. Then, the inheritance trust estimate arising out of tasks of roles can be calculated as (5).

$$IIT(CR) = IR \times IS^{CSR} + \frac{\sum_{i=1}^{n} IT(T)}{NT} \times IS^{T} + IT(CR) \times IS^{CR} \qquad (5)$$

If the role inherits one or more subroles, then IR will be the average of the trust values of all subroles, but if there is no subrole for this role, the IR will be equal to the trust value of the consumer role.

$$\begin{cases} \textbf{\textit{if}}\ NCSR \geq 1\ \ \textbf{\textit{then}}\ \ IR = \dfrac{\sum_{i=1}^{n} IT(CSR_i)}{NCSR} \\ \textbf{\textit{else}} \qquad\qquad\qquad\qquad IR = IT(CR) \end{cases}$$

We can examine the example of the role hierarchy detailed in Figure 5. The example should exhibit how the role hierarchy impacts the roles' trust values.

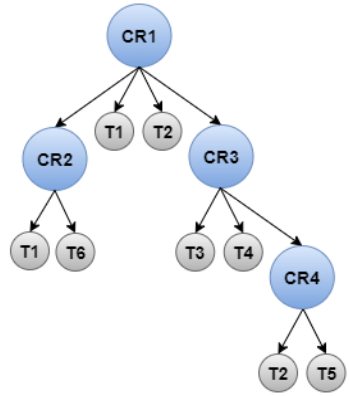


Figure 5: Example of roles and tasks in T-RBAC.

As shown in Figure 5, $CR1$ inherits from both $CR2$ and $CR3$ and $T1$ and $T2$ are assigned to $CR1$, For simplification, let us assume that values of $IT(CR1) = 80\%$, $IT(CR2) = 80\%$, and $IT(CR3) = 60\%$; $IT(T1) = 20\%$ and $IT(T2) = 60\%$; and the impact size factors are $(IS^{CSR} = 0.1, IS^{T} = 0.2\ , IS^{CR} = 0.7)$. This yields:

$$IIT\left(CR1\right) = \frac{0.8+0.6}{2} \times 0.1 + \frac{0.6+0.2}{2} \times 0.2 + 0.8 \times 0.7 = 77\%$$

where in calculating the trust value of $CR4$, it does not inherit from any subrole. Let us further assume values of $IT(CR4) = 90\%$, $IT(T2) = 60\%$, and $IT(T5) = 80\%$. This yields:

$$IIT\left(CR4\right)=0.9\times0.1+\frac{0.6+0.8}{2}\times0.2+0.9\times0.7=86\%$$

Nonsymmetry

For each interaction, there is unilateral trust; for instance, if node A trusts node B, this does not necessarily mean that node B trusts node A. The owner obtains the role interaction's history $H(CR)$ and the history of the task interaction $H(CT_R)$ from the central repository where the history of the interaction is derived from the owner feedback of a role or task. The individual trust value or the interaction history of the role R or task T is computed as:

$$H_F(CR) = (H_1^{CR}, H_2^{CR}, \dots, H_n^{CR})$$

$$H_F(T_R) = (H_1^{T_R}, H_2^{T_R}, \dots, H_n^{T_R})$$

$$H_i^{CR} = (ID_i\ , T(V))$$

The algorithm of interaction trust (IT) is calculated as follows:

Algorithm 1: Interaction Trust 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 $P^{O^2} = 1$
  end if
  if $\alpha_n^t < II$ ***then*** $P^{TD} = PC^{TD}$
    else $P^{TD} = 1$
  end if
  for $i = 1, \quad i \leq n-1$
    $IT(CR) \leftarrow (\alpha_i^t(CR) + P_{CR}) / \left((\alpha_i^t(CR) + P_{CR}) + \left(\beta_i^t(CR) + N_{CR} \times P^{O^2} \times P^{TD}\right)\right)$
end for

15: ***end procedure***
16: ***procedure*** $Task\ Trust$
17: $\alpha_n^t(T) = F$
18: $\beta_n^t(T) = F - 1$
19: $P_T \leftarrow (\alpha_n^t(T) \times II)/NF$
20: $N_T \leftarrow (\beta_n^t(T) \times II)/NF$
21: ***for*** $i = 1, \quad i \leq n-1$
22: $IT(T) \leftarrow (\alpha_i^t(T) + P_T)/\left((\alpha_i^t(T) + P_T) + (\beta_i^t(CR) + N_T)\right)$
23: ***end for***
24: ***end procedure***
25: ***procedure*** $Inheritance\ Interaction\ Trust$
26: ***for*** $i = 1$ ***to*** n
27: ***if*** $NCSR \geq 1\ \ then\ \ IR = IT(CSR_i)/NCSR$
28: ***else*** $IR = IT(CR)$
29: ***end if***
30: ***end for***
31: ***for*** $i = 1$ ***to*** n
32: $IIT(CR) \leftarrow (IR \times IS^{CSR}) + (IT(T)/NT \times IS^{T}) + (IT(CR) \times IS^{CR})$
33: ***end for***
34: ***end procedure***

Recommendation Trust

If there is insufficient proof or information to calculate interaction trust about specific consumer, a service provider needs to evaluate recommendation trust (RT). RT refers to a service provider's trust in a consumer by accepting the recommendations of other service providers that have already interacted with that consumer. Recommender importance, subjectivity, collusion attack, and sybil attack are problems that ought to be considered in a recommendation trust evaluation. To deal with these, some critical issues should be connected in the recommendation trust evaluation.

Collusion Attack

TMSs can also be compromised via collusion attacks. Collusion implies a group of users cooperating to decrease or increase the trust value of any consumer in the system [32,40]. In other words, a collusion attack happens when a group of users collaborate with one another to destroy another person's reputation or raise their own position by giving false

recommendations [41]. If more than 50% of users in a trust model system are malicious, such attacks will be ineffective. In any case, however, collusion attacks threaten the accuracy of recommendation trust (RT). To reduce the danger of this attack, Mahdi et al. [32] proposed a solution that includes the minimum number of recommenders that should contribute to the computation of recommendation trust. An RBAC system will readily adjoin virtual recommenders to the compilation process of the recommendation trust, but there remains the danger of coming under attack.

There are two kinds of collusion attacks. In a self-promoting attack, a group of users cooperate to increase the trust value of any consumer in the system. In a slandering attack, a group of users cooperate to decrease the trust value of any consumer in the system. In order to avoid this type of attack, we need to calculate three criteria, each with several factors. The first criterion, malicious recommendation detection (MRD), assesses the probability of a suspicious group being a collusion group. To detect these groups, we need to calculate the time range of feedback given to detect all suspicious feedback, which will be very small for those that set out to attack a suspicious consumer. After that, the trust model will compute the second criterion, which is malicious recommenders' behavior (MRB), to measure the similarity of all recommenders' behavior, which will be high when malicious recommenders attack any consumer. The malicious recommendations detection (MRD) is calculated as follows.

$$
\begin{cases}
\textbf{for } i = 1 \textbf{ to } n-1 \\
\quad \begin{cases}
\qquad\qquad \textbf{if } T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \le TR \\
\textbf{and } V(F_n, CR)^{FS} \ge V(F_i, CR)^{FS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \le maxVR \\
\qquad\qquad \textbf{then move } (F_i, CR)^{FS} \textbf{ from } FS \textbf{ to } SS \\
\qquad\qquad \textbf{elseif } T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \le TR \\
\textbf{and } V(F_n, CR)^{FS} < V(F_i, CR)^{FS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{FS} \ge minVR \\
\qquad\qquad \textbf{then move } (F_i, CR)^{FS} \textbf{ from } FS \textbf{ to } SS \\
\qquad\quad \textbf{endelseif} \\
\quad \textbf{endif}
\end{cases} \\
\textbf{endfor}
\end{cases}
$$

$$\begin{cases} \textbf{for } i = 1 \textbf{ to } n \\ \quad \begin{cases} \quad\quad\quad\quad \textbf{if } T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \le TR \\ \textbf{and } V(F_n, CR)^{FS} \ge V(F_i, CR)^{SS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \le maxVR \\ \quad\quad\quad\quad \textbf{then move } (F_n, CR)^{FS} \textbf{ from } FS \textbf{ to } SS \\ \quad\quad\quad\quad \textbf{elseif } T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \le TR \\ \textbf{and } V(F_n, CR)^{FS} < V(F_i, CR)^{SS} \textbf{and } V(F_n, CR)^{FS} - V(F_i, CR)^{SS} \ge minVR \\ \quad\quad\quad\quad \textbf{then move } (F_n, CR)^{FS} \textbf{ from } FS \textbf{ to } SS \\ \quad\quad\quad \textbf{endelseif} \\ \quad\quad \textbf{endif} \\ \quad \textbf{endfor} \end{cases} \end{cases}$$

$$\textbf{where } \begin{cases} TR = T(F_n, CR)^{FS} \times TC \\ maxVR = V(F_n, CR)^{FS} \times VC \\ minVR = -V(F_n, CR)^{FS} \times VC \end{cases} \quad (6)$$

The trust model will compare the time of the last feedback $T(F_n, CR)^{FS}$ given to a specific consumer in a feedback set (FS) with the times of all $T(F_i, CR)^{FS}$ except the last feedback, and the value of that the last feedback $V(F_n, CR)^{FS}$ given to a specific consumer in a feedback set (FS) with the value of all $V(F_i, CR)^{FS}$ except the last feedback. Then, the trust model will compare between the time of the last feedback $T(F_n, CR)^{FS}$ given to a specific consumer in a feedback set (FS) and the times of all $T(F_i, CR)^{SS}$ for a specific consumer in a suspected set (SS), and between the value of the last feedback $V(F_n, CR)^{FS}$ that is given to a specific consumer in feedback set (FS) and the value of all $V(F_i, CR)^{SS}$ for a specific consumer in SS. In this way, if any two feedback items in FS have the same time and value ranges, the trust model will move these feedback items to the suspected set (SS). VC and TC are two parameters to determine the time and value ranges of feedback.

The third criterion is collusion attack frequency (CAF), which detects malicious recommendations for each recommender with feedback in the suspected set $SS = \{SF_1, SF_2, \dots, SF_n\}$, where the strength of the attacks is measured in terms of frequency of attack. Thus, we have (7):

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

$$\begin{cases} if\ CAF(SR_i, CR) \geq FL\ \textbf{then move}\ SF(SR, CR)\ to\ CS \\ \textbf{else} \qquad\qquad\qquad\qquad \textbf{move}\ SF(SR, CR)\ to\ FS \\ \textbf{endif} \end{cases} \quad (7)$$

where $CAF(SR_i, CR)$ is equal to the number of feedback items given from a specific recommender to a specific consumer in the suspected set (SS) divided by number of all feedback items that are given to the same consumer in the suspected set (SS). If $CAF(SR_i, CR)$ is greater than or equal to the feedback limit (FL), then the trust model will move the suspected feedback $SF(SR, CR)$ to the collusion set (CS), or else the trust model will move this feedback to the feedback set (FS). Based on these three criteria, the trust model can either detect or ignore potential collusion feedback. The algorithm of collusion attack is calculated as follows:

Algorithm 2: Collusion Attack Algorithm

Input: FS;
Output: SS, CS;
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
elseif $T(F_n, CR)^{FS} - T(F_i, CR)^{FS} \leq TR$
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
endelseif
endif
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
elseif $T(F_n, CR)^{FS} - T(F_i, CR)^{SS} \leq TR$
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

```
15:          endelseif
16:        endif
17:      endfor
18:      for i = 1 to n
19:       CAF(SR_i, CR) ← FN(SR_i, CR)/FN(SS, CR)
20:         if CAF(SR_i, CR) ≥ FL then move SF(SR_i, CR) to CS
21:            else
22:             move SF(SR_i, CR) to FS
23:         endif
24:       endfor
25:    end procedure
```

Sybil Attack

A sybil attack is a strategic attack done by a user creating more than one account in the trust management system. The attacker utilizes these fake accounts to decrease or increase the trust value of any user in the system [35]. In the case of malicious activities, a satisfactory solution for this attack is by applying multi-identity detection (M_{id}) and sybil attack detection.

To avoid this attack, we can compare the attributes of users' credentials values in the Trust Identity Registry. R represents the identity records as a list of all users' primary identities $UP = \{PI_1, PI_2, \dots, PI_m\}$ and users' credentials $UC = \{CA_1, CA_2, \dots, CA_n\}$, where the matrix $UP \times UC$ represents the users' credentials matrix UM, which contains all user credentials CA registered in the trust model. The trust model will classify patterns in users' anonymous credentials.

To calculate the value of multi-identity detection (M_{id}) , we need to divide the appearance times (Q) of each credential attribute value (CA) by the number of identity records (R_{id}), where the frequency of any credential attribute value (CA) is represented as similar credentials attribute value spread into different identity records. M_{id} is calculated as (8).

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

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

$$Q(CA_1), Q(CA_2) \cdots Q(CA_n) \tag{8}$$

We need to apply a new procedure to allow the trust model to ignore fake trust results for a specific consumer role (CR). $RT\big(first(t)\big)$ represents the recommendation trust model for a specific consumer at a previous time. $RT\big(last(t)\big)$ represents the recommendation trust model for the same consumer at the last time the trust results were calculated without considering the proposed method. RL represents the maximum number of records that have same credential attribute. $RT(CR, first(t), last(t))$ is the factor that represents the change rate of the trust result, calculated as (9).

$$RT\big(CR, first(t), last(t)\big) = \begin{cases} \textbf{\textit{if}}\ M_{id} \geq RL \quad \textbf{\textit{then}} & RT\big(first(t)\big) = RT\big(last(t)\big) \\ else & RT\big(first(t)\big) = RT\big(last(t)\big) \end{cases} \tag{9}$$

The time factor should be used to detect the sybil attacks SA, where any change in the recommender's behavior in a short small time $RB(t)$ indicates this attack. $|N_{id}(t)|$ represents the number of identities for all recommenders who have evaluated a specific consumer in a short time $[first(t), last(t)]$. We need to measure the periodic and occasional percentage of all changes in the number of identities among the entire behavior of identities (the entire recognized identities for all recommenders that provided feedback to a specific consumer). The SA attacks factor $SA[first(t), last(t)]$ of a specified consumer in a small time is calculated as (10):

$$C(t) = \lfloor N_{id}(F(t)) \times SC \rfloor \tag{10}$$

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

where $C(t)$ represents the identities curve in a specific time, and the sybil attack curve is represented as SC, which means the allowable percentage of new identities between two specified times $N_{id}(F(t))$ and $N_{id}(L(t))$. The algorithm of the sybil attack is calculated as follows:

Algorithm 3: Sybil Attacks Algorithm

Input: RL ;
Output: $SA(F(t), L(t))$;
procedure *Sybil Attack*
for $t = 1$ *to* n
  for $CA = 1$ *to* n
    $M_{id} \leftarrow Q(CA_n)/R_{id}$
  if $M_{id} \geq RL$ ***then*** $RT\big(F(t)\big) = RT\big(F(t)\big)$
    else
      $RT\big(F(t)\big) = RT\big(L(t)\big)$
  endif
  endfor
endfor
$C(t) = \lfloor N_{id}(F(t)) \times SC \rfloor$
for $t = 1$ *to* n
  if $N_{id}\big(L(t)\big) > N_{id}\big(F(t)\big) + C(t)$
    then $SA(F(t), L(t)) = \big(N_{id}(L(t)) - N_{id}\big(F(t)\big) + C(t)\big) / |N_{id}(F(t))|$
  endif
  endfor
end procedure

Recommender Importance

Finally, we have a recommender importance, which recognizes the particular importance of some service providers in their fields of interaction and their trustworthiness as recommenders. This gives different weights to recommendations. Although most people care more about their own opinions, data owners might trust the recommendations of more experienced recommenders

than themselves. Therefore, we need to add a weight function (W^{EX}) multiplied by a number of exchange transactions (EX_i) that a recommender has had with a specific consumer, and which is dependent on the experience of recommender interactions $W(R_i, CR)$. There are an additional two factors that should be utilized. The time factor (t) is a variable used in applying the impact of the elapsed time. $last(t)$ represents the last time that the recommender transacted with the intended consumer multiplied by a weight function (W^L). $RT(CR)$ represents the trust value of a specific consumer role, where $F(R_i, CR)$ is the feedback values that are provided from trust recommenders. The value of recommendation trust is calculated as (11).

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

$$\boldsymbol{where}\ W^{EX} + W^L = 100 \tag{11}$$

$$\begin{cases} \boldsymbol{if}\ W(R_i, CR) \geq II \quad \boldsymbol{then} \quad RT\big(CR, F(t), L(t)\big) = \frac{\sum_1^n F(R_i, CR)}{N^R} \\ \boldsymbol{else} \qquad\qquad\qquad\qquad\qquad\qquad \boldsymbol{donothing} \end{cases}$$

Subjectivity

In some cases, we have more than one value of trust in the same consumer. When recommenders have different opinions and experiences, a "subjectivity" problem is often encountered in a trust management system. This refers to inaccurate user ratings, reviews, opinions, and subjective inputs to the designed trust criteria. Certain parts of TMSs are dependent on people's thinking, and those parts are related to users' behavior through people's judgments, evaluations, or are influenced by their personal feelings, beliefs, and emotions that are not precise [32,42]. The basis of this trust diversity can simply be interpreted as how much a person deals with trust criteria. This can differ from user to user depending on the users' understanding, psychological condition, and knowledge of trust. We need to build a dynamic system that can take these criteria into consideration. Trust is a subjective concept that can cause

serious problems, as the varying opinions of the users influence the precision of the interaction trust evaluation. Therefore, service providers need to view service providers' functional trust in specific consumers in accordance with their own perspectives. We need to use some parts of the solutions introduced in [34], which, in evaluating the tendency of other service providers to trust, consider user trust as an estimate rather than a value.

A straightforward execution of the computation seeks to give trust in the form of a percentile. A percentile estimate designates a recommender perception about a specific consumer in conjunction with others that have already been evaluated by the recommender. Accordingly, the SP aims to alter consumer trust as well as create the $SP's$ interaction-based functional trust about a specific consumer according to its perception. I_R represents a combination of interaction trust evaluation of recommender (R) to all consumers role CR, which it has transacted. As such, $I_R = \sum_1^n IT(R, CRs)$ represents the disposition of the recommender (R) to the collection of trust. The estimates accompanying the $SP's$ interactional trust collection is classified in an ascending sequence:

$$P = \frac{100 \times FI}{N_R^{CR} + 1}$$

where P is the consumer trust's percentile in the classified combination of I_R, $FI\ is\ first\ (IT(CR), I_R)$, which refers to the index of the first trust value in the collection of I_R, and N_R^{CR} is the number of all consumers that have transacted with $SP's$ recommender R.

$$A = \left\lfloor \frac{P \times \left(N_R^{CR} + 1\right)}{100} \right\rfloor$$

$$B = \frac{P \times (N_R^{CR} + 1)}{100} - A$$

N_R^{CR} is number of all consumers that have transacted with $SP's$ recommender (R), the parameter A is an integer, and B is a fraction ≥ 0 and < 1. S represents the interaction trust from

service providers to consumers depending on SP behavior. $I_{SP}[i]$ which represent i^{th} in I_{SP}.

$$S(R,CR)^{SP} = I_{SP}[i] + B \times (I_{SP}[i+1] - I_{SP}[i])$$

$$\begin{cases} \textbf{\textit{if}} \;\; 0 < A < N^{CR} & \textbf{\textit{then}} \quad i = A \\ \textbf{\textit{elseif}} \;\; A = 0 & \textbf{\textit{then}} \quad i = 1 \\ \textbf{\textit{elseif}} \;\; A = N^{CR} & \textbf{\textit{then}} \quad i = N^{CR} \end{cases} \tag{12}$$

To eliminate the challenge of subjectivity from the posited trust model, $S(R_i,CR)^{SP}$ shall be substituted for the $F(R_i,CR)$.

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

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

$$\begin{cases} \textbf{\textit{if}}\; W(R_i,CR) \geq II \quad \textbf{\textit{then}} \quad RT\big(CR,F(t),L(t)\big) = \frac{\sum_1^n S(R_i,CR)^{SP}}{N^R} \\ \textbf{\textit{else}} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textbf{\textit{donothing}} \end{cases} \tag{13}$$

The algorithm of subjectivity and recommender importance is calculated as follows:

Algorithm 4: Subjectivity and Recommender Importance Algorithm

Input: W^{EX}, W^{L};
Output: $RT\big(CR,F(t),L(t)\big)$;
procedure *Subjectivity*
$P \leftarrow (100 \times FI)/(N_R^{CR}+1)$
$A \leftarrow \left|\left(P \times \left(N_R^{CR}+1\right)\right)/100\right|$
$B \leftarrow \left(P \times (N_R^{CR}+1)\right)/100 - A$
if $0 < A < N^{CR}$ ***then*** $i = A$
  elseif $A = 0$ ***then*** $i = 1$
    elseif $A = N^{CR}$ ***then*** $i = N^{CR}$
      $S(R,CR)^{SP} \leftarrow I_{SP}[i] + B \times (I_{SP}[i+1] - I_{SP}[i])$
    endelseif
  endelseif
endif

12: ***end procedure***
13: ***procedure*** *Recommender Importance*
14: ***for*** $i = 1$ ***to*** n
15: ***for*** $t = 1$ *to* n
16: $W(R_i, CR) \leftarrow \left((EX_i \times W^{EX} + L_i(t) \times W^{L})/100\right)$
17: ***if*** $W(R_i, CR) \geq II$ ***then*** $RT\left(CR, F(t), L(t)\right) = S(R_i, CR)^{SP}/N^{R}$
18: ***endif***
19: ***endfor***
20: ***endfor***
21: ***end procedure***

The Proposed Joint Trust Model

To calculate the joint trust value for a specific consumer, we must coordinate the proposed inheritance interaction trust model and recommendation trust model by applying the importance of these trust models. Service providers might consider their interaction trust values as more important than the recommendation values of other service providers. In this way, we need to add two parameters W^{IIT} and W^{RT} to manage the importance of interaction trust as well as trusted recommendations associated with the consolidated trust model, where W^{IIT} represents the specific importance of the inheritance interaction trust model and W^{RT} represents the specific importance of the recommendation trust model. The joint trust value for a specific consumer is calculated as (14):

$$\boldsymbol{Trust}(\boldsymbol{CR}) = (IIT \times W^{IIT}) + (RT \times W^{RT}) \quad (14)$$

The Approach

This section illuminates the goals, architecture, and general functionality of the suggested system.

Objectives

The objectives of this study are to first conduct research and then design, implement, and monitor an access control application to validate the research hypothesis. The following are the main research aims and objectives
:

- To offer a comprehensive study on cloud computing, investigate its characteristics, and specify the design requirements for an access control solution that will be tailor-made for those characteristics.
- To design an effective and reliable architecture based on a T-RBAC access control that will improve the security of stored data in cloud storage systems.
- Introduce task-role-based access control (T-RBAC) as a new access control model integrated with a comprehensive trust model by applying a range of criteria to provide high security for cloud storage systems, while taking flexibility into consideration.
- Reduce the number of untrusted consumers with T-RBAC, and thus achieve a safer nonstop work environment.
- Build a flexible architecture based on T-RBAC wherein the owners of the resources in the system can give permission for roles or tasks, and if there is a data leak, the system will stop the task or role.
- Our trust model design involves hierarchy and inheritance in the evaluation of the trustworthiness of roles and tasks.
- The trust model provides solutions for different attacks, such as on/off attacks, collusion attacks, and sybil attacks.
- To provide high security by accounting for a wide range of criteria such as interaction importance, trust decline, task trust, conditional transfer, and subjectivity.

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. TMS will be the component of the proposed design that evaluates the extent to which providers of the cloud services are willing to trust the consumers of the cloud services. 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 present in the cloud storage systems. Figure 6 illustrates the proposed system architecture.

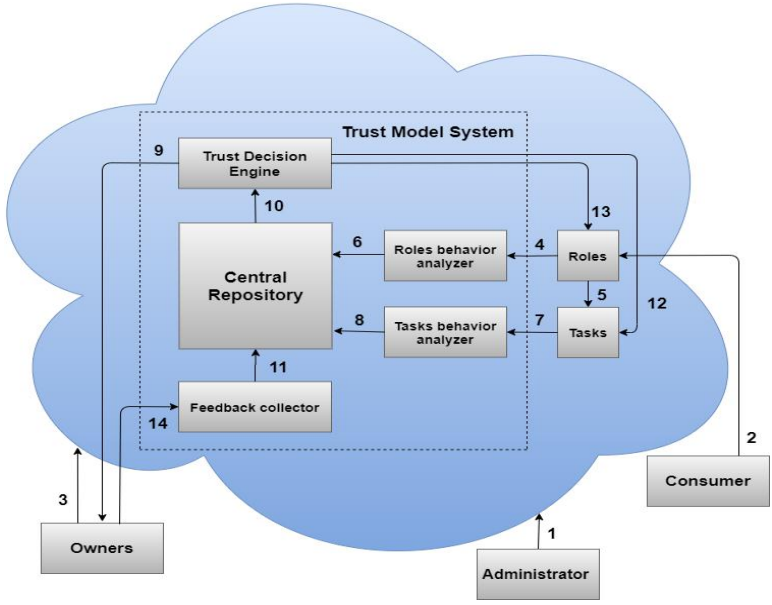


Figure 6: Trust model system architecture.

As identified in the above diagram, the trust model is composed of all the entities in the cryptographic T-RBAC system. The administrator serves as a certificate generator for the system and provides all the necessary access credentials to the consumer. Additionally, it is the role of the administrator to provide the hierarchical structure of the system. Lastly, the administrator comes up with parameters that present the position of a role in the system. The generated hierarchical roles are stored in the cloud and are available to anyone who needs them. The roles define the consumers' memberships and serve as intermediaries between the tasks and the owners. Each role contains role parameters that define the membership of the user. The tasks are the units that link the roles and the consumers. Each task has a unique task criterion that elucidates the smallest entity based on user membership. The owners consist of the groups that own the data and seek to keep their encrypted data in the cloud for easy access by consumers. They have the responsibility for determining who can access their data using the task/role-based policies. The owner of the data can also be a user, but for an owner–user to access the information stored in the system, they

must first request it of the cloud, and then the data is decrypted for them. Consumers or service users are secondary units that require accessing shared services or resources. The suggested trust model comprises five parts. The details of these parts are as follows.

Trust Management System (TMS): The Trust Management System is an additional layer of the trust model aimed at making the whole system work effectively. It is made up of different components, as discussed below.

Central Repository: This can be considered as a store of interactions. It is mainly used to store all interaction histories and trust records generated when roles and the tasks interact, which are later utilized by the Trust Decision Engine to evaluate the trust values of roles and tasks. All elements that are not in TMS are not allowed to access the central repository.

Role Behavior Analyzer: This entity is responsible for analyzing roles and functions concerning minimum trust level regulations upon accessing shared resources. It evaluates rules determined within the trust level according to the feedback from service providers and stores computed trust values in the central repository. This entity is connected to the roles entity to collect information about roles when a report on a data leak is generated. The role behavior analyzer needs to specify the identifying user and track the activities 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 if a consumer account has been involved in the incident.

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 the owners through a computed trust value, and stores this value in the central repository. It listens to channels to collect information, including reports from the tasks regarding the leakage of data and reports from the role behavior analyzer to identify

consumers' histories in regard to the stored data. The task behavior analyzer needs to specify consumers to identify and track 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 if a consumer account has been involved in the incident.

Feedback Collector: This refers to an agent that regulates the feedback from the owners to the central repository, and automatically allocates it. The feedback on roles and tasks, on the other hand, represents the trustworthiness of a consumer. The collected feedback is transmitted to the central repository for future reference. For security reasons, the feedback collector protects the integrity of the feedback on tasks and roles. The role of this component is to ensure that those who upload feedback into the system are authorized to do so. The component has the ability to identify invalid feedback and discard it from the system. Additionally, the role feedback collector collects information about data assignments for roles and tasks, after which the feedback collector updates the central repository on what has been assigned.

Trust Decision Engine: This has an evaluation role in determining the trust value of roles and tasks for the owners of the data, and for the roles and tasks entities. It collects all information about the interaction histories from the central repository as well as the outputs of the trust values of particular consumers and makes a decision for the system to respond with.

Proposed Method

- The administrator starts the system and specifies the hierarchy of roles and tasks in the system. Channel 1 facilitates the uploading of the system's generated parameters for the roles and tasks to the cloud.
- When a consumer needs access to data in the cloud, they first send an access request via Channel 2, according to their roles and tasks.

- If the request is accepted, the Roles entity relies on Channel 5 to relay the accepted request to the Tasks entity. Then, the cloud will respond by providing a user with a normal cryptographic T-RBAC plan.
- The owner can only give an encryption and uploading go-ahead via Channel 3 if they believe that the role or task can be trusted. In this process, the owner also makes the feedback collector aware of the identity of the consumers.
- In case an owner identifies leakage of their resources due to an untrusted consumer, they provide feedback about this role or task to the feedback collector via channel 14.
- If the feedback is provided by an authorized owner, the collector will forward this feedback to the central repository via Channel 11 to store each trust record and interaction history generated when the roles and tasks interacted.
- The central repository then stores those interaction histories, which are later utilized by the Trust Decision Engine to evaluate the trust value of the roles and tasks through Channel 10.
- The roles entity can accept trust evaluations about the roles from the trust management system at any time, after which the trust management system responds with the information from the trust decision engine via Channel 13.
- After receiving the information from the trust decision engine, the roles entity updates the role variables that determine a consumer's role membership in the cloud. The memberships of any malicious consumers are revoked.
- In the event of an owner providing negative feedback about a role because of resource leakage, the roles entity sends the information about the leakage to the role behavior analyzer via Channel 4.
- The tasks entity takes the trust evaluation about the tasks from the trust management system at any time, after which the trust management system responds with the information from the trust decision engine via Channel 12.

- After receving the information from the trust decision engine, the tasks section updates the task parameters that determine the consumer's task membership. The membership of any malicious consumers are revoked.
- In the event of negative feedback by an owner about a task due to resource leakage, the tasks entity sends information about the leakage to the task behavior analyzer via Channel 7.
- The analyzers then update the trust records of the roles and tasks in the central repository via Channels 6 and 8.
- When an owner needs their data to be uploaded and encrypted in the cloud, they request a trust evaluation, which is the role of the trust management system. When the query is posted to the trust management system, the system responds to the owners via Channel 9.
- The trust decision engine gives the owners the trust evaluation results for the roles and tasks. Depending on those results, the data owners can decide to whether to give consumers permission to access their resources.

Application Scenario

Real-world sensitive cloud applications are those that regulate access to the personal health records (PHRs) and electronic health records (EHRs) in e-health systems. An example of this is the "MyPHRmachines" system which was developed at the Eindhoven University. This system is an integration of PHR system and cloud technology and makes it possible to share selected health information of patients with a physician. It has been asserted that the physicians or other parties do not have to worry about incorrect storage because shared data is stored for a specified time period.

Using the above observation, we were able to apply our trust-based T-RBAC model for sharing medical data in an e-health application. For instance, consider an application scenario where personal health records are shared by a patient with a hospital department so as to acquire treatment services from the physicians of that department. Here, the patient can be considered as the data owner, the hospital department functions

as a role, the health records of patients are considered as tasks, the physician signifies the users, and the head of the hospital is the administrator.

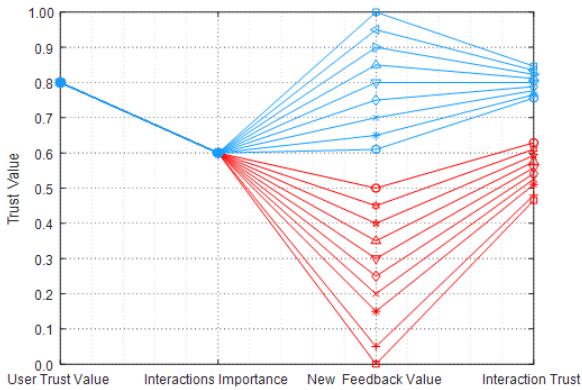
Here, we assume that the T-RBAC system is used by the application to regulate access to personal health data. Hence, when a conventional T-RBAC system is used, the health information of the patient can be exposed by a malicious physician. Revealing this private medical information to the public serves as a serious threat, particularly for prominent individuals. The real identity of the users and owners should be kept confidential as it is an issue of privacy. A pseudonym-based identity could be an appropriate solution to deal with this. However, this paper concentrates on using the trust factor to improve a T-RBAC model in the cloud environment so as to decrease the security risks and deal with issues pertaining to the trust individuals have on one another in this kind of open environment. We will now describe how the trust model put forward can be used in this scenario to decrease the threat of the personal medical records of patients being accessed illegally. We first assume that personal medical data is shared by a patient with a hospital department with the intention of acquiring treatment. The patient first examines the trust level of that particular department by considering their treatment history, if there is any, with this department and also by taking the views of other patients. Hence, the decision to share personal medical data is made by the patient on the basis of the sensitivity of their medical data, condition and the least trust value required so as to trust that hospital department (HD). After this, the HD, depending on its own reputation, hires physicians whose trust level is more than the minimum trust level required by the HD. There may also be multiple immediate successors of every HD, who serve as the hospital subdepartments (HSD) that can receive access to the medical data of the patient. All these physicians and HSDs are considered by the HD as the users. If the trust level of these users is less than the trust requirement of the HD, then their access rights will be immediately withdrawn. In addition, an entity's trust in an entity will decrease if leaked data is accessed by the entity as determined by its access time. There will be leakage of a patient's medical data when malicious

physicians, HDs or HSDs disclose this data to an unauthorized entity. This example makes it evident that it is possible to effectively use our trust model for safe storage of data on the cloud.

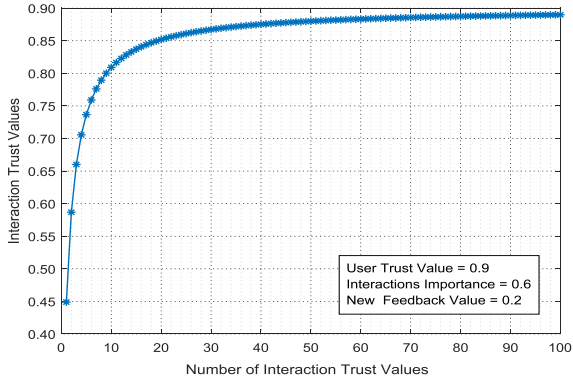
Simulation Results

We built a C#.net Windows Forms application to compare our architecture with several related works. Our trust model can find malicious consumers by applying different criteria. In this section, we will highlight the simulation results of the criteria that we have applied to block all security threats and raise the accuracy of the results.

In Figure 7a, we can see the impact of new feedback values in the interaction trust value IT for malicious consumers in red and trusted consumers in blue. As we see in Figure 7a, this impacts the penalties on the results of malicious consumers more than it raises the results of trust for good consumers; the reason behind this is that we need to stop malicious behaviors in the cloud environment. In Figure 7b, the interaction trust value IT is affected by the number of feedback items NF for each trust value. The greater the number of consumer interactions the less impact on trust results there will be, so a trusted consumer must have a lot of trusted interactions to maintain his trust value.



(**a**)



(**b**)

Figure 7: (**a**) Impact of new feedback; (**b**) impact of number of feedback.

There are two penalties in this system. First, we identified two conditions for identifying on/off attacks: when the interaction importance $II \geq 0.7$ and when $SP's$ feedback F is less than the interaction importance II. With these two conditions, the trust value is calculated with the penalty of an on/off attack P^{O^2}. The second penalty is the trust decline penalty P^{TD} which applies under the condition when the SP feedback is less than the interaction importance II. Figure 8 shows the impact of the penalty of on/off attacks and resulting trust decline in the interaction trust values.

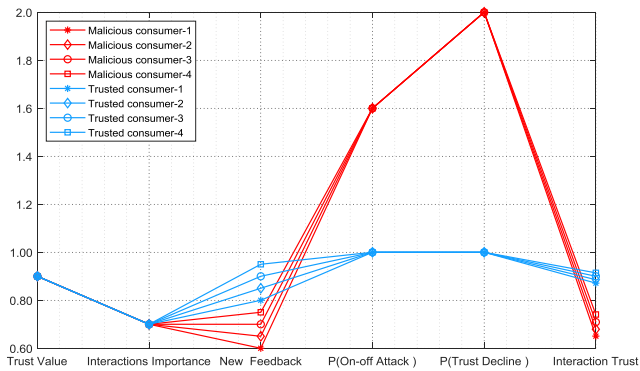


Figure 8: On/off attack and trust decline.

Figure 9 shows the impact of new feedback on the interaction trust in trusted consumers, where the system will compute the interaction trust for trusted consumers by applying different criteria to make the result of trust value very accurate.

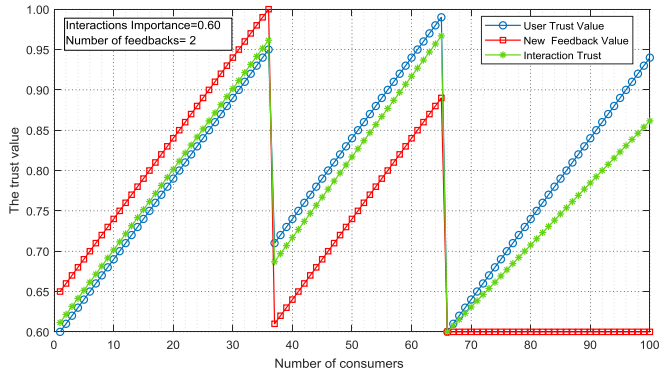


Figure 9: Interaction trust for trusted consumers.

Figure 10 shows the impact of new feedback on the interaction trust in malicious consumers, where the system will compute the interaction trust for malicious consumers by applying the penalties of on/off attack P^{O^2} and of the trust decline P^{TD}.

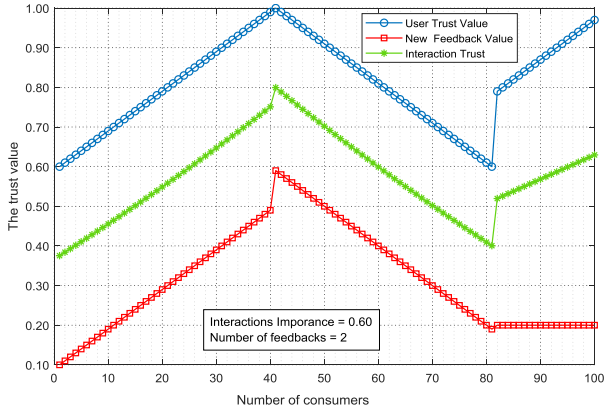


Figure 10: Interaction trust for malicious consumers.

In Figure 11, in order to compare interaction trust $IT(CR)$ with inheritance interaction trust $IIT(CR)$, the trust value of the subroles IR and the trust value of tasks $IT(T)$ will affect the trust value of the consumer role. After each interaction, the trust model will recalculate the trust value of each role inherited from subroles in the same enterprise.

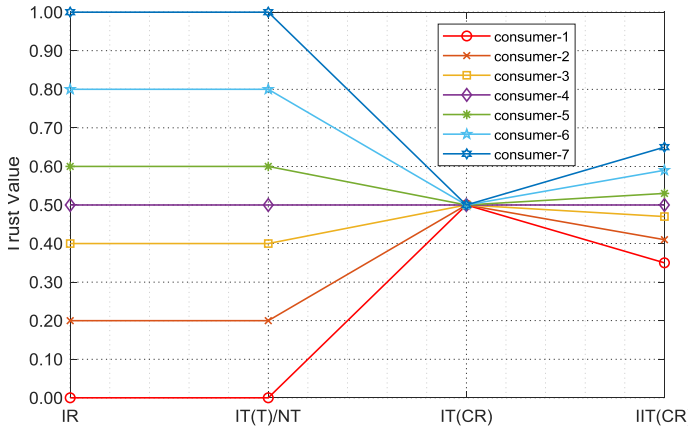


Figure 11: Interaction trust vs. inheritance interaction trust.

The trust model will alter consumer trust $IT(R_i, CR)$ according to SP perception $S(R, CR)^{SP}$. Figure 12 shows the differences between SP feedback I_{SP} and recommender feedback I_R and the impact of these values in $IT(R_i, CR)$ and $S(R, CR)^{SP}$. With these criteria, our model will remove the subjectivity and offer accurate results.

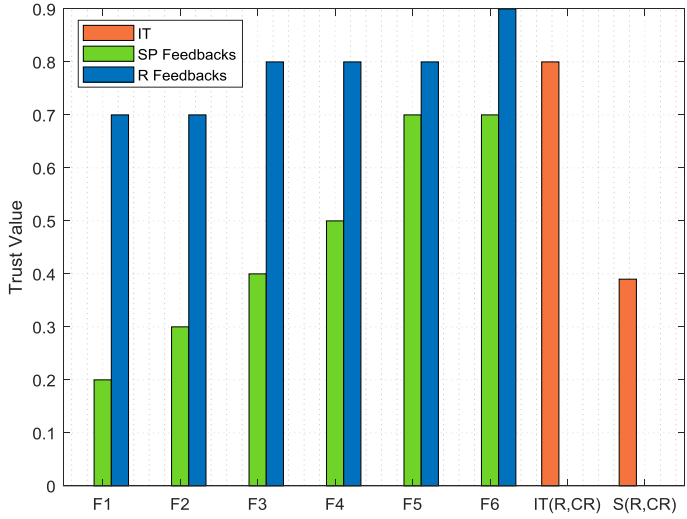


Figure 12: Subjectivity.

To detect collusion attacks, the trust model will analyze different criteria that affect the trust value of any consumer. One of these criteria is the collusion attack frequency (CAF), where the value of the feedback frequency is directly proportional to the collusion feedback and inversely proportional to the credibility of the aggregated feedback. The value of the feedback frequency depends on the number of feedback items that have been provided by any recommender and the number of feedback items in the suspected set (SS).

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

Let us assume the table above is the frequency of seven suspected recommenders, and the feedback limit $(FL) = 7\%$, The trust model will move the feedback of any suspected recommenders who have a feedback frequency greater than the feedback limit (FL) to the collusion set (CS), or else the trust model will move the feedback to the feedback set (FS). As we

see in Figure 13, there are five suspected recommenders who have feedback frequency greater than the feedback limit (FL).

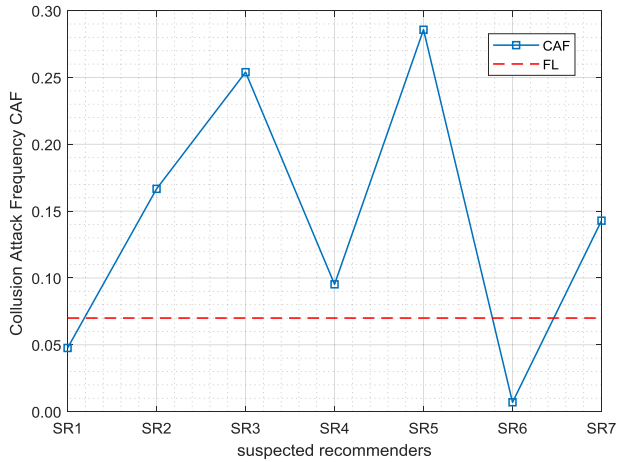


Figure 13: Collusion attacks frequency.

Finally, in order to calculate the joint trust value for specific CR, as we see in Figure 14, we have to coordinate the proposed Interaction Trust Model and Recommendation Trust Model to give the two values as one trust value $Trust(CR)$, where this value is affected by two parameters W^{IIT} and W^{RT} to manage the importance of interaction trust as well as recommendation trust.

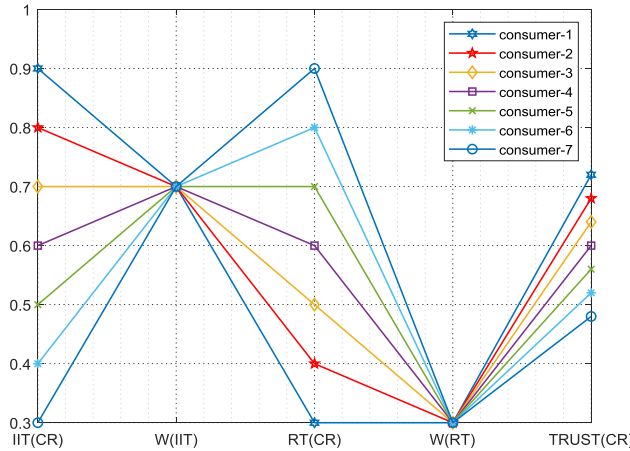


Figure 14: The proposed joint trust model.

Comparison of Security and Accuracy

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, 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 and those given in related works.

Table 2: Comparison of security and accuracy.

Addressed Metrics	[13]	[31]	[32]	[43]	[44]	OURS
Interaction importance	✗	✗	✓	✗	✗	✓
On/off attack	✓	✓	✓	✗	✗	✓
Trust decline	✗	✗	✓	✓	✗	✓
Recommender importance	✗	✗	✓	✗	✗	✓
Collusion attack	✓	✗	✓	✓	✓	✓
Collusion attack frequency	✗	✗	✗	✗	✓	✓
Subjectivity	✗	✗	✗	✗	✗	✓
Sybil attack	✓	✗	✓	✓	✗	✓

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.

Conclusions

In this paper, we have shown how a secure and flexible trust-based cloud storage system can be built wherein the owners of the resources in the system can give permission for roles and tasks, and if there is a data leak, the system will stop the task or role. We proposed a design that can decrease risk by providing high security for cloud storage systems by introducing the T-RBAC as a new model of access control integrated with a comprehensive trust model, which improves the quality of the decisions made by the cloud operators and the data owners. It was determined that trust models should involve inheritance and hierarchy in the trustworthiness evaluation of roles and tasks; this is the basis of our design for the proposed trust model based on cloud storage system. This design allows the trust model to be integrated with a system based on a cryptographic T-RBAC approach to reduce the number of untrusted consumers with T-RBAC, and thus achieve a safer nonstop work environment. Finally, in our trust model we proposed solutions for different attacks, such as on/off attacks, collusion attacks and sybil attacks. Additionally, we applied different criteria in this trust model, such as interaction importance, trust decline, task trust, conditional transfer, and subjectivity, to provide high security.

References

1. Noor TH, Sheng QZ, Bouguettaya A. Trust Management in Cloud Services. Cham: Springer. 2014.

2. Brooks TT. Cyber-Assurance for the Internet of Things. Hoboken: John Wiley & Sons. 2017.
3. Bhatt S, Patwa F, Sandhu R. An access control framework for cloud-enabled wearable internet of things. In Proceedings of the 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC). San Jose, CA, USA. 2017; 328–338.
4. Firdhous M, Ghazali O, Hassan S. Trust management in cloud computing: A critical review. arXiv 2012, arXiv:1211.3979.
5. Zhou L, Varadharajan V, Hitchens M. Trust enhanced cryptographic role-based access control for secure cloud data storage. IEEE Trans. Inf. Forensics Secur. 2015; 10: 2381–2395.
6. Bhattasali 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.
7. Marudhadevi D, Dhatchayani VN, Sriram VS. A trust evaluation model for cloud computing using service level agreement. Comput. J. 2015; 58: 2225–2232.
8. Tsai WT, 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. 2009; 1–8.
9. Varsha M, Pramod 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.
10. Zhang P, Kong Y, Zhou M. A domain partition-based trust model for unreliable clouds. IEEE Trans. Inf. Forensics Secur. 2018; 13: 2167–2178.
11. Iltaf N, Ghafoor A, Hussain M. Modeling interaction using trust and recommendation in ubiquitous computing environment. EURASIP J. Wirel. Commun. Netw. 2012; 119.
12. Tan Z, Tang Z, Li R, Sallam A, Yang L. Research on trust-based access control model in cloud computing. In Proceedings of the 2011 6th IEEE Joint International

Information Technology and Artificial Intelligence Conference. Chongqing, China. 2011; 2: 339–344.
13. 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.
14. Smari WW, Clemente P, Lalande JF. An extended attribute based access control model with trust and privacy: Application to a collaborative crisis management system. Future Gener. Comput. Syst. 2014; 31: 147–168.
15. Whitman M, Mattord HJ. Principles of information security. Boston: CENGAGE Learning. 2011; 433–469.
16. 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.
17. Yu H, Shen Z, Miao C, Leung C, Niyato D. A survey of trust and reputation management systems in wireless communications. Proc. IEEE. 2010; 98: 1755–1772.
18. Chang W, Xu F, Dou J. A Trust and Unauthorized Operation Based RBAC (TUORBAC) Model. In Proceedings of the 2012 International Conference on Control Engineering and Communication Technology. Shenyang, China. 2012; 811–814.
19. Liu K, Zhou Z, Chen Q, Yang X. Towards an attribute-based authorization model with task-role-based access control for WfMS. In Proceedings of the 2015 IEEE 16th International Conference on Communication Technology (ICCT). Hangzhou, China. 2015; 361–371.
20. Wang P, Jiang L. Task-role-based access control model in smart health-care system. In Proceedings of the MATEC Web of Conferences International Conference on Engineering Technology and Application (ICETA 2015). Xiamen, China. 2015; 22: 01011.
21. Fan YQ, Zhang YS. Trusted Access Control Model Based on Role and Task in Cloud Computing. In Proceedings of the 2015 7th International Conference on Information Technology in Medicine and Education (ITME). Huangshan, China. 2015; 710–713.
22. Huang L, Xiong Z, Wang G. A trust-role access control model facing cloud computing. In Proceedings of the 2016

35th Chinese Control Conference (CCC). Chengdu, China. 2016; 5239–5242.
23. Chakraborty S, Ray I. TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies. Lake Tahoe, CA, USA. 2006; 49–58.
24. 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. 2012; 400–404.
25. Oh S, Park S. Task–role-based access control model. Inf. Syst. 2003; 28: 533–562.
26. Zhao L, Liu S, Li J, Xu H. A dynamic access control model based on trust. In Proceedings of the 2010 the 2nd Conference on Environmental Science and Information Application Technology. Wuhan, China. 2010; 1: 548–551.
27. 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. 2013; 560–569.
28. 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.
29. Zhu C, Nicanfar H, Leung VC, Yang LT. An authenticated trust and reputation calculation and management system for cloud and sensor networks integration. IEEE Trans. Inf. Forensics Secur. 2004; 10: 118–131.
30. 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.
31. Uikey C, Bhilare DS. 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. 2017; 1–7.
32. 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. 2018; 30: 778–788.
33. Ko Ryan KL, Peter Jagadpramana, Miranda Mowbray, Siani Pearson, Markus Kirchberg, et al. "TrustCloud: A framework for accountability and trust in cloud computing." In 2011 IEEE World Congress on Services. IEEE. 2011; 584–588.
34. Hasan O, Brunie L, Pierson JM, Bertino E. Elimination of subjectivity from trust recommendation. In Proceedings of the IFIP International Conference on Trust Management. West Lafayette, IN, USA, 15–19 June 2009. Berlin: Springer. 2009; 65–80.
35. Noor TH, Sheng QZ, Alfazi A. Reputation attacks detection for effective trust assessment among cloud services. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. Melbourne, Australia. 2013; 469–476.
36. Hassan H, El-Desouky AI, Ibrahim A, El-Kenawy ESM, Arnous R. Enhanced QoS-based model for trust assessment in cloud computing environment. IEEE Access. 2020; 8: 43752–43763.
37. Han HX. Research on Adaptive Relationship between Trust and Privacy in Cloud Service. IEEE Access 2021.
38. Josang A, Ismail R. The beta reputation system. In Proceedings of the 15th Bled Electronic Commerce Conference. Bled, Slovenia. 2002; 5: 2502–2511.
39. Van Gorp P, Comuzzi M. MyPHRMachines: Lifelong personal health records in the cloud. In 2012 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS). Rome, Italy. 2012; 1–6.
40. Noor TH, Sheng QZ, Yao L, Dustdar S, Ngu AH. CloudArmor: Supporting reputation-based trust management for cloud services. IEEE Trans. Parallel Distrib. Syst. 2015; 27: 367–380.
41. Oleshchuk V. Trust-aware rbac. In International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security. St. Petersburg, Russia, 17–19 October 2012. Berlin: Springer. 2012; 97–107.

42. Zupancic E, Juric MB. TACO: a novel method for trust rating subjectivity elimination based on Trust Attitudes COmparison. Electron. Commer. Res. 2015; 15: 207–241.
43. Noor TH, Sheng QZ, Alfazi A. Detecting occasional reputation attacks on cloud services. In Proceedings of the International Conference on Web Engineering. Aalborg, Denmark, 8–12 July 2013. Berlin: Springer. 2013; 416–423.
44. Fortino G, Fotia L, Messina F, Rosaci D, Sarné GM. Trust and reputation in the internet of things: state-of-the-art and research challenges. IEEE Access. 2020; 8: 60117–60125.