Hindawi
Scientific Programming
Volume 2022, Article ID 9995023, 10 pages
https://doi.org/10.1155/2022/9995023

Research Article

Enhancing Security of Mobile Cloud Computing by Trust- and Role-Based Access Control

Arif Mohammad Abdul,[1] **Arshad Ahmad Khan Mohammad**,[1] **P. Venkat Reddy**,[1] **Praveena Nuthakki**,[2] **Rakesh Kancharla**,[3] **Rahul Joshi**,[4] **and N. Kannaiya Raja**[5]

[1]*Department of CSE, GITAM Deemed to Be University, Hyderabad 502329, India*
[2]*Department of Information Technology, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India*
[3]*Department of Computer Science and Engineering, Sasi Institute of Technology and Engineering, Andhra Pradesh, India*
[4]*Department of CSE, Symbiosis Institute of Technology, Pune, Symbiosis International (Deemed University), Pune, India*
[5]*Department of Computer Science, IOT-HH Campus Ambo University, Ambo, Ethiopia*

Correspondence should be addressed to P. Venkat Reddy; vpatloll@gitam.edu and N. Kannaiya Raja; kannaiya.raja@ambou.edu.et

Received 9 July 2022; Revised 15 August 2022; Accepted 22 August 2022; Published 13 September 2022

Academic Editor: Punit Gupta

The tremendous development in mobile technology attracts users' attention. Thus, the users are shifting from traditional computational devices to smartphones and tablets, and because of that, mobile devices have anticipated most of the global IP traffic. However, mobile device's resource-constrained behaviour cannot handle the heavy computational load. Mobile cloud computing (MCC) mitigates resource-constrained issues by enabling computing resources with minimal effort. However, providing security in MCC is an obstacle due to users' uncertain and dynamic behaviour and the explosion of online computerized data. Providing security, confidentiality, and authentication is not enough in MCC; therefore, the users need authorization. Thus, the paper designs an access control mechanism by computing the trust based on the user's uncertain behaviour. This mechanism mitigates the malicious actions caused by authenticated users. Performance results indicate that the access control mechanism accurately detects and mitigates malicious users from the MCC environment.

1. Introduction

In the early, computing takes place through desktops, but nowadays, users prefer mobile devices, like smartphones and tablets. According to a recent review [1], the IP channel is occupied with the utilization of mobile devices, and users are interested in computing through mobile devices. Still, computing over mobile devices fails to provide good performance due to its constrained nature. This constraint issue in mobile devices can be mitigated by depicting the cloud services in mobile devices and is referred to as mobile cloud computing (MCC). Cloud contains services like software, platform, and infrastructure, and it has different types like public, private, and hybrid. The main goal of the cloud is to provide a pool of resources and a computing environment to all users. So, incorporating the cloud into mobile solves the constrained resources [2]. Cloud service utilization reduces costs, processing time, and resource management [3]. Due to the elasticity behavior of the cloud, many users can participate and use the services. Cloud computing also provides a database-as-a-service for users to store its sensitive information and access it remotely. These services are conveniently offered to users by cloud service providers known as CSPs. Besides infrastructure, CSP provides deployment and delivery capabilities to users through the internet [4, 5]. The cloud enables impressive changes in our daily lives, from evolutions in health care centres to entertainment organizations. With the support of the cloud, local business becomes a global business and breaks the language barriers locally and internationally. Doctors can attend to the

patients from anywhere at any time. Scientists can predict natural disasters with accuracy. Education reaches your doorsteps more conveniently, from cities to villages.

Nowadays, the utilization of smartphones is rapidly growing, and desktops are rapidly shrinking because of increasing wireless technology. Flexible and convenient features of wireless technology occupy the global IP traffic and encourage to use of mobile devices. It becomes the user's first choice and part of daily life for working and entertainment. Computing multiple tasks over a mobile device become overhead due to limited dynamic nature, various network fluctuations, signal problem, low storage, security and confidentiality, limited power, and limited computing ability. These challenges can be overcome by combining a ubiquitous mobile network through cloud computing, which generates a new computing method, MCC. MCC is the integration of cloud computing and mobile computing, which can resolve complications by supplying a pool of services to resource-constrained devices. Mobile cloud computing fills the limitations of mobile resource gaps, such as utilization cost of bandwidth, network connectivity, energy consumption, storage capacity, computational speed, power, and security. Also, mobile device applications run in a cloud environment without computation overhead. With cloud computing support, MCC enhances mobile devices' performance [6, 7]. MCC can deploy privately, publicly, and hybrid.

Hybrid is a combination of private + public MCC. Private MCC deploys only by private organizations, and they will own all resources and not share with unauthenticated users. Public MCC deploys on the internet, and anyone can freely access all resources, including software, hardware, platforms, and infrastructure, in pay-per-usage mode [8, 9]. Private MCC is more secure than public MCC because only authenticated users can access the services of MCC. Apart from all facilities of MCC, one of the main challenges is the dynamic behavior of authenticated users. Systems face the challenge of identifying users' dynamic, uncertain behavior in MCC due to mobility, and more utilization of mobile devices. Access control and confidentiality, along with authentication, are to control and address the users' uncertain behavior [10].

In a recent survey [11], many technologies propose confidentiality by encrypting the outsourced data to assure security. The encryption method will be able to secure the data, but by enabling security, it cannot recognize the changing behavior of the user. Insider attacks, DOS attacks, and man-in-the-middle attacks are more likely to happen due to the dynamic behavior of the users. Thus, the proposed paper aims to design an access control mechanism by computing the trust based on the user's uncertain behavior, which mitigates the malicious activities of authenticated users through trust and role-based access control.

The remainder of the paper is organized as follows. Section 2 talks about the survey of various methodologies, Section 3 gives information regarding the proposed work, Section 4 discusses the computing part of the trust, Section 5 shows the result, and the further paper concludes with the proposed work.

Table 1: ASO of DAC.

Subject	Object		
	File 1	File 2	File 3
User 1	Write, read, execute	Read	Write
User 2	Read	Write, read, execute	Read
User 3	Read	Write, read, execute	Write, read, execute

2. Literature Survey

The MCC environment involves many resources and users using unique sources to perform various tasks through various end devices. Due to more users in the MCC, the resource load can increase and decrease the system's efficiency. The utilization of more resources by any user can lead to security issues. To increase the efficiency and security in MCC, it is necessary to restrict the users. So, the initial step is to prove user's identity for MCC administration.

Protecting data and resources against losing secrecy, modifications, and unavailability from adversaries. Users will not reduce the usage of resources by showing their identification. Using various resources in MCC without restriction leads to resource misuse because of users' changing nature. There is a need for a method to capture the user's nature and prevent the wrongful use of resources. In the recent survey, research on access control takes excellent attention because it is the method to control the behavior of the users. Access control determines what resource, user, how, and when to access the resources. There are different classes of access control models, and based on user requirements, these models are used. These access control models are DAC, MAC, RBAC, and ABAC.

Discretionary access control is based on object and subject, where subjects are owners, users, and objects are resources. DAC mechanism identifies the objects to be prevented from subjects by maintaining an access matrix [12, 13]. This approach is used to determine the objects to protect and identify which subject has the permissions (read or write or execute) to access the objects. Each subject has permission to access one object or more than one object.

Table 1 shows Access Matrix (A) in the form of privileges to subjects (S) on objects (O). The object owner has all permissions and control over the object(s). Generally, the owner can perform primitive operations like creating the object, assigning the object to subject, removing the object from the subject, deleting the object, changing the permissions to subject, and destroying the object. These six primitive operations are executed through commands by the object's creator. Read, write, execute commands referred to as rwx; each object is associated with the owner, user, and group. User1 has control over file1 as rwxrwxrwx, file2 as ---r-----, and file 3 as ----w---. User2 has control over file1 as---r-----, over file2 as rwxrwxrwx, and over file3 as ----w----. User3 has control over file1 as ---r-----, over file2 as rwxrwxrwx, and over file3 as rwxrwxrwx. It has significant advantages for small-scale organizations like user-friendly,

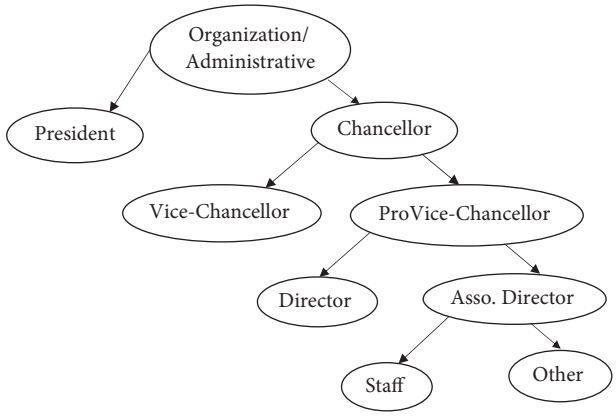

Figure 1: Hierarchical structure of mandatory access control.

flexible, and less administrative effort, but it is not suitable for large enterprises and is also less secure.

Mandatory access control (MAC) is introduced by Bell and LaPadula [14]. MAC addresses the issue of a large organization but contains excessive security because it provides security in a hierarchical pattern from a lower level to a higher level. MAC administrator decides access permissions on an object, not by subject. It defines mainly as a security framework because of its hierarchy, and here, MAC contains a set of security levels from higher-level order to lower-level order. If $L = L_1L_2L_3L_4 \ldots\ldots L_n$ is a set of security levels, then $L_i > L_j$, where L_i is higher-level security order and L_j is lower-level security order. Top-order has the highest security than the low cader security. MAC contains two security level policies: (1) simple policy and (2) restricted policy. Simple policy allows subject S to perform primitive operations like read r on object O when the security level of $L_o > L_s$. Restricted policy allows subject S to perform primitive operations like write w on object O when the security level of $L_o = L_s$.

Figure 1 shows the hierarchical structure of mandatory access control, where the root level has the highest security than the base level $L_{\text{rootlevel}} > L_{\text{firstlevel}} > L_{\text{secondlevel}} > L_{\text{baselevel}}$. The overall system is controlled by an administrator [15]. The role-based access control model (RBAC) is introduced to set the roles of users on resources and deal with the security issues [16]. Based on RBAC [16], many models have been proposed, like RBAC96 [17], ARBAC97 [18], ARBAC99 [18], ARBAC02 [19], and NIST RBAC [20]. Here, users operate the resources based on their assigned roles in the organization, and these roles can be varied accordingly [21]. RBAC administrator is engaged in keeping track and giving the roles to the users but cannot verify the user's role violations. This paper designs a novel approach to identifying users' actions by calculating their activities by incorporating trust and RBAC, namely, TRBAC. This access control mitigates the malicious actions caused by authenticated users.

The paper designs an access control mechanism by computing the trust based on the user's uncertain behavior. In literature, various mechanisms are designed to enable access control mechanisms. Still, they fail to prevent the uncertain activities caused by authenticated users [22], as one cannot predict the behavior of mobile computing users. The proposed work mitigates the malicious actions caused by authenticated users by considering users' dynamic, uncertain behavior, and roles.

3. TRBAC Approach for Mobile Cloud System

Mobile cloud computing architecture comprises the client, users, and cloud service provider (CSP). CSP provides storage services to clients with minimum managemental efforts [23]. The client has data and stores it in the cloud in a secure manner by encrypting it. Consider the client who stored the n data files in the cloud in an encrypted manner, say $F_1, F_2 \ldots\ldots\ldots\ldots\ldots F_n$. Users can retrieve data files from the cloud by proper authentication with the cloud. Further, the stored files are categorized according to their sensitivity, and users are assigned roles [24]. File access permission to users is decided based on their roles. However, the behavior of the users in the MCC environment is uncertain and leads to denial of service [25]. Thus, the paper aims to mitigate the unsteady behavior of the authenticated users by computing trust and monitoring the behavior of the users [26].

3.1. Process of Access Control in MCC. The user who wants to access the files from the cloud must authenticate himself with the cloud by providing user credentials. After verifying his submitted credentials, the user can access cloud services, like file access, request, and upload. Further, the user can access the files according to his assigned role [27]. To accomplish this, the cloud maintains a user behavior database, which consists of information about resource access permissions to users (i.e., users' roles and their access permissions) [28]. Cloud administrators are responsible for deciding the roles of individual cloud users according to their positions in the organization and assigning the resources according to their roles. If the user is authenticated and has appropriate access permissions to access the requested file, then he gets permission to access the files [29].

Further, users' uncertain behavior is continuously monitored by the system. If it finds uncertainty in the behavior, it removes the access permission to the users. The users' uncertain behavior is computed by the module called trust computing [30].

The architecture of the proposed system is shown in Figure 2, which takes the authenticated user request as input and processes the request through the trust module. The trust module consists of a user behavior database, trust computing, user's trust value, and trust updating. The user behavior database maps the user role with access permission [31]. The trust computing module computes the user's trust by continuously monitoring the users. The trust value is calculated by user actions (i.e., how many activities are correctly attempted to how many are incorrectly attempted).

Further, trust computation is evaluated based on four parameters: duplicate uploads, repeated dummy requests, malicious program uploads, and role violation. Trust value is updated for every user's calculated trust. Based on the trust value, the system decides whether the user is permitted to

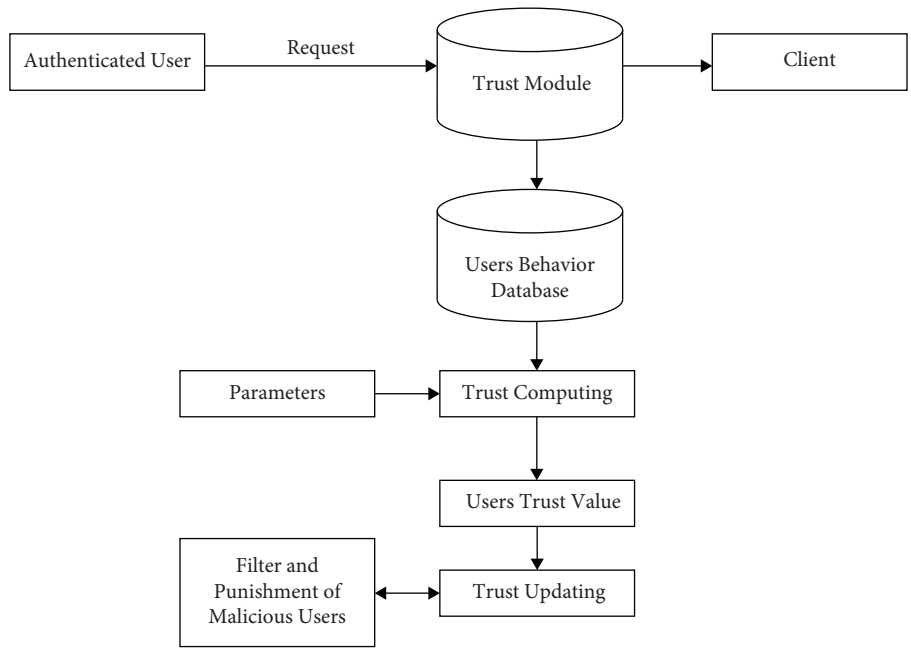

Figure 2: Architecture of TRBAC.

access the cloud or forbidden. If the user's computed trust is more than the specific level (threshold), the system decides the user is malicious. The system may occasionally choose the user as malicious due to its accidental uncertain actions. Then, the user must revoke the permission by approaching the client. Trust computing parameter values are retrieved from user's actions and explained.

3.1.1. Repeated Requests Rate (RRR). The work computes the request rate of the users by (1) to determine the uncertain behavior of the user. Usually, the user sends the query to the cloud to access the files stored in the cloud and gets responses from the query. If the user sends the repeated requests in a short time interval, this causes unusual traffic at the channel and creates a denial-of-service attack. Thus, the uncertain behavior of the user needs to be identified; otherwise, it demeans the system's performance.

$$\text{RRR} = \frac{\sum_{r=1}^{n} (\text{RepeatedRequest})_t}{\text{TotalReques}}, \quad (1)$$

where t = time, r = request, and n = number of times the same request is repeated.

If an authentic user usually behaves according to their role, for example, in one ms of the period, users send ten unique requests, RRR = 1/10. If the user acts uncertainly, for example, in one ms of the period, users send ten identical requests; then, the RRR = 10/10. The range of RRR is from 0 to 1, where the RRR value toward the one indicates malicious behavior and the RRR value toward zero means the authorized behavior.

3.1.2. Duplicate Uploads Rate (DUR). Usually, the user uploads the files to the cloud. If the user tries to upload the same file repeatedly, the cloud responds that the same file is already available, which causes unnecessary traffic. Thus, the work computes the duplicate upload rate of the users by

$$\text{DUR} = \frac{\sum_{f=1}^{n} (\sum_{u=1}^{m} \text{Repeated Uploads})}{\text{Total Request}}, \quad (2)$$

where f = file, n = number of times same file is uploaded, u = user, and m = number of times the same user is repeated.

If an authenticated user usually behaves indeed and uploads individual files, for example, the user uploads the ten individual files, then DUR = $(1 * 1)/10$. If the user is malicious and uploads the same file ten times, then DUR = $(10 * 1)/10$.

3.1.3. Role Violation Rate (RVR). Users are assigned roles, and access permissions are assigned to them according to their roles. If the user tries to access the files without permission, it is said to be a role violation and impacts system performance. This uncertain behavior of the user can be computed by

$$\text{RVR} = \frac{\sum_{d=1}^{n} (\sum_{u=1}^{m} \textit{Unautorized attempts})}{\textit{total access}}, \quad (3)$$

where d = unauthorized service, n = number of attempts to access the service, u = user, and m = number of times the same user is access

3.1.4. Malicious Program Upload Rate. The uncertain user can upload malicious codes or files to the cloud, affecting the system performance according to the malicious code. The following equation can compute these malicious actions of the users:

$$\text{MPR} = \frac{\sum_{f=1}^{n} (\text{malicious programs uploads})}{\text{total Request}}, \quad (4)$$

where f = malicious file and n = number of malicious files.

Cloud scans the files before uploading the files to the cloud database regarding malicious codes.

4. Trust Computing Criteria

One can predict the user behavior either reputed or malicious based on computed trust values. The trust value of each user is computed in each predefined period; the user accessing behavior may change the status from the past to the current period. Trust value varies between 0 and 1, where 1 is completely trusted and 0 indicates completely malicious.

To monitor the user's behavior, the cloud service provider must maintain a log file to capture the actions on each parameter. The parameters may not have the same potential to affect an organization's security, as they are computed based on the user's different uncertain behavior. So, each parameter needs to assign weights, which are initially decided by the organization based on the security requirement. Figure 3 shows the trust value computation process by maintaining the user's behavior database based on the computed trust parameters with assigned weights. Due to users' dynamic behavior, all parameters initially assigned with weights need to be dynamically reassigned with different weights based on their usage by users. The following equation is used to compute the user's trust:

$$w_i p_i;\ \text{i} = 1 \text{ to } 4, \quad (5)$$

where w = weight of parameter, p_1 = repeated requests rate (RRR), p_2 = duplicate uploads rate (DUR), p_3 = role violation rate (RVR), and p_4 = malicious programs upload rate (MPR).

If the users behave according to their assigned role, the RRR, DUR RVR, and MPR values will not change, and their values will equal zero. We have used the frequency of usage to dynamically update the weight values. And it is computed by p_i. However, the values of w_1, w_2, w_3, w_4 are not going to change as users behave according to their assigned roles. If the user does not behave according to their assigned role (in worst case), the values of RRR, DUR, RVR, and MPR become one. Further, the values of w_1, w_2, w_3, w_4 will change as users are not behaving according to their assigned roles. According to the value of p_i, value is computed.

The efficiency of a system depends on the users, who behave according to their role among the total number of users.

$$\text{Efficiency} = \frac{\text{Reputed users}}{\text{Total number of users} - \text{Malicious users}} * 100. \quad (6)$$

Equation (6) is used to identify and remove malicious users, which increases the system's efficiency.

The equations are used for evaluating the overall trust value (OTV) of a user, and the average trust value (ATV) of an individual user is shown below.

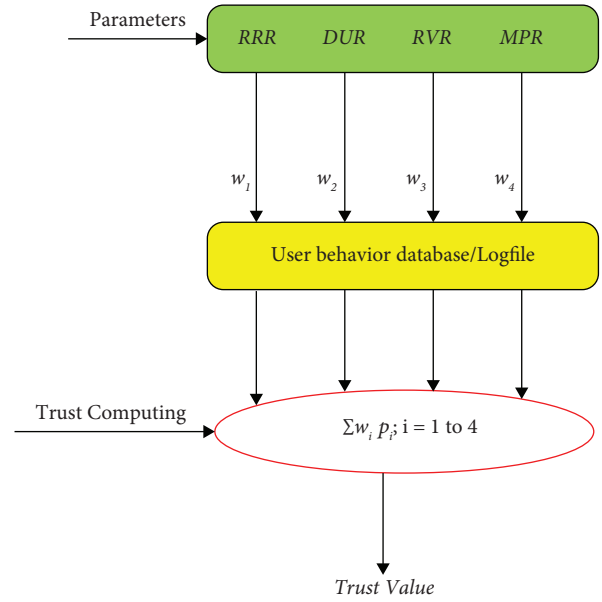

Figure 3: Trust computation criteria.

4.1. Overall Trust Value (OTV). We compute the performance of each user accessing the cloud services based on their attempts on each parameter in a unit time interval. The OTV is computed by using

$$\text{Overall Trustvalue (OTV)} = 1 - (w_1 * \text{RRR} + w_2 * \text{DUR} + w_3 * \text{RVR} + w_4 * \text{MPR}), \quad (7)$$

where $w1$, $w2$, $w3$, and $w4$ are positive weights of the trust parameters such that

$$w_1 + w_2 + w_3 + w_4 = 1. \quad (8)$$

Each weighted parameter value is varied between 0 and 1. The sum of all parameter weights must be equal to one. Initially, the weights of each parameter need to be decided by the organization; the organization has assigned each parameter with equal trust priority. Later, these parameter values are updated based on the parameter usability. The usability of each parameter is calculated by using

$$\text{Usability of } p_i = \frac{\sum p_i}{N}, \quad (9)$$

where $i = 1, 2, 3, 4$. $\sum p_i$ is the number of times parameter p_i accessed and N is the total number of parameters accessed by the users.

Initial weights of the parameters are decided based on the following criteria. The weighted parameter w_1 associated with RRR, that is, repeated request rate, and w_2 associated with DUR, that is, duplicate request rate, security point of view, both parameters are used to launch the denial-of-service attack. The weighted parameter w_3 associated with RVR, that is, role violation rate, and it is used to launch fabrication attacks. The weighted parameter w_4 is associated with MPR, that is, malicious program upload rate, and it is used to launch modification attacks and further impact data

Step 1: the organization sets the trust threshold value, i.e., 0.5, and the initial weights are $w_1 = 0.25$, $w_2 = 0.25$, $w_3 = 0.025$, and $w_4 = 0.25$.
Step 2: initially, all the users are assigned with trust equal to one. That is, all the users are completely trustworthy.
Step 3: compute each user's overall trust value OTV. Overall Trustvalue (OTV) = $1 - (w_1 * \text{RRR} + w_2 * \text{DUR} + w_3 * \text{RVR} + w_4 * \text{MPR})$.
Step 4: compute the usability of each parameter to assign priority.
Usability of $p_i = \sum p_i / \sum p_i$.
Update the weight values as $w_i = p_i$.
Step 5: compute the average trust value (ATV) by using an exponential weighted moving average method with previous trust and current computed trust.
ATV = $CV * (OTV)t_n + (1 - CV) * (\text{ATV})t_{n-1}$.
Step 6: compare the ATV of each user with the threshold value, i.e., ATV ≤ 0.5.
Step 7: if ATV < 0.5, then the malicious user or ATV ≥ 0.5 reputed user.
Step 8: return a malicious or a reputed user.

Algorithm 1: User Behaviour.

integrity. Thus, organizations need to assign the weights to each parameter initially. Further, we compute the user's average trust value (ATV) based on the exponential moving average method.

4.2. Average Trust Value (ATV). Let t_n and t_{n-1}be the time interval for the current time and previous time, respectively.

$$\text{ATV} = \text{CV} * (\text{OTV})t_n + (1 - \text{CV}) * (\text{ATV})t_{n-1}. \quad (10)$$

The user is considered malicious if his computed average trust value is less than the predefined threshold value. If his computed average trust value exceeds the predefined threshold value, then the user is considered a reputed user. The work considered the threshold value as 0.5, and it may vary according to the application and organizational sensitivity. The system finds that the user is malicious and filters those users. Unintentional user actions may cause him to be malicious, so he must contact the client, get verified, and revoke the permission.

4.3. Access Control Algorithm. Algorithm 1 explains the procedure to compute the user behavior regarding access control. The algorithm takes trust computing parameters and associated weights as input and provides the output as malicious or reputed users.

4.4. Trust Calculation. Initially, the client authenticates the user and CSP. Then, communication is established between users and CSP. The system has 100 users with reputed and malicious behaviors, one client, and one CSP. A user's reputed behavior signifies one, and a user's malicious behavior signifies zero. Based on this, significance threshold value is considered to be 0.5. The user is considered malicious if his computed average trust value is less than the predefined threshold value. If his computed average trust value exceeds the predefined threshold value, the user is considered a reputed user. The utilization of each parameter correctly considers reputed users, otherwise malicious users.

α = count of user's reputed actions

β = count of user's malicious actions. (11)

Initially, $\alpha = 0$ and $\beta = 0$, and trust (OTV) = 1, and $w_1 = 0.25$, $w_2 = 0.25$, $w_3 = .025$, $w_4 = 0.25$. The system has no new user record and cannot describe the user as malicious and reputed. Therefore, the system considers the threshold value as the basis for new users. Thus, we consider CV = 0.5.

Case 1. Initially, when user behaves according to their assigned role, that is, $\alpha = 1$ and $\beta = 0$, then $\sum w_i p_i = 0$.

$$\begin{gathered}\text{Overall Trust value (OTV)} = 1\\ -(w_1 * \text{RRR} + w_2 * \text{DUR} + w_3 * \text{RVR} + w_4 * \text{MPR})\\ \text{Overall Trust value (OTV)} = 1 - 0 = 1,\\ \text{ATV} = \text{CV} * (\text{OTV})t_n + (1 - \text{CV}) * (\text{ATV})t_{n-1}.\end{gathered} \quad (12)$$

No previous history, $(\text{ATV})t_{n-1} = 0$.

$$\begin{aligned}ATV &= 0.5 * 1 + (1 - 0.5) * 0,\\ ATV &= 0.5 + 0,\\ ATV &= 0.5.\end{aligned} \quad (13)$$

Case 2. Again, in the following action, user behaves according to their assigned role, which is $\alpha = 2$ and $\beta = 0$; then $\sum w_i p_i = 0$:

$$\begin{aligned}&\text{Overall Trust value (}OTV\text{)} = 1,\\ &ATV = 0.5 * 1 + (1 - 0.5) * 0.5,\\ &ATV = 0.5 + 0.25,\\ &ATV = 0.75.\end{aligned} \quad (14)$$

Case 3. User behaves against their assigned role; then the count ofα remains the same, i.e., $\alpha = 2$, but the count of β increases, i.e., $\beta = 1$, considered equal weight, i.e., 0.25.

Then, $\sum w_i p_i = 0.25 * 1 = 0.25$.

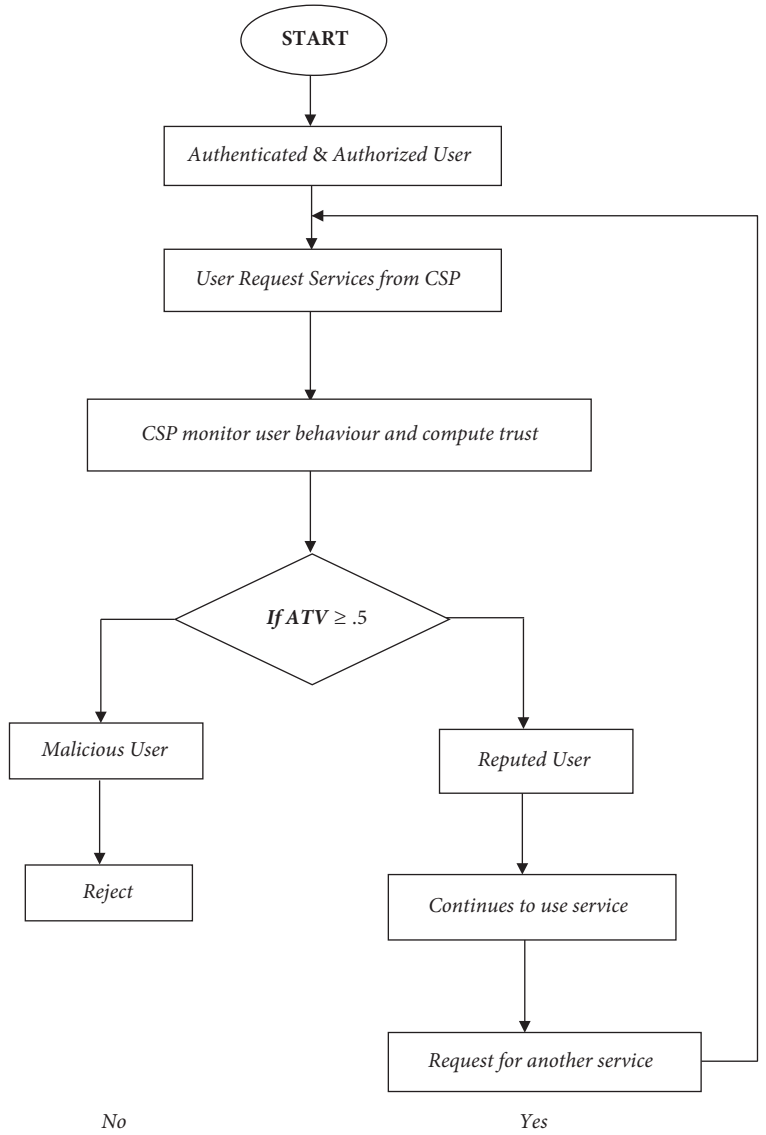

Figure 4: Flowchart for verifying the reputed user.

$$\begin{aligned}\text{Overall Trust value (OTV)} &= 1-(0.25*1+0.25*0\\ &\quad +0.25*0+0.25*0)\\ &= 1-(0.25*1)=0.75,\\ \text{ATV} &= 0.5*0.75+(1-0.5)*0.75,\\ \text{ATV} &= 0.375+0.37, \text{ATV}=0.75.\end{aligned} \tag{15}$$

Case 4. User behaves against their assigned role, and then the count ofα remains the same, i.e., $\alpha = 2$, but the count of β increases, i.e., $\beta = 2$.

Then, $\sum w_i p_i == 0.25 * 2 = 0.5$.

$$\begin{aligned}\text{Overall Trustv alue}\,(OTV) &= 0.5,\\ ATV &= 0.5*0.5+(1-0.5)*0.75,\\ ATV &= 0.25+0.375,\ ATV=0.625.\end{aligned} \tag{16}$$

Case 5. User behaves against their assigned role, and then the count ofα remains the same, i.e., $\alpha = 2$ but the count of β increases, i.e., $\beta = 3$.

$$\begin{aligned}\text{then } \sum w_i p_i &= 0.75,\\ \text{Overall Trustvalue}\,(OTV) &= 0.25,\\ ATV &= 0.4375.\end{aligned} \tag{17}$$

4.5. Access Control Flowchart. Figure 4 shows the procedure to verify and monitor user behavior. Initially, reputed users who are authorized and authenticated by the client request the services from the cloud. Then, CSP monitors their behavior and continues to provide users with the best-effort service. Computing a user's trust value helps the CSP decide whether the user is reputed or malicious.

5. Result Analysis

The efficiency analysis of the proposed work has been done on software platforms such as JDK 1.7, NetBeans-IDE, and Oracle 11 g. The efficiency analysis environment comprises the client, CSP, and 100 users, of which 95 are reputed and five are malicious. The client consists of files and uploads them to the cloud in an encrypted form. To access the files from the cloud, the user needs to authenticate themself with CSP. Further authenticated users can access the files according to their access permissions. The malicious user violates their roles and behaves uncertainly. Uncertain behavior of the users includes sending the same request repeatedly, uploading the same document frequently, and uploading malicious documents. Users' uncertain activities and role violations negatively impact the system's performance in terms of the availability of resources.

The proposed work mitigates the uncertain user's behavior and role violation by computing the users' average trust value (ATV) regarding their activities. User activities are continuously monitored and stored in a log file. The computed average trust value is compared with the threshold trust value and decides whether the user is reputed or malicious. The trust computing procedure is explained in the previous section. The weighted parameters considered to compute the trust value are $\mathbf{w_1}$, $\mathbf{w_2}$, $\mathbf{w_3}$, and $\mathbf{w_4}$. The initial value of these parameters is decided based on the sensitivity of the uncertain action.

Figure 5 shows the computed trust value for reputed action and uncertain action. The uncertain actions' trust value is below 0.5. Thus, we have taken the threshold value as 0.5.

Figure 6 shows the comparison results between the trust-based, role-based, and proposed access control mechanisms. The system efficiency for the role-based access control mechanism is 20%, as it cannot prevent the role volition users. The system efficiency for the trust-based access control mechanism is 75%, as it can avoid the role volition users but fail to mitigate uncertain actions of users. The proposed system performance is greatly improved compared to the role and trust-based access control mechanisms as it mitigates both role volition and uncertain actions of users. Figure 6 shows the overall efficiency performance comparison, indicating that the proposed work's efficiency outperforms. RBAC mechanism performance is low, as it cannot prevent uncertain activities, which are actually malicious actions of the authenticated and authorized users. TBAC mechanism performance is also low, showing that in a cloud environment, only uncertain behavior mitigation is not enough but also needs to consider role violation.

If the system operates based on only trust, it means no role is assigned to users. Users can access any parameter, which means efficiency decreases. If the system operates based on role, it means no trust between users and the system. But system restricts users and forces them to behave according to their assigned roles. If the user acts against the role, then the system cannot identify the user as malicious. These actions of users reduce efficiency. Figure 7 shows the result of a combination of trust and a role-based hybrid approach (TRBAC), which improves the system's efficiency.

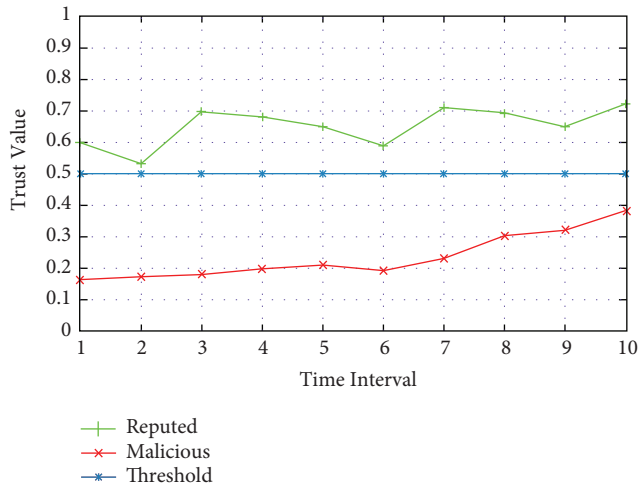

Figure 5: Trust value for reputed actions and uncertain actions.

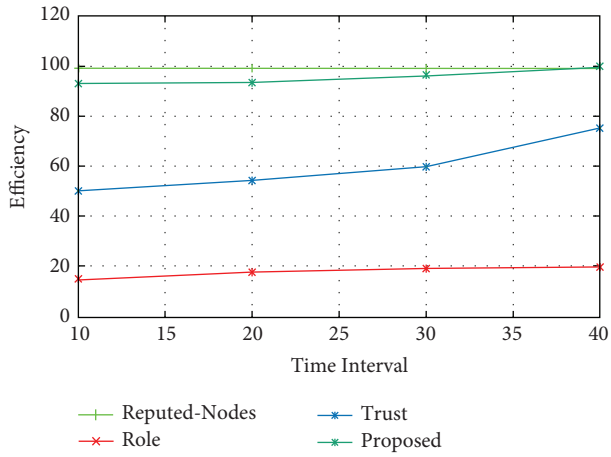

Figure 6: Comparison of efficiency proposed protocol with trust-based and role-based access control mechanisms.

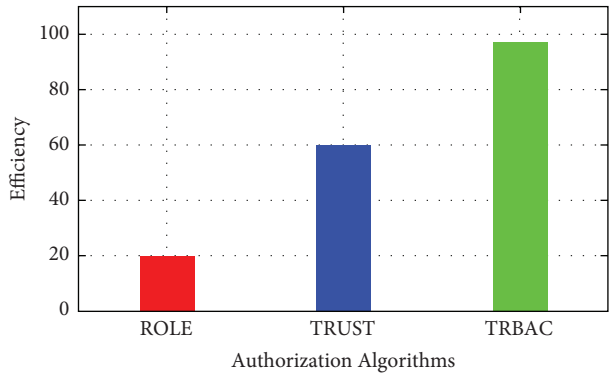

Figure 7: The average efficiency comparison proposed protocol with trust-based and role-based access control mechanisms.

Figure 8 shows the comparison between reputed users and malicious users; here 0 is considered as not malicious, and one is malicious. If the system considers all users are reputed users, then it equals 0 ratios, and if all users are malicious users, then it equals one ratio. Good utilization of

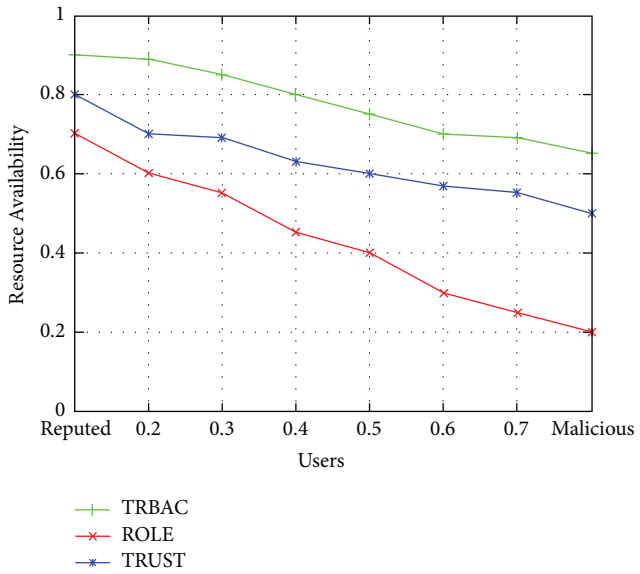

Figure 8: The resource availability comparison between the ratio of reputed and malicious users.

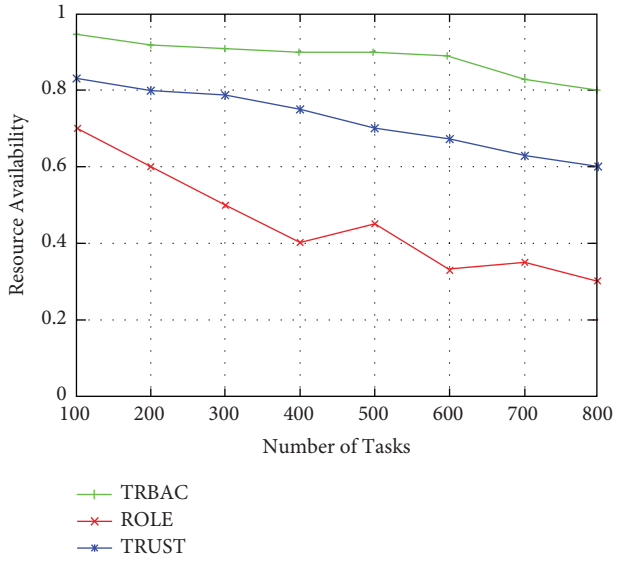

Figure 9: A number of tasks versus resource availability.

resources depends on the maximum number of reputed users. As shown in Figure 8, if the system consists only role, utilization of resources decreases 100 to 60 percent at only 20 percent of malicious users in the system. If the malicious users are increased to 80 percent, utilization of available resources is only 20 percent. Similarly, if the system consists only of trust, which is better than the alone role, utilization of resources decreases 100 to 70 percent at only 20 percent of malicious users in the system. If the malicious users are increased to 80 percent, utilization of available resources is 50 percent. Compared to the role and trust proposed work, TRBAC has better resource utilization, which means this work restricted malicious users from getting into the system. This approach consists of minimum malicious activities and maximum utilization of resources if the system consists of TRBAC, maximum utilization of resources (i.e., 93 percent at only 20 percent of malicious users in the system).

Figure 9 discusses resource availability on the number of tasks performed by the system. Task 1 is a set of 100 tasks, and task 2 is a group of 200, task 3 is a group of 300, and so on. If the tasks are increasing, availability of resources is reduced. Three scenarios are mentioned in Figure 9—role, trust, and TRBAC. If the system uses a role-based access control, then for 100 jobs, the availability of resources is 70 percent, whereas in trust-based one, 82% in the proposed work (i.e., TRBAC 93%). Compared with role-based and trust-based schemes proposed work; that is, the TRBAC scheme restricts the malicious users and maximizes the utilization of resources among reputed users. If the number of malicious users increased, it minimized the utilization of resource availability.

6. Conclusion

Authorization is one of the developing aims to implement security, authentication, and confidentiality. The only authentication is insufficient to provide security, as reputed users may behave maliciously. Access control among the users is required to mitigate malicious activities. Further, the mobile users' uncertain behavior enforces designing suitable access control mechanisms in MCC with less computational overhead.

The chapter designed the access control mechanism by calculating the trust to minimize the malicious activities caused by reputed users. Further, performance results indicate that the system's efficiency increases compared to trust and role-based access control mechanisms.

Data Availability

The data used to support the findings of this study are included within the article. Should further data or information be required, these are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] I. Ray and I. Ray, "Trust-based access control for secure cloud computing," *High-Performance Cloud Auditing and Applications*, pp. 189–213, Springer, New York, NY, USA, 2014.

[2] S. Magesh, N. V.r., P. S. Rajkumar, and S. Radha RamMohan, "Pervasive computing in the context of COVID-19 prediction with AI-based algorithms," *International Journal of Pervasive Computing and Communications*, vol. 16, no. 5, pp. 477–487, 2020.

[3] J. A. Lochhead, R. Rowland, R. Stephen, and J. Johnson, "Manikandan Subramanian, and Emmanuel Kothapally. "System and method for customer provisioning in a utility computing platform," vol. 8, p. 484, 2013.

[4] K. Nagarajan, A. Rajagopalan, S. Angalaeswari, L. Natrayan, and W. D. Mammo, "Combined economic emission dispatch of microgrid with the incorporation of renewable energy sources using improved mayfly optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 6461690, 22 pages, 2022.

[5] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[6] K. R. Vaishali, S. R. Rammohan, L. Natrayan, D. Usha, and V. R. Niveditha, "Guided container selection for data streaming through neural learning in cloud," *International Journal of System Assurance Engineering and Management*, vol. 16, pp. 1–7, 2021.

[7] Z. Zhibin and H. Dijiang, "Efficient and secure data storage operations for mobile cloud computing," in *Proceedings of the 8th International Conference and Workshop on Systems Virtualization Management*, pp. 37–45, Las Vegas, NV, USA, October 2012.

[8] M. Sookhak, H. Talebian, E. Ahmed, A. Gani, and M. K. Khan, "A review on remote data auditing in single cloud server: taxonomy and open issues," *Journal of Network and Computer Applications*, vol. 43, pp. 121–141, 2014.

[9] A. sendrayaperumal, S. Mahapatra, S. S. Parida et al., "Energy auditing for efficient planning and implementation in commercial and residential buildings," *Advances in Civil Engineering*, vol. 2021, Article ID 1908568, 10 pages, 2021.

[10] R. Buyya, "Introduction to the IEEE transactions on cloud computing," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 3–21, 2013.

[11] D. K. Jain, S. K. S. Tyagi, S. Neelakandan, M. Prakash, and L. Natrayan, "Metaheuristic optimization-based resource allocation technique for cybertwin-driven 6G on IoE environment," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4884–4892, 2022.

[12] A. M. Abdul, J. Sudarson, and M. Bal Raju, *Efficient computing over mobile cloud computing by probabilistic unigram keyword search*, vol. 12, no. 6, pp. 290–298, 2020.

[13] M. Sajjad, A. Ahmad, A. W. Malik, A. B. Altamimi, and I. Alseadoon, "Classification and mapping of adaptive security for mobile computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 814–832, 2020.

[14] S. S. Sundaram, P. Sakthi Sundaram, N. Hari Basker, and L. Natrayan, "Smart clothes with bio-sensors for ECG monitoring," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 4, pp. 298–301, 2019.

[15] J. J. Stapleton, *Book: Security without Obscurity: A Guide to Confidentiality, Authentication, and Integrity*, CRC Press, 2014.

[16] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in *Proceedings of the IEEE INFOCOM 2016- the 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.

[17] B. W. Lampson Protection, "Protection," *ACM SIGOPS - Operating Systems Review*, vol. 8, no. 1, pp. 18–24, 1974.

[18] P. Asha, L. Natrayan, B. T. Geetha et al., "IoT Enabled environmental toxicology for air pollution monitoring using AI techniques," *Environmental Research*, vol. 205, Article ID 112574, 2022.

[19] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, 1976.

[20] C. S. S. Anupama, L. Natrayan, L. Laxmi, and W. S. Abdul Rahaman, "Deep learning with backtracking search optimization-based skin lesion diagnosis model," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 1297–1313, 2021.

[21] D. E. Bell and L. J. LaPadula, *Secure Computer System:unified Exposition and MULTICS*, Technical Report ESD-TR-75-306, The MITRE Corporation, Bedford, MA, USA, 1976.

[22] L. P. Natrayan, S. Shunmuga Sundaram, and J. Elumalai, "Analyzing the Uterine physiological with MMG Signals using SVM," *International journal of pharmaceutical research*, vol. 11, no. 2, pp. 165–170, 2019.

[23] G. S. Mahmood, J. H. Dong, and A. J. Baidaa, "A secure cloud computing system by using encryption and access control model," *Journal of Information Processing Systems*, vol. 15, no. 3, pp. 538–549, 2019.

[24] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[25] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 model for role-based administration of roles," *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 105–135, 1999.

[26] R. Sandhu and Q. Munawer, "The ARBAC99 model for administration of roles," in *Proceedings of the 15th Annual Computer Security Applications Conference*, pp. 229–238, New York, NY, USA, December 1999.

[27] S. Oh, R. Sandhu, and X. Zhang, "An effective role administration model using organization structure," *ACM Transactions on Information and System Security*, vol. 9, no. 2, pp. 113–137, 2006.

[28] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[29] G. Kanimozhi, L. Natrayan, S. Angalaeswari, and P. Paramasivam, "An effective charger for plug-in hybrid electric vehicles (PHEV) with an enhanced PFC rectifier and ZVS-ZCS DC/DC high-frequency converter," *Journal of Advanced Transportation*, vol. 2022, Article ID 7840102, 14 pages, 2022.

[30] Q. Kharma, T. Nedal, S. Qusai, and H. Mohammad, "Secure cloud-mediator architecture for mobile-government using RBAC and DUKPT," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, pp. 44–60, 2020.

[31] A. A. K. Mohammad, M. M. Ali, and S. Vemuru, "Providing security towards the MANETs based on chaotic maps and its performance," in *Microelectronics, Electromagnetics and Telecommunications*, pp. 145–152, Springer, Singapore, 2019.