Computers, Materials & Continua
DOI:10.32604/cmc.2021.018993
Article

Tech Science Press

Enhanced Trust Based Access Control for Multi-Cloud Environment

N. R. Rejin Paul[1,*] and D. Paul Raj[2]

[1]R M K College of Engineering and Technology, Puduvoyal, Gummidipoondi, 601206, India
[2]R M D Engineering College, Kavaraipettai, Gummidipoondi, 601206, India
[*]Corresponding Author: N. R. Rejin Paul. Email: nr.rejinpaulphd@gmail.com
Received: 28 March 2021; Accepted: 29 April 2021

Abstract: Security is an essential part of the cloud environment. For ensuring the security of the data being communicated to and from the cloud server, a significant parameter called trust was introduced. Trust-based security played a vital role in ensuring that the communication between cloud users and service providers remained unadulterated and authentic. In most cloud-based data distribution environments, emphasis is placed on accepting trusted client users' requests, but the cloud servers' integrity is seldom verified. This paper designs a trust-based access control model based on user and server characteristics in a multi-cloud environment to address this issue. The proposed methodology consists of data encryption using Cyclic Shift Transposition Algorithm and trust-based access control method. In this trust-based access control mechanism framework, trust values are assigned to cloud users using direct trust degrees. The direct trust degree is estimated based on the following metrics: success and failure rate of interactions, service satisfaction index, and dishonesty level. In addition to this, trust values are assigned to cloud servers based on the metrics: server load, service rejection rate, and service access delay. The role-Based Access control policy of each user is modified based on his trust level. If the server fails to meet the minimum trust level, then another suitable server will be selected. The proposed system is found to outperform other existing systems in a multi-cloud environment.

Keywords: Cloud computing; trust; access control; cloud service provider; cloud data user; CSTA

1 Introduction

Cloud computing is an open standard model, which can empower universal computing and pride request-based access to a pool of configurable computing devices. It is a promising cutting-edge computing worldview that fundamentally depends on innovations, for example, virtualization, utility computing, Service Oriented Architecture, etc. It is Internet-driven and gives the entirety of its assets as administrations, for example, stockpiling, calculation and correspondence. It is a one-of-a-kind mix of capacities and development innovations. It needs negligible administration exertion from specialist co-ops and conveys versatile and dynamic foundation, remote access, and

use control and evaluating [1]. With the rise of enterprise-level cloud computing activities, cloud computing has become more of a reality than just an idea [2].

Since a cloud domain includes different assets having numerous customers cooperating in difficult habits, appropriate access control to these assets is essential [3]. The idea of trust will compare to many relations among substances interested in a social procedure. These relations all around include two substances, the specialist co-op is named trustor, and the subject expecting access to the administrations of trustor is named trustee. Trust foundation depends on the information or encounters gathered from the past associations of substances. Since in any relationship, trust precedes approval, there is a compelling need for research towards a trust-based security environment and instrumentation inside cloud condition. Nonetheless, there exist not many trusts the board models in cloud computing condition [4].

Both the cloud clients and specialist co-ops ought to have a confided relationship. (i.e.,) Both elements ought to be trusted. Be that as it may, many works consider it confided in clients as it were. In [5], different degrees of trust is viewed as, for example, client's trust, supplier's trust, gadget's trust dependent on which various jobs are allocated. Be that as it may, it neglects to rattle off the components considered for trust assessment. In [6], the intra-area and between-space trusts are resolved depending on the client connection and administration fulfillment. In any case, it ought to consider the server's remaining burden and disappointment probabilities. In [7], the client's criticism, the server's outstanding task at hand, and the number of solicitation dismissals made by the server are considered for trust estimation. However, it neglects to consider the trust value of cloud clients.

To overcome the issues mentioned above, we propose a trust-based access control framework for a multi-cloud environment. The proposed ACM is designed based on server-level trust value and user-level trust value. Using this method, we can avoid unauthorized user login. To further tighten the security to avoid data leakage, the stored data is encrypted with the CSTA algorithm's help. In this way, we can avoid sensitive information loss and un-authorized user login processes. The significant contribution of the proposed methodology is listed below;

- A novel user-server trust-based access control mechanism is proposed to overcome the intrusion of the unauthorized user.
- CSTA algorithm is used to encrypt data to avoid data leakage on communications.
- The proposed system is compared with existing systems in terms of various evaluation metrics like encryption, decryption time, running time, memory consumed, success rate, and access delay.

2 Related Works

Many researchers have developed various access control mechanisms on the cloud. Indrajit Ray has planned a trust-based access control model in such access control models, which characterizes many components and relations between those components with trust-based limitations characterized on these relations. To support access verification of applications and data in cloud infrastructure, a trusted cloud client was set up to run diagnostic tests using the cryptographic hash-based test. Additionally, Yuyu Bie et al. have planned to provide a trust-based access control instrument for a multi-space cloud environment. Right off the bat, trust value is introduced among clients and cloud storage. Also, the contrast between intra-area trust and intra-space trust is analyzed. Besides, a role-based access control system joined with trust degree in multi-area is

introduced. Access control in neighborhood space applies RBAC model joined with trust degree, while in multi-area it contains the origination of role interpretation.

Another security system was proposed around the same time [8], whereinto maintain the predefined trust levels, load offsetting observation on security conditions with proactive activities are monitored. A notoriety-based trust model assessed specialist organizations' notoriety [9] by utilizing a trust assessment calculation that will take clients' input, server dismissal rate, and server outstanding tasks at hand into thought. The trials show that the trust result is progressively effective.

Another trust-based model for security participation [10], named DBTEC, was built to advance vehicles' security collaboration in VCC. This joins the circuitous trust estimation in the Public security board and the immediate trust estimation in the Private security board to process the trust estimation of vehicles while picking agreeable accomplices; a reliable participation way producing a plan is proposed to guarantee the wellbeing of collaboration and increment the participation finishing rates in VCC. Correspondingly, Tawalbeh et al. [11] have read the cloudlet engineering for MCC. They discovered that utilizing this model improved the exhibition of numerous applications and lessened the system's idleness. Numerous QoS components like accessibility, adaptability, and throughput were improved when utilizing the cloudlet model over non-cloudlet cloud engineering. Likewise, they introduced secure usage and model for the cloudlet MCC model utilizing the dynamic trust appointment procedure to give better security and protection to the client's information in MCC.

In addition to all the trust-based systems under review, the encryption mechanism employed for data access plays a key role in deciding the integrity of the data and the system's security in use. The initial models used are a modified Searchable Symmetric Encryption (SSE) algorithm called Two-Round Searchable Encryption (TRSE) [12] that helps in avoiding data leakage with major cipher text operation on the server-side. With the evolution on peer–to–peer cloud storage (P2P), the idea of Ciphertext Policy Attribute-Based Encryption (CP-ABE) with proxy re-encryption scheme [13,14] gained popularity as they provide secure and efficient access control.

As security in data storage became the priority, the validity of data came into the limelight. To ensure the credibility of the data accessed, Identity based Encryption on Revocable Storage (RS–IBE) elements was suggested [15]. This forward/backward security of cyphertext showed better performance in the case of efficiency and functionality. However, [16] questioned this algorithm's correctness and suggested using self-updatable encryption that could boost the RS-IBE algorithm's performance.

As data protection became feasible and quite frankly mandatory in all cloud storage access, the degree of usability was the next performance metric that gained attention. Hence a data protection mechanism with a self-contained module called Role-based access control enhanced using data-centric attribute-based encryption (DC–RBAC) was popularized [17]. The encryption was further strengthened by adding a trust value calculated using a Fuzzy Analytic Hierarchy Process (FAHP) [18] that provided better granularity and flexibility. Thus, add a trust value seemed like the best option to provide flexible yet best data security.

The most common encryption algorithm for access control was Ciphertext Policy-Driven Attribute-Based Encryption (CP-ABE). They form the basis for many mobile multimedia data sharing [19,20]. In addition to other attributes, sometimes data creation and data access are also considered an attribute that helps in dual data access control and data integrity verifiability, thereby strengthening the CP-ABE one parameter at a time [21]. After a thorough examination

of different encryption algorithms and various access control models, it is clear that not many have dealt with trust values as the main factor, and even then, the concept of cyclic shifting is seldom heard. So, we have decided to proceed with our work along the tracks of cyclic shift-based encryption for securing data and a trust-based model for access control and compared its results with the existing models.

3 System Model

The outline of the system proposed is shown in Fig. 1. The system consists of three entities, namely, data owner (DO), Data server (DS), and Cloud Service Provider (CSP).

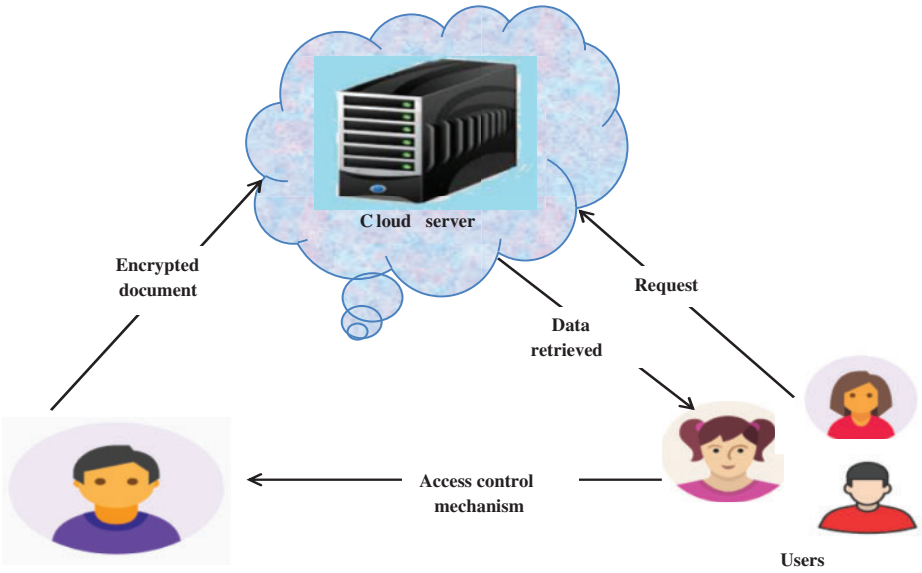

Figure 1: System model

- Data Owner (DO)–DO is responsible for collecting the data from different resources. For security reasons and to avoid data loss, the collected data is encrypted before storage by the data owner.
- Cloud Service Provider (CSP)–CSP manages and stores data in the cloud. When the user wants data from CSP, a request is sent. If the request is valid, the CSP sends back the requested data.
- Data server (DS)–DS is responsible for checking the request sent to CSP and sending the secure data back to the user. This is done by checking the checksum in ACM.

4 Proposed Methodology

The purpose of the proposed methodology is to securely transmit or store the data on the cloud using a trust-based access control mechanism with the CSTA approach. As to the growing size of data, industries now prefer cloud-based data storage. Due to the enormous amount of data floating around us, single-cloud-based storage is avoided as they face many issues like limited free storage, vendor lock-in, and data loss. Keeping these issues in mind, multi-cloud storage services are preferred as they provide a single platform configuration for multiple cloud storage services.

Moreover, security is a key issue in this multi-cloud storage system. To avoid facing many security issues, in this paper, multi-level security frameworks are introduced. In the first level, data are encrypted using a novel algorithm called the CSTA algorithm and the second one is the trust-based access control mechanism. The former provides data security while the latter is to avoid the unauthorized user login process.

4.1 Data Security Using Cyclic Shift Transposition Algorithm

One of the key issues of the cloud is security. Due to security issues, companies are reluctant to store cloud data. Therefore, data confidentiality is an essential task in the cloud. To avoid this problem, CSDA has introduced an efficient security mechanism in this paper. The proposed CSTA algorithm does not depend on any centralized authority like the central management system. Using this can avoid major related issues. This method consists of two steps, namely, encryption and decryption. The encryption process is used to convert the original data into ciphertext. And decryption is the process of converting ciphertext into original data without losing original information. The step-by-step process of CSTA is explained below;

4.1.1 Encryption Process

Encryption is the process of original hiding information using secret codes. The encryption process is done using the CSTA algorithm that performs row and column transition-based partitioning and primary and secondary diagonal transformation. The encryption process is explained in detail in the following steps

Step 1: Let the input document containing data be D_i. To start the process, the input document D_i is converted into $N \times N$ matrix format.

Step 2: After that, the Shift Column (SC) operation is applied to the $N \times N$ matrix. The SC calculation is given in Eq. 1.

$$D'_{r,c} = D_{r+shift(r,M_b) \bmod M_{b,c}} \tag{1}$$

where $shift(r, M_b)$ depends only on the key value. The key can take up any value between 0 and 9. It denotes the number of elements that need to be shifted, and the mod represents the arithmetic function.

Step 3: After the SC operation, the Shift Row (SR) operation is applied. The SR calculation is given in Eq. (2).

$$D'_{r,c} = D_{r,c+shift(r,M_b) \bmod M_b,\ c} \tag{2}$$

Step 4: Then, we performed Diagonal Shift (DS) operation. In DS operation, the diagonal elements from top left to right bottom are shifted. The DS function can be written as following, Eq. (3).

$$D'_{r,c} = D_{r+shift(r,M_b) \bmod M_{b,c+shift(r,M_b) \bmod M_b}} \tag{3}$$

Step 5: Then, again, we perform the DS operation in the order given. The function can be written as Eq. (4);

$$D'_{r,c} = D_{(r-1) \bmod M_{b,c.}} \tag{4}$$

Step 6: Then, the output is derived from the given values, Eq. (5).

$$D'_{r,c} = D_{(c+(M_{b-1})),c.} \tag{5}$$

Step 7: Now, the output is converted to ASCII format to obtain encrypted text.

Step 8: Finally, we compute the hash value with the timestamp to be sent along with the encrypted data and store it in the cloud.

4.1.2 Decryption Process

Decryption is the process of reversing the ciphertext of the encryption process to its original form. In general, all data sent from cloud servers are encrypted before transmission to the cloud user, who then decrypts it to retrieve the original message. The decryption process is explained below;

Step 1: Initially, the hash value is calculated along with a timestamp from the encrypted data, and this hash value and timestamp are transferred to the receiver.

Step 2: Then, encrypted data is converted into ASCII format.

Step 3: After that, the SR operation is applied into a specific order.

Step 4: Then, we applied SC operation in a specific order

Step 5: Then, the DS operation is applied diagonally.

Step 6: After that, again DS operation is applied to the output in a specific order.

Step 7: Finally, we obtain the decrypted output.

4.2 Trust-Based Access Control Framework Based on User and Server Characteristics

Malicious users have been a complicated problem in a cloud setup that jeopardizes the safety of communicating sensitive data. Access control models (ACM) play a vital role in implementing security for these sensitive data. Access control is checking the requests sent by every user and scrutinizing the legitimate ones from it. The request is either granted or denied based on the pre-defined control policies framed by different models. There are some ACMs proposed by various authors before. But due to the constantly revamping security needs and non-predictive user behaviors that make the sensitive data vulnerable, the models face many threats and challenges and require constant upgradation. To overcome this obstacle, A Trust-Based Access Control (TBAC) Framework is introduced. TBAC is designed based on the characteristics of the user and server. Fig. 2 shows the architecture of the TBAC framework.

In this framework, trust values are assigned to cloud users from direct and recommendation trust degrees. The trust degree is calculated by combining metrics such as number of successful interactions, index of service satisfaction, dishonesty for a user (access violations), number of failed interactions. Similarly, trust values are assigned to cloud servers based on the following metrics: server load, the number of rejected requests, and service access delay. Then RBAC is assigned to each cloud user after checking his current trust value. If his current trust value does not meet the necessary conditions pre-defined for this system, his request will be denied. Similarly, if the service provider fails to meet the minimum requirements, then another suitable provider will be selected.

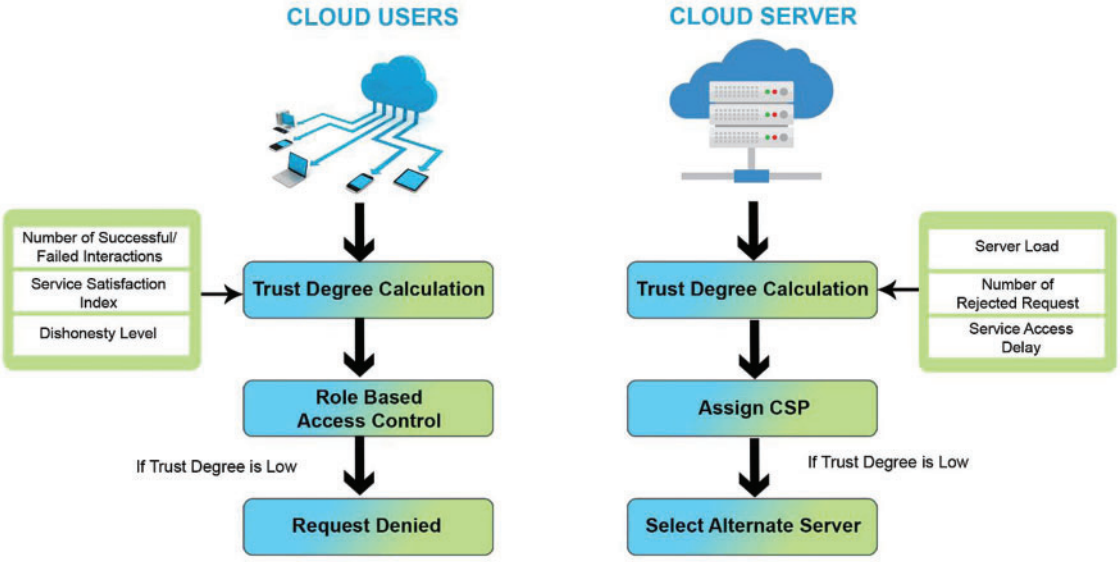

Figure 2: Architecture of TBAC-USC framework

4.2.1 Estimation of Trust Degree for Users (TR_{user})

In the given domain, an interaction trust value TR_{int} is assigned when a user U_j completes the interaction with another user U_i. The value is determined based on the number of successful or failed transactions. The success rate after k interactions is given by Eq. (6).

$$SUC\_RATE_K = \frac{No\_Suc\_Int}{Tot\_Int} \tag{6}$$

where, No_Suc_Int and Tot_Int represent the number of successful interactions and total interactions.

Similarly, the failure rate after k interactions is given by Eq. (7).

$$FAIL\_RATE_k = \frac{No\_Fail\_Int}{Tot\_Int} \tag{7}$$

where, No_Fail_Int and Tot_Int represent the number of failed interactions and total interactions.

The interaction trust value of U_i assigned by U_j after the k^{th} interaction is given by Eq. (8).

$$TR_{\text{int}}\left(U_i, U_j\right)^k = \alpha.\left[SUC\_RATE_k - FAIL\_RATE_k\right] \tag{8}$$

A service satisfaction index SSindex is assigned in the operation domain when an entity obtains multiple services from another entity. After k interactions, the SS_{index} of user U_i assigned by user U_j is given by Eq. (9).

$$SS_{index}\left(U_i, U_j\right)^k = \beta.SS_{index}\left(U_i, U_j\right)^{k-1} + (1-\beta).TR_{\text{int}}\left(U_i, U_j\right)^k \tag{9}$$

The service provider is capable of tracking the access privileges of any malicious user U_i. Then, the dishonest level (DH) of user U_i is calculated using Eq. (10).

$$DH\ (U_i) = \gamma .No\_AV \tag{10}$$

where NO_AV represents the number of access violations performed by U_i.

Then, the total trust degree of user U_i over all other users U_j, $j = 1, 2, \ldots, n$ can be derived as Eq. (11)

$$TR_{user}\ (U_j) = \sum_{j=1}^{n} TR_{\text{int}}\ (U_i,\ U_j)^k + \sum_{j=1}^{n} SS_{index}\ (U_i,\ U_j)^k - DH(U_i) \tag{11}$$

4.2.2 Estimation of Trust Degree for Cloud Server (TR_S)

Initially, the server will be ideal. When a user sends service requests, all requests will be in the queue to get service. The server load is calculated using relative transaction time T_{trans} of service requests. The server's delay time is noted while calculating the trust evaluation of each server. The Server Load (SL) with m user's request is then given by Eq. (12).

$$SL = \frac{\sum_{i=1}^{m} T_{trans}\ (U_i)}{m} \tag{12}$$

where $T_{trans}(U_i)$ represents the transaction time of service request of user U_i.

The request rejection rate RR_{rate} of a server is based on the number of service requests rejected by the server.

$$RR_{rate} = \frac{No\_\text{Re}j\_\text{Re}q}{Tot\_\text{Re}q} \tag{13}$$

where $No\_Re_j\_Re_q$ represents the number of service requests of all m users.

The service access delay D_{sa} is the time taken between issuing the service request to the CSP and obtaining access to the requested service, Eq. (14).

$$D_{sa} = \frac{\sum_{i=1}^{m} \left[T_{rep}\ (U_i) - T_{req}\ (U_i)\right]}{m} \tag{14}$$

T_{rep} and T_{req} are when the service reply is received and when the service request is issued, respectively, by users U_i, $i = 1, 2, \ldots, m$.

Then the total trust degree of server S_i can be derived as Eq. (15)

$$TR\,S_i = w_1 .SL + w_2 .RR_{rate} + w_3 .D_{sa} \tag{15}$$

where w_1,w_2 and w_3 are weight values ranging from [0, 1].

4.3 Trust Based Access Control

A type of access control called Role-Based Access Control (RBAC) is predominantly used within organizations for administrating and controlling the type of concessions of communications. This is achieved using permissions on functional roles rather than individual identities. The access decisions depend on the users' roles within the organization, which furthers decides the users' membership. This paper combines trust parameters and access control models to provide a trust-based access control framework in the cloud computing environment. RBAC's trust degree

reflects cloud users' fundamental property, servers, and the transmitted resources. The Trusted Authority Centre (TAC) is responsible for access control authentication and trust management in the cloud computing environment.

In this framework, users can obtain their access rights initially based on their roles, but they need to possess the required trust degree to use the assigned rights. When a cloud user requests access to a cloud service or resource, the TAC will check whether the user's trust level matches the threshold defined by the system. If the user's request for access is authorized, TAC provides a certificate to the requested user to obtain permission to use the access rights corresponding to his role. The access control for cloud users based on their trust degree levels is shown in Tab. 1. The threshold values T1, T2, and T3, are fixed based on the TR_{user} trust degree levels.

Table 1: Access control levels for different users based on trust degree

Trust levels	ACL
$TR_{user} > T3$	Allow full access (administrative)
$TR_{user} >= T2$ *and* $< T3$	View/Edit
$TR_{user} >= T1$ *and* $< T2$	Only view
$TR_{user} < T1$	Deny access

The following algorithm summarizes the steps involved in the Trust based access control for cloud users.

Algorithm 1: Algorithm for RBAC

Let RU be the set of users registered in a CSP
Let U $\subset$ SU be the sub-set of cloud users who need to access a service.

1. Start
2. For Each $U_j \in$ RU
3. U_j submits its user ID, password, and role to CSP
4. U_j obtains its access rights based on its role
5. End For
6. For Each $U_i \in$ U
7. U_i submits its access request with user id, password, and requested resources
8. The CSP authenticates U_i
9. If U_i is a registered user, Then
10. CSP forwards the access request to TAC
11. TAC sends trust request for U_i to all U_j, $j \neq i$
12. Uj send $TR_{user}(U_i, U_j)$ to TAC
13. TAC computes $TR_{user}(U_i)$ using Eq. (6)
14. TAC assigns ACL based on the trust levels listed in Tab. 1
15. TAC returns the ACL to CSP
16. CSP modifies the access control policy of U_i based on ACL
17. CSP provides the requested resource to U_i by applying ACL
18. Else

(Continued)

19. The access request is rejected
20. End If
21. End For
22. Stop

According to Algorithm-1, a user with a minor trust degree could not access the resources, and a user with the highest trust could perform all functions on the resources.

4.4 Server Selection Based on Trust Level

Once a user's service request is received, the CSP will allocate a server based on its trust degree. The following algorithm summarizes the steps involved in the server selection process:

Algorithm 2: Algorithm for CSP

Let $\{RS\} = \{S_1, S_2, \ldots.. S_r\}$ be the set of registered servers in a CSP.
Let U_{ij} be the set of users U_i who accessed the services from server S_j
Let TRS_{th} be the threshold value of the trust degree of a server

1. Start
2. For each user U_{ij}
3. If U_{ij} completes its service, then
4. U_{ij} feedbacks its service completion status [accepted or rejected] and D_{sa} to TAC
5. End If
6. End For
7. TAC obtains SL (S_j) from CSP
8. The feedback of U_{ij}, TAC computes RR_{rate} using Eq. (8)
9. TAC then calculates TRS_j using Eq. (10)
10. If $TRS_j < TRS_{th}$, then
11. TRS_j is a trusted server
12. Else
13. TRS_j is not trusted
14. TAC send notification about TRS_i to CSP
15. CSP shifts all the resources and services to S_k, $k \neq j$
16. End If
17. Stop

In Algorithm 2, if the trust degree of a server S_j is below the threshold TRS_{th}, then that server will be removed by the CSP. All the resources and services corresponding to S_j will be shifted to another server whose trust degree is above TRS_{th}. The subsequent service requests from the users will then be submitted to this new server.

5 Experimental Results

To validate the TBAC-CSC framework presented in this paper, the implementation is done using a java-based CP-ABE toolkit and the Java Pairing-Based Cryptography library (JPBC). The proposed TBAC-CSC framework's performance has been compared with the traditional RBAC model and TBAC policy scheme. The experiments are carried out using Java on the system with an Intel Core processor at 3.00 GHz and 4 GB RAM running Windows 7 Ultimate. The results are taken as an average of 10 trial data exchanges. Tab. 2 shows the experimental settings used

in the simulation. The available number of servers registered users and malicious users were fixed on values from the previous works studied to compare them.

Table 2: Experimental settings

Settings (#)	Value
Servers	4
Registered users	10
Malicious users	4
Requested services	2–10
The average size of each request	500 to 1000 *kb*

The main objective is to securely store and transmit the data on multi-cloud using a trust-based access control mechanism with the CSTA approach. In this paper, ACM is utilized not to allow any unauthorized person to access the data. To further improve data security, the CSTA algorithm is utilized. The performance of the proposed methodology is analyzed in this section and is compared with different existing systems. Tab. 3 shows the values for Running Time, Encryption Time, Decryption Time, Memory Size of Cyclic Shift Transposition Algorithm

Table 3: Evaluation metrics for CSTA algorithm

Parameters file size (KB)	Running time (s)	Encryption time (s)	Decryption time (s)	Memory size (Bits)
2000	1234	252	480	22152362
4000	1456	345	540	23643725
6000	1663	445	620	25783425
8000	1822	500	700	27563902
10000	1932	575	800	28764523

In Fig. 3, the Running time of the proposed methodology is analyzed. For data security, in this paper CSTA algorithm is utilized. Our algorithm compared with two different cryptography algorithms, namely, Advanced Encryption Standard (AES) and Data Encryption Standard (DES), to prove the proposed methodology's effectiveness. In Fig. 3, *X-axis* shows the file size n KB, and the y-axis represents the running time in seconds. The proposed method takes 755 *s*. to run 10000 *kb* of data, which is 850 *s*. When using the AES algorithm and 830 *s*. when using the DES algorithm. Similarly, as file size increases, the running time also increases gradually.

The encryption time for varying data sizes is analyzed in Fig. 4. A sound system should take minimum time for encrypting the data. When analyzing Fig. 5, our proposed method is taken 153425 *s* for encrypting 2000 *kb*, 175638 *s* for 4000 *kb*, 196342 *s* for 6000 *kb*, 202253 *s* for 8000 *kb*, and 243263 *s* for 10000 *kbe*. Compared to the existing method, the proposed method is taken minimum time for the encryption process. This is due to the decentralized architecture that does not depend on a third-party system for encryption.

The decryption time of varying file sizes is analyzed for the proposed method. When analyzing Fig. 5, the proposed method taken minimum time to decrypt the data compared to existing methods.

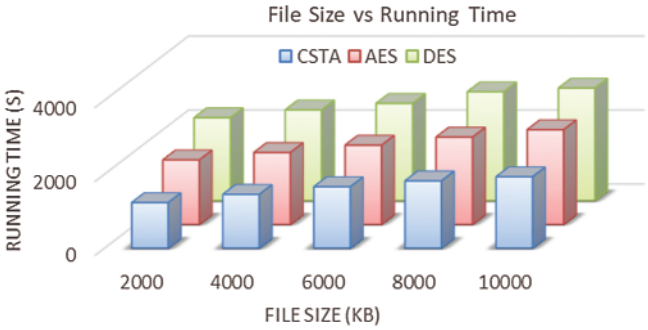

Figure 3: Running time

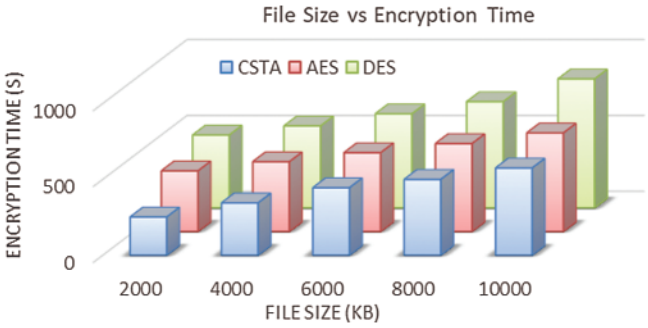

Figure 4: Encryption time

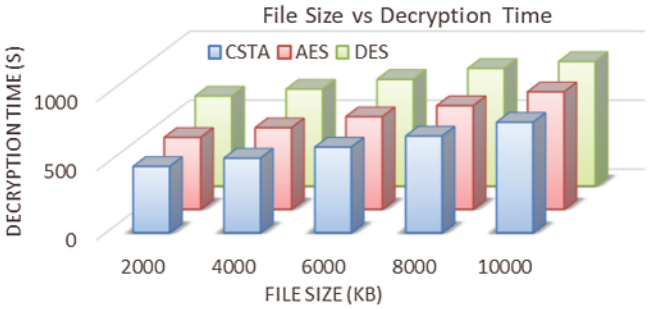

Figure 5: Decryption time

Fig. 6 shows the memory size (in bits) secured by the proposed method. Accordingly, our proposed method has taken 22152362 *bits* of memory for securing 2000 *kb* file, 23643725 bits for 4000 *kb* file, 25783425 *bits* for 6000 *kb* file, 27563902 *bits* for 8000 *kb* file, and 28764523 *bits* for 10000 *kb* file. The memory utilization of existing methods are high compared to the proposed method.

Fig. 7 shows the success rate of service requests granted by the CSP based on user and server trust values. Success rate is directly proportional to the trust degree of a server, that is, when a service request is handled successfully by a server the level of reliability of the server increases. Since RBAC concentrates more on the role and not trust for users, it has the least success rate

in users' presence, successful service requests for genuine users are low for RBAC. TBAC does not check the servers' trust value, and it achieves a lesser success rate than TBAC-CSC. Hence TBAC-CSC has a 6% higher success rate than RBAC and a 2% higher success rate than TBAC.

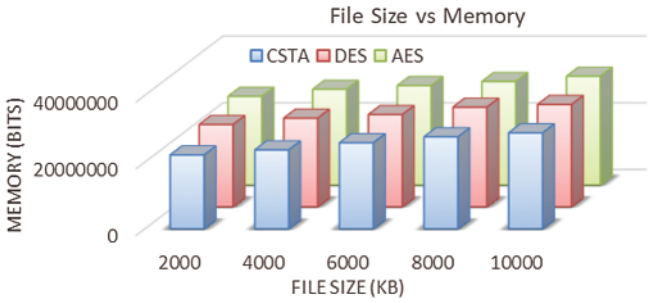

Figure 6: Memory size

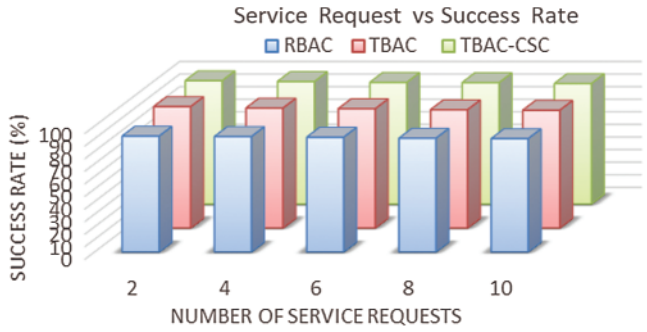

Figure 7: The success rate for service requests

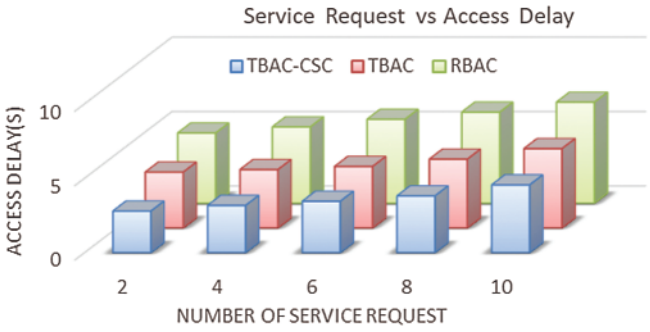

Figure 8: Service access delay for service requests

Fig. 8 shows the delay over various service requests for the varying number of users. It usually reflects the level of server trust. the delay in a service request is inversely proportional to the trust degree of the server, (i.e.,) when the delay is minimum the accessibility of the server increases. Since RBAC does not maintain any users' trust values, the service requests are usually not granted easily to malicious users. Hence the delay in service is higher. Since TBAC does not check the

trust value of servers, it achieves a higher delay than TBAC-CSC. Hence TBAC-CSC has a 37% lesser delay rate than RBAC and 18% lesser delay than TBAC.

Fig. 9 shows the trust value of servers. It shows that server 4 has the highest trust value, followed by servers 1, 3, and 2. The trust values can change based on the server load, service delay and so on which will be reflected in the graph.

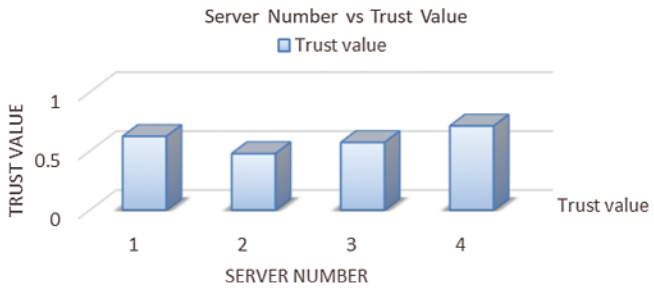

Figure 9: Trust values of servers

6 Conclusion

This paper has presented a trust-based access control framework with secure data storage in a multi-cloud environment. A trust-based access control framework is based on user and server Characteristics. In this framework, trust values are assigned to cloud users using direct trust degrees. The direct trust degree is estimated based on the following metrics: success and failure rate of interactions, service satisfaction index, and dishonesty level. In addition to this, trust values are assigned to cloud servers based on the metrics: server load, service rejection rate, and service access delay. The role-Based Access Control (RBAC) policy of each user is modified based on his trust level. If the server fails to meet the minimum trust level, then another suitable server will be selected. Further, enhance data security, the data has been encrypted using CSTA and stored on the cloud. The basic CSTA algorithm is found to be sufficient for the current application. Enhancements to the algorithm can be done on demand for the future systems it is applied to Experimental results show that the proposed framework achieves reduced access delay with an increased success ratio compared to the RBAC model and TBAC scheme.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. A. Younis, K. Kifayat and M. Merabti, "An access control model for cloud computing," *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 45–60, 2014.

[2] J. Li, G. Zhao, X. Chen, D. Xie, C. Rong *et al.,* "Fine-grained data access control systems with user accountability in cloud computing," in *IEEE Second Int. Conf. on Cloud Computing Technology and Science*, Indianapolis, Indiana, USA, pp. 89–96, 2010.

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

[4] S. K. Prajapati, S. Changder and A. Sarkar, "Trust management model for cloud computing environment," in *Proc. of the Int. Conf. on Computing. Communication and Advanced Network*, Chennai, India, pp. 1–5, 2013.

[5] M. Sinha, S. Silakari and R. Pandey, "Trust-based mechanism for secure cloud computing environment: A survey," *International Journal of Engineering Science Invention*, vol. 5, no. 3, pp. 17–23, 2016.

[6] C. Wu and S. Marotta, "Framework for assessing cloud trustworthiness," in *IEEE 6th Int. Conf. on Cloud Computing*, Washington, DC, United States, pp. 956–957, 2013.

[7] G. Lin, Y. Bie and M. Lei, "Trust-based access control policy in multi-domain of cloud computing," *JCP*, vol. 8, no. 5, pp. 1357–1365, 2013.

[8] V. K. Prasad, M. Shah, N. Patel and M. Bhavsar, "Inspection of trust-based cloud using security and capacity management at an IaaS level," *Procedia Computer Science*, vol. 132, no. 1, pp. 1280–1289, 2018.

[9] P. S. Challagidad, V. S. Reshmi and M. N. Birje, "Reputation-based trust model in cloud computing," *Internet Things Cloud Computing*, vol. 5, no. 1, pp. 5–12, 2017.

[10] Z. Tang, A. Liu, Z. Li, Y. J. Choi, H. Sekiya *et al.,* "A trust-based model for security cooperating in vehicular cloud computing," *Mobile Information Systems*, vol. 2016, pp. 1–22, 2016.

[11] L. A. A. Tawalbeh, F. Ababneh, Y. Jararweh and F. AlDosari, "Trust delegation-based secure mobile cloud computing framework," *International Journal of Information and Computer Security*, vol. 9, no. 2, pp. 36–48, 2017.

[12] J. Yu, P. Lu, Y. Zhu, G. Xue and M. Li, "Toward secure multikey word top-k retrieval over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 239–250, 2013.

[13] H. He, R. Li, X. Dong and Z. Zhang, "Secure, efficient and fine-grained data access control mechanism for P2P storage cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 471–484, 2014.

[14] S. S. Hussain and V. Yuvaraj, "A secure data access control method using AES for P2P storage cloud," in *Int. Conf. on Innovations in Information, Embedded and Communication Systems*, Coimbatore, India, pp. 1–5, 2015.

[15] J. Wei, W. Liu and X. Hu, "Secure data sharing in cloud computing using revocable-storage identity-based encryption," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 1136–1148, 2016.

[16] K. Lee, "Comments on secure data sharing in cloud computing using revocable-storage identity-based encryption," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1299–1300, 2020.

[17] B. Lang, J. Wang and Y. Liu, "Achieving flexible and self-contained data protection in cloud computing," *IEEE Access*, vol. 5, pp. 1510–1523, 2017.

[18] C. Gu, F. Luo, Y. Li and W. Ding, "Dynamic access control model based on FAHP in cloud environment," in *IEEE 4th Int. Conf. on Computer and Communications*, Chengdu, China, pp. 1938–1943, 2018.

[19] Q. Li, Y. Tian, Y. Zhang, L. Shen and J. Guo, "Efficient privacy-preserving access control of mobile multimedia data in cloud computing," *IEEE Access*, vol. 7, pp. 131534–131542, 2019.

[20] S. Zhou, G. Chen, G. Huang, J. Shi and T. Kong, "Research on multi-authority CP-ABE access control model in multi-cloud," *China Communications*, vol. 17, no. 8, pp. 220–233, 2020.

[21] Q. Zhang, S. Wang, D. Zhang, J. Wang and Y. Zhang, "Time and attribute-based dual access control and data integrity verifiable scheme in cloud computing applications," *IEEE Access*, vol. 7, pp. 137594–137607, 2019.