

Service-Centric Dynamic Trust AnalysisBased Access Restriction and Blockchain Public Auditing for Efficient Collaborative System in Cloud Environment

VelMurugeshKumar. N¹, R. Nedunchelian²

¹Research Scholar, Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamilnadu, India

²Professor, Department of Electronics and Communication Engineering, Karpaga Vinayaga College of engineering and technology, Madhuranthagam, Chengalpattu District, Tamilnadu, India

Abstract:

The cloud environment becomes more supportive for collaborative systems and has been used by several organizations in maintaining and access the data. The cloud environment introduces different services in different levels, which have been accessed by different users of the collaborative environment. However, restricting malformed access and public auditing becomes a key issue in a collaborative environment. To provide such operational and data transparency and consistency, different approaches are discussed in the literature. The methods perform public auditing at feature level and another scenario. Still, access restriction and public auditing-related parameters affect the entire performance in the cloud. To overcome these issues, a service-centric dynamic trust analysis access restriction and blockchain public auditing (DTAR-BCPA) model is presented in this article. The method initially receives the service request and performs service-centric feature trace trust analysis (SCFTTA) to perform the accessrestriction. The method estimates Trusted Access Score (TSA) and Feature Trust Score for specific service requested according to feature grants of the user according to the taxonomy and the traces of access. According to these two, the method estimates the trusted access score towards access restriction. Second, the method performs public auditing with blockchain technology. The method block-level trust score is measured based on the approval of different users. The method improves the performance of public auditing and access restriction towards performance improvement of a collaborative environment.

Index Terms:

Cloud Systems, Collaborative Environment, Access Restriction, Trust Analysis, SCFTTA, Public Auditing, TSA.

1. Introduction

Modern organizations maintain the number of data of huge volume in different locations of their units. As the volume of data becomes higher at each fraction, which challenges the organization in maintaining the data in their dedicated data servers due to the cost and space complexity. To handle this, the organizations move towards a cloud environment which provides several services in different layers and level of the environment. By using the services provided, the user can access and modify the data to perform their routine job. This makes it feasible for the users of a collaborative environment to work together.

The collaborative environments are meant for the users or employees of any organization to work together on a shared resource. For example, consider a design process of an automobile manufacturing unit, where the employees of the organization located in various geographic units work on the same design. Whatever changes made by a unit or employee should be reflected on the resource as well reflect on the copy of others. The collaborative environment performs this task by maintaining a shared resource.

In any collaborative environment, the organization has the responsibility in maintaining the privacy of the user data as well as it has the responsibility in providing exact and correct data to all the users. The user always looks for the correct and meaningful data for their job. As the data has been accessed by different users in the collaborative environment, the cloud should provide the exact data or correct data to all the users. Towards this, public auditing is performed. Public auditing is the process of providing correct data to the user which has been enforced at the update process. Public auditing is performed in different ways either by feature level which enforces public auditing in feature level as whatever the update made by the user has been updated to the copy of others. Similarly, public auditing is performed in block-level which performs public auditing at each block.

On the other side, the performance of a collaborative environment is depending on how the security has been enforced. If the data present in a collaborative environment is accessed by malformed users, then they would perform different threats to damage the data which in turn would affect the performance of the entire system. This is where access restriction comes to play, which would restrict the malformed access. The access restriction has been enforced in different ways. For example, it has been performed according to the feature and profile-based approaches. Also, it has been performed on several levels towards access restriction. Due to the demerits, Service-Centric Dynamic Trust Analysis Based Access Restriction and Blockchain Public Auditing approach is presented in this paper. The blockchain is the technology recently used towards restricting access as well as public auditing. In this approach, the method maintains the data in form of blockchain and metadata available in the cloud system. By enforcing and maintaining the data in the blockchain, the public auditing and access restriction can be enforced greatly.

2. Related Works

Various methodologies were discussed towards access restriction and public auditing in a collaborative environment.

Simulation tools for the cloud environment were described in [1]. The author describes that the GDC tool was suitable for simulating applications with data centers. The cloud sim has been identified as a tool to manipulate network operations where the hardware-related problems are supported with MDC. In [2], the problem of scheduling various services and applications in the cloud has been considered. The author performed a detailed study on how energy and load balancing can be performed. However, the author identified different simulators that support performance analysis. Similarly, a detailed review of various technologies that support cloud services based on user needs was discussed in [3].

In [4] and [5], authors presented the importance of implementation of cloud service for various applications such as agriculture, medicine, yield prediction data storage, and so on. A detailed study on the performance analysis of security-related quality of service (QoS) was proposed in [6]. The analysis is performed due to the infeasibility of analyzing the performance using hardware as they cost higher. The analysis is performed through different simulation tools. In [7], the author presents a prediction model that measures the performance of different cloud resources. The model has been evaluated the scaling utility of computing resources. The

behaviors of the resources and systems have been captured and used to measure the performance of unknown metrics.

In [8], the author presents a comparative study on the performance of simulation tools like Opnnebula and Openstack. The analysis is performed according to the architecture, security, and provenance. Finally, the author provides different recommendations for the deployments based on the demands of users. In [10], the author presented an Open cloud Testbed for reliable network services with an exclusively adapted 4 data centers where each has 30 nodes. The data centers are located in Chicago, San Diego, and Baltimore. The testbed is widespread geographically which is connected with a 10Gb/s network that supports high-performance requirements.

In [11], a testbed named Open Cirrus is presented which provides data centers being distributed to support research and open-source service stacks in the cloud. Similarly, a testbed on the same provides both physical and virtual machines. The services are distributed globally to perform monitoring and job submission. In [14], the author presented a flexible infrastructure supportive simulator for a cloud environment named iCanCloud. The tool provides higher usability, flexibility, and more scalable. It also provides a user interface to perform various operations.

Software-hardware-based Cloud simulators and classifiers were proposed in [15], their features of each simulator class have been analyzed and discussed. In [16], the author describes the features of the Cloud Sim simulator and compares its variants like green cloud, cloud analyst, EMUSIM, and MDCSIM. The author presents a comparative study on support towards platform services, network services, and language services.

In [17], the author presented a survey on cloud simulators according to mathematical models and different testbeds are used to measure their performance in various factors. The survey is performed to identify the suitable modeling type for the evaluation of different cloud strategies. In [18], the author presents an open-source framework to support cloud functionality named Eucalyptus to support control on the virtual machines. It is capable of controlling the virtual machines globally and the author outlined the architecture to support portability and simpler which can be deployed even in an academic environment.

In [19], the author presented a resource allocation strategy that has been simulated over CloudSim. As the simulator provides API to control the systems and behavior of the resources of cloud-like virtual machines. It supports the extension of any resource provisioning algorithm by modifying the existing ones. In [20], the author simulated the infrastructure services and how they can be controlled through the Cloudsim simulator. A framework developed by the authors allowed modeling the self-contained platforms to support scheduling and resource allocation policies.

In [21], the problem of environmental monitoring has been performed through cloud computing. The method developed an infrastructure to maintain the earth observation data. They designed a testbed that allows on-demand data discovery services which can be accessed by different scientists and researchers. The framework allows the integration of multiple source earth observation data obtained from different geographic locations and enables coordinative resource computing strategies. In [22], a platform named HiTempo has been designed which supports the

analysis of time-series data. The platform supports the analysis of satellite images and analyses various algorithms. A model-based algorithm like the Kalman filter has achieved a higher detection rate. Similarly in [23], the multi-source remote sensing data have been integrated to identify the changing conditions of land cover. The land cover maps have been used to perform the changing conditions.

In [24], the author a software architecture to support network-based applications. The author presents a detailed survey of various architectures that enable network services. Finally, a REST (Representational State Transfer) is presented and describes the guidelines to use through the modern web. Resource-oriented cloud applications were created by the author in [25] for solving the problems of previous models.

In [26], the author presented an updated version of Multi-Verse Optimizer to achieve higher task planning in the cloud. The author compared their proposed work with the existing algorithms and achieved minimized makespan of time. In [27], the author proposed an advanced cloud computing environment for library management systems based on the cloud. They have tested the performance of their system with various librarians, universities and proved that their design was secured and reliable. In [28], the author presented a new complex system simulation method over the existing methods. This method provides a cloud environment service to perform simulations over the data.

Novel methodology related to the cloud was proposed in this study to increase the performance of cloud storage and security.

3. Service-Centric Dynamic Trust Analysis Access Restriction and Blockchain Public Auditing (DTAR-BCPA) model

The proposed method receives the user request and identifies the service being requested. According to the service requested, the method applies service-centric feature trace trust analysis (SCFTTA) to perform the access restriction. The SCFTTA approach estimates trusted access score (TAS) for specific services requested according to feature grants of the user according to the taxonomy and the traces of access. According to these two, the method estimates the trusted access score towards access restriction. Second, the method performs public auditing with blockchain technology. The method block-level trust score is measured based on the approval of different users. The detailed approach is presented in this section.

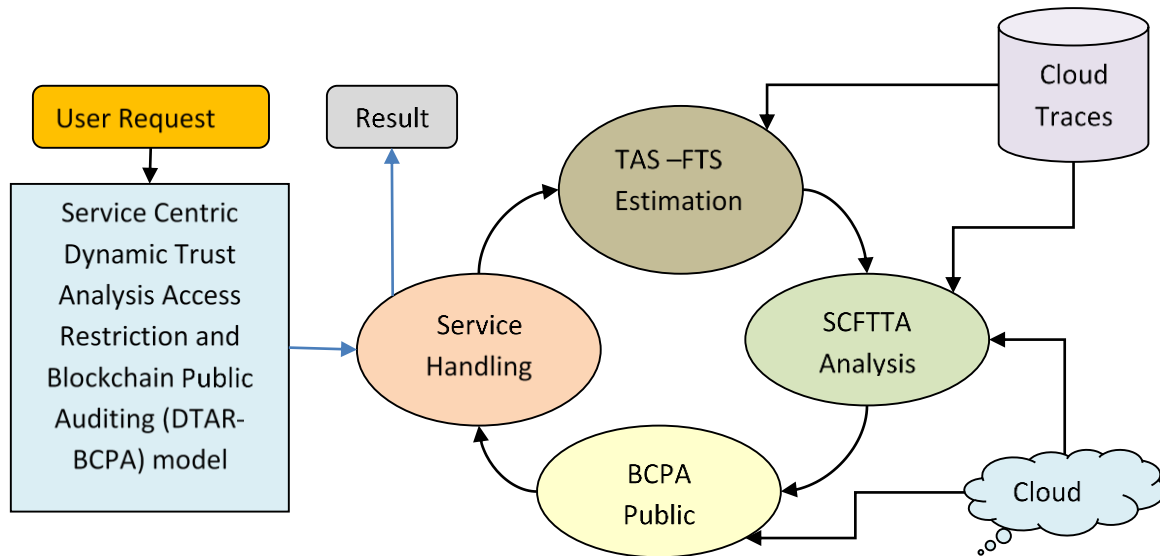


Figure 1: Architecture of Proposed DTAR-BCPA Model

The architecture of the proposed DTAR-BCPA model has been presented in Figure 1 and shows various functional components incorporated. Also, the working components are detailed in this section sharply.

3.1 Service Handling

The proposed method receives the user request and identifies the service being requested. Initially, using the service taxonomy, the user request has been validated for the possession of access for the user. If the user has access to the service then the method performs SCFTTA (Service centric feature trust and trace analysis). According to the result of SCFTTA, the method allows or denies the request. If the user does not have access to the service, then the request is denied. Similarly, when the user clears the trust check, the method performs Blockchain public auditing to enforce legitimacy. The BCPA approach enforces data encryption and validates the correctness of the data. If the user clears the BCPA test, then the data has been updated successfully. The result of service handling has been given to the user.

Algorithm:

Given: Service Taxonomy ST, Service Request Se_Req.

Obtain: Null

Start

Read Service Taxonomy ST and Service Request Se_Req

Identify the user $Uid = UserId \in Sr_Req$

-- (1)

Identify the service requested $Sid = ServiceId \in Sr_Req$

-- (2)

Compute Trust Weight Tw = perform SCFTTA Analysis.

Compute Block Level Trust Score BLTS = Perform BCPA.

If $TW > Th$ and $BLTS > BTh$ then

Allow access and perform access.

Else

Deny access

End

Terminate

The proposed service-centric dynamic trust analysis request-response model has been presented in the above pseudo-code. According to that the method identifies the user id and estimates TSA and BLTS. Based on these values, service handling is performed.

3.2 Trusted Access Score Estimation:

The trusted access score represents user access authentication. The service allows multiple access for multi-user service. Sometimes, the trust of service access is justified according to the behavior of how he accesses the service. If the user accesses the service and breaks the service access on the fly then it would damage the service data as well as its performance. So, by counting the number of service access and number finished the service access, the value of TAS (Trusted Access Score) can be measured. Also, it has been measured by counting the number of levels of service he has completed and the total number of levels it has.

Consider the service claimed is S , and the user trace is maintained as UT , then the value of TAS is measured as follows:

Compute the number of service access $NA = \sum_{i=1}^{size(UT)} UT(i).Service = S$ -- (3)

The above equation (3) finds the number of times the service has been accessed by the user. Now, the number of times the service access finished as using equation (4).

Compute the Number of times completed NTC.

$NTC = \sum_{i=1}^{size(UT)} UT(i).Service = S \ \&\& \ UT(i).state = Complete$ -- (4)

The service would have several levels and it has been identified as follows:

Compute several levels the service has as Nl .

$Nl = \sum_{i=1}^{size(ST)} ST(i).service == S \ \&\& \ ST(i).levels$ -- (5)

Compute Average no of levels completed $Anlc$.

$Anlc = \frac{\sum_{i=1}^{size(UT)} UT(i).service == S \ \&\& \ UT(i).level}{size(UT)}$ -- (6)

Using all these values, the value of the Trusted access score has been measured as follows:

Compute $TAS = \frac{NTC}{NA} \times \frac{Anlc}{Nl}$ -- (7)

Calculated TAS was implemented in SCFTTA analysis for user authorization.

3.3 Feature Trust Score Estimation

The feature trust score is the measure that represents the trust of service access according to the service taxonomy. The service taxonomy contains the protocols of the service as well as the set of features to which it has access to complete the service. Similarly, the taxonomy has user profiles that contain details of features to which the user has access. Using both of them, the method computes the value of the feature trust score (FTS). It has been measured as follows:

First, compute the number of features the service needs to access as follows:

$NFNA = \sum_{i=1}^{size(Taxonomy)} Taxonomy(i) \rightarrow Sreq.$ -- (8)

Second compute the number of features to which the user has access as follows:

$NFA = \sum_{i=1}^{size(Taxonomy)} Taxonomy(i) \rightarrow Sreq \ \&\& \ Taxonomy(i) \in User$ -- (9)

Compute the value of $FTS = NFA / NFNA$ -- (10)

The estimated value of FTS was used to perform authorization.

3.4 Service-Centric Feature Trace Trust Analysis (SCFTTA)

The SCFTTA approach read the traces of user access. Using the user access trace the method computes the value of trusted access score (TSA) which is calculated based on the number of service access and service completion. Also, the method computes the feature trusted score (FTS) according to the service taxonomy which specifies the number of features the service needs to be accessed and several features to which the user has access. Using both of them, the method computes the value of trust weight, based on which the access restriction is performed.

Algorithm:

Given: User Trace UT, Service Taxonomy ST, User Request Ureq

Obtain: Trust Weight Tw

Start

Read UT, ST, Ureq.

TAS = Estimate Trusted Access Score.

FTS = Estimate Feature Trust Score.

Compute Trust Weight Tw = TAS×FTS. – (11)

Stop

The above-discussed algorithm estimates trusted access to score and feature trust score to measure the trust weight towards access restriction in the cloud environment.

3.5 Blockchain Public Auditing (BCPA)

The proposed approach enforces public auditing with blockchain technology. The method maintains the data in form of blocks and each block has been tied with a reference number which denotes the link to the next block. However, the user has access to different data in the chain, they will be allowed to access only specific blocks and features of the chain. To enforce this, when a user request for update request, the method computes the block level trust score (BLTS) towards various blocks the user request claims. It has been measured based on the genuine of the data submitted. To estimate this, the method decrypts each data in the block where the data update to be done and verify each of them. According to that, the method computes the number of blocks the request accesses and the number of blocks verified correctly. Also, the method collects approval from limited owners and count on them. Using both of them, the method computes the value of BLTS and based on that the method performs public auditing and allows the update service.

Algorithm:

Given: Update Request Ureq, Blockchain BC

Obtain: BLTS

Start

Read Ureq and BC.

Identify the number of blocks to be updated
$$\text{Nobu} = \sum_{i=1}^{\text{size}(BC)} Ureq \in BC(i) \quad \text{-- (12)}$$

For each block b

Block Data Bd = *Decrypt*(Bc(b))

Boolean bool = *Verify* Bd.Signature

If true then

 Increase count

End

End

$$\text{Compute BLTS} = \frac{\text{count}}{\text{Nobu}} \times \frac{\sum_{i=1}^{\text{Nobu}} \text{BC}(i).\text{access}==\text{true}}{\text{Nobu}} \quad \text{-- (13)}$$

Stop

The above-discussed algorithm represents how the blockchain public auditing is performed by counting the signature as well as the number of blocks to be updated and the number of blocks to which the user has access. Also, the method verifies the signature of the blocks to enforce public auditing and according to that, the method computes the value of BLTS. The value of BLTS has been used in enforcing the service handling.

4. Results and Discussions

The present approach has been implemented in the Microsoft Azure platform and its metrics were calculated on different parameters. Obtained results have been compared with the results of other approaches. The details of the evaluation are presented in Table 1 below.

Table 1: Evaluation Details

Parameter	Value
Tool	Microsoft Azure
Programming	Advanced Java
Number of Services	200

The details of evaluation towards performance analysis of proposed approach have been presented in Table 1. The methods are evaluated on different metrics and obtained results are presented and discussed in this section.

Table 2: Analysis of content sharing

Performance on Content Sharing			
	50 Services	100 Services	200 Services
BT	76	81	84
OCT	82	85	87
REST	85	87	89
STS	87	89	91
DTAR-BCPA	89	93	97

The performance on content sharing has been measured and presented in Table 2. The proposed DTAR-BCPA method improves the performance in content sharing up to 97% which is higher than any other approach.

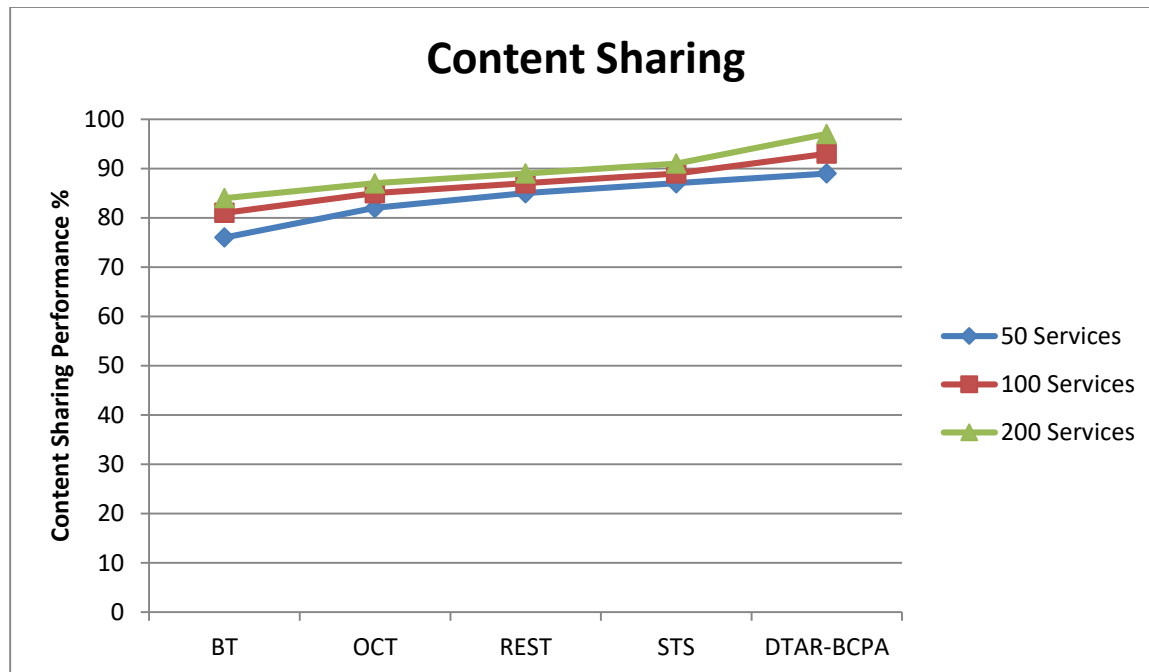


Figure 2: Analysis of content Sharing performance

The performance of DTAR-BCPA towards content sharing has been measured on a varying number of services. The proposed DTAR-BCPA approach resulted in improved performance in data sharing.

Table 3: Analysis of Access restriction

Performance on Access Restriction			
	50 Services	100 Services	200 Services
BT	72	76	79
OCT	75	78	83
REST	78	82	86
STS	81	85	89
DTAR-BCPA	84	87	96

The performance on access restriction has been measured and presented in Table 3. The proposed DTAR-BCPA method improves the performance in access restriction up to 96% which is higher than any other approach.

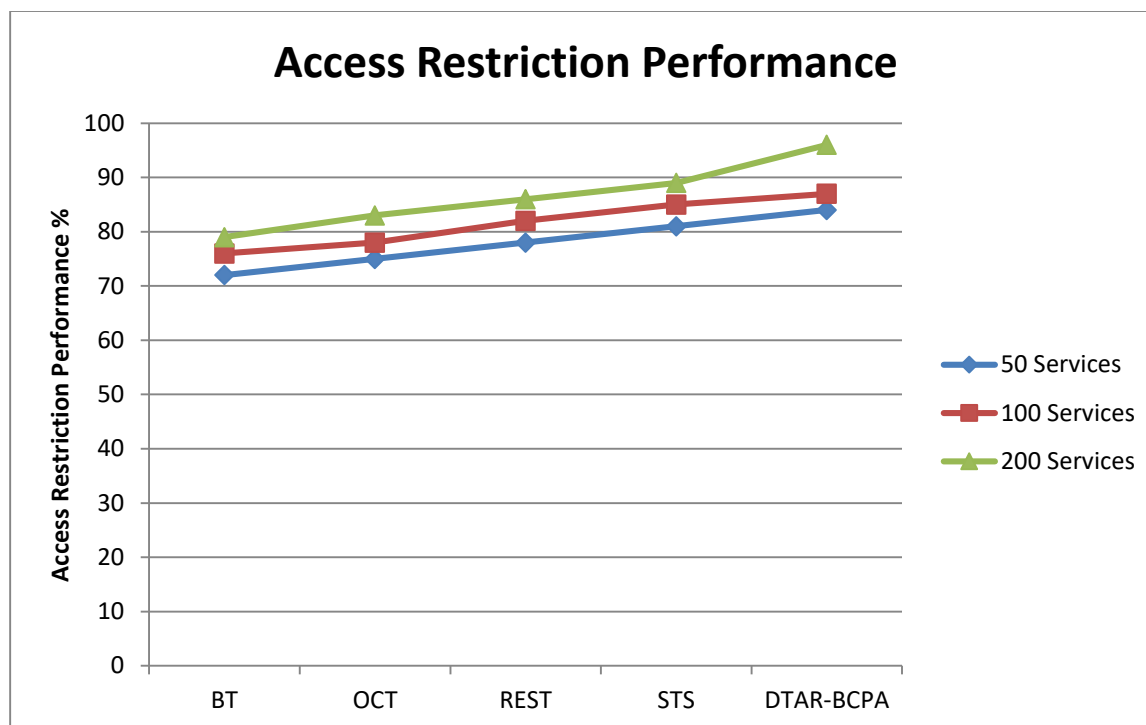


Figure 3: Analysis of access restriction performance

The performance of DTAR-BCPA towards access restriction has been measured on a varying number of services. The proposed DTAR-BCPA approach resulted in improved access restriction compared to other approaches.

Table 4: Analysis of Public Auditing Performance

Performance on Public Auditing			
	50 Services	100 Services	200 Services
BT	68	71	74
OCT	71	74	78
REST	75	78	81
STS	81	85	89
DTAR-BCPA	84	87	96

The performance on public auditing has been measured and presented in Table 4. The proposed DTAR-BCPA method improves the performance in public auditing up to 96% which is higher than any other approach.

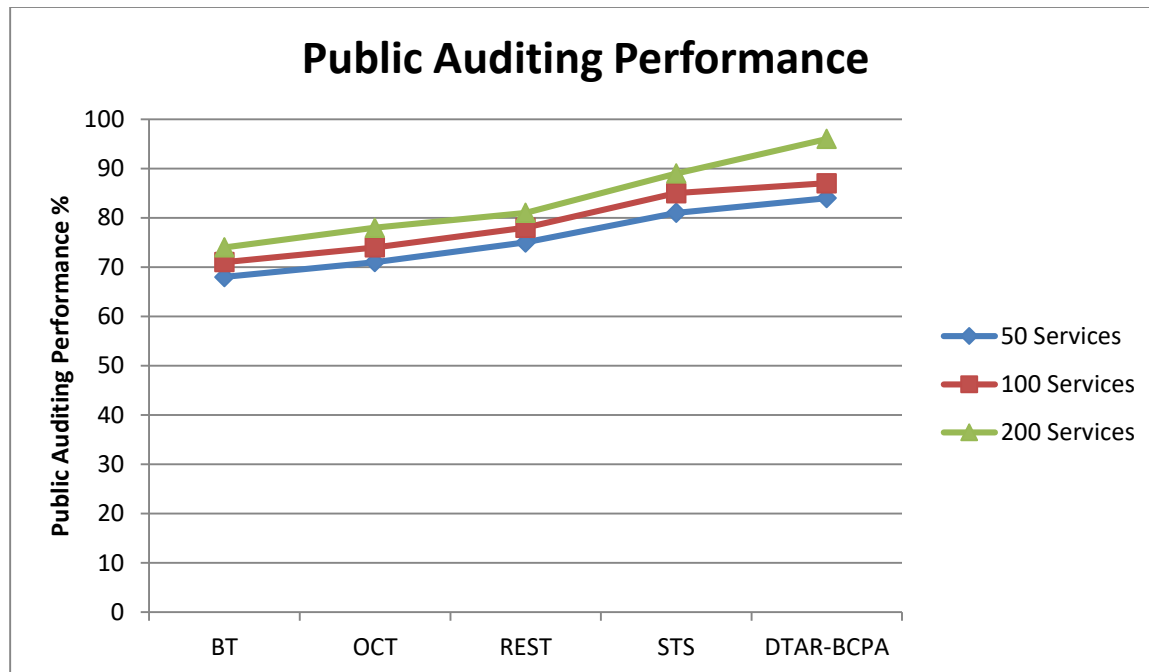


Figure 4: Analysis of public auditing performance

The performance of DTAR-BCPA towards public auditing has been measured on a varying number of services. The proposed DTAR-BCPA approach resulted in improved performance in public auditing compared with other models.

Table 5: Analysis of Throughput Performance

Performance on Throughput			
	50 Services	100 Services	200 Services
BT	72	76	79
OCT	75	78	83
REST	78	82	86
STS	81	85	89
DTAR-BCPA	84	87	96

The performance on throughput has been measured and presented in Table 3. The proposed DTAR-BCPA method improves the performance in throughput up to 96% which is higher than any other approach.

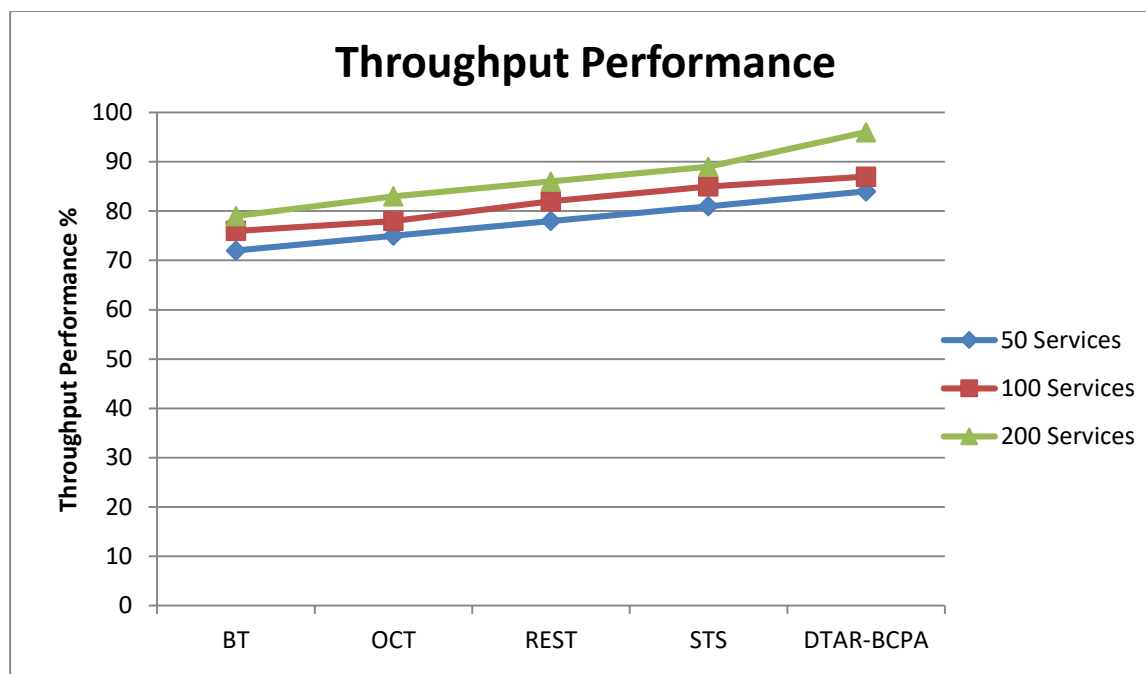


Figure 5: Analysis of throughput performance

The performance of DTAR-BCPA towards throughput achievement has been measured on a varying number of services. The proposed DTAR-BCPA approach has shown progress in throughput than other approaches mentioned.

5. Conclusion

This article presented a novel service-centric dynamic trust analysis access restriction and public auditing scheme for the cloud environment. The method receives the user request and analyzes the trust of the user by applying a service-centric feature trust trace analysis scheme which measures features trust score and trusted access score to measure trust weight. Further, the method applies, blockchain public auditing to enforce public auditing. The proposed method estimates blockchain trust scores to enforce blockchain public auditing. The proposed method improves the performance of access restriction and public auditing up to 97%.

References

1. Rizwana Shaikh, M. Sasikumar, "Cloud Simulation Tools: A Comprehensive Analysis, International Conference on Green Computing and Technology", 2013.
2. Khairunnisa, M. Nazreen Banu, Cloud Computing Simulation Tools: A study, International Journal of Fuzzy mathematic Archive, January 2015.
3. Rydhm Beri, Veerawali Behal, Descriptive Study of Cloud Computing: An Emerging Technology, IJRITCC, March 2015.
4. Godfrey Winsters, S., Siva Kumar, A., Gopirajan, P.V., Loganathan, V., "Behaviour analysis of social network application using natural language processing - a machine learning approach", European Journal of Molecular and Clinical Medicine, 7(4), pp. 2297-2306. Dec. 2020.
5. Gopirajan, P.V., Gopinath, K.P., Sivaranjani, G., Arun, J., "Optimization of hydrothermal liquefaction process through machine learning approach: process conditions and oil yield", Biomass Conversion and Biorefinery, <https://doi.org/10.1007/s13399-020-01233-8>, Jan. 2021

- 6.KaurGaganjot, Kumar Pawan,"Study of Comparison of Various Cloud Computing Simulators", 2nd National Conference in Intelligent Computing & Communication.
- 7.Bashar, "Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models," in IEEE International Conference on Cloud Networking (CLOUDNET 2013), Nov. 2013, pp. 200–204.
- 8.D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan, "GreenCloud: A packet-level Simulator of Energy-Aware Cloud Computing Data Centers," in IEEE GLOBECOM 2010, Dec. 2010, pp. 1–5.
- 9.X. Wen, G. Gu, Q. Li, Y. Gao, and X. Zhang, "Comparison of open-source cloud management platforms: Openstack and Opennebula," in IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2012), May 2012, pp. 2457–2461.
- 10.R. L. Grossman, Y. Gu, M. Sabala, C. Bennet, J. Seidman, and J. Mambretti, "The Open Cloud Testbed: a wide-area testbed for cloud computing utilizing high-performance network services," in Gridnets 2009, Sep. 2009, pp. 89–97.
- 11.I. A. et al., "Open Cirrus: a global cloud computing testbed," IEEE Computer, vol. 43, no. 4, pp. 35–43, 2010.
- 12.R. C. et al., "Open Cirrus Cloud Computing testbed: federated data centers for open source systems and services research," in USENIX Hotcloud '09, June 2009, pp. 1–5.
- 13.B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications," in 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010), Apr. 2010, pp. 446–452.
- 14.Nez, J. L. Vazquez-Poletti, A. C. Caminero, G. G. Casta, J. Carretero, and I. M. Llorente, "iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator," Journal of Grid Computing, vol. 10, no. 1, pp. 185–209, Mar. 2012.
- 15.W. Zhao, Y. Peng, F. Xie, and Z. Dai, "Modeling and simulation of cloud computing: A review," in IEEE Asia Pacific Cloud Computing Congress (APCloudCC 2012), Nov. 2012, pp. 20–24.
- 16.R. Malhotra and P. Jain, "Study and Comparison of CloudSim Simulators in the Cloud Computing," SIJ Transactions on Computer Science Engineering and its Applications (CSEA), vol. 1, no. 4, pp. 111–115, Sep. 2013.
- 17.G. Sakellari and G. Loukas, "A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing," Simulation Modelling Practice and Theory, vol. 39, no. 0, pp. 92–103, Dec. 2013.
- 18.D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus open-source cloud-computing system," in 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009, May 2009, pp. 124–131.
- 19.R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, Jan 2011.
- 20.R. N. Calheiros, R. Ranjan, C. A. F. D. Rose, and R. Buyya, "Cloudsim: a novel framework for modeling and simulation of cloud computing infrastructure and services," University of Melbourne, Tech. Rep., 2009.

21. Wenyang Yu ; JiboXie ; Guoqing Li, Cloud computing for environmental monitoring using multi-source Earth Observation data, Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics), 2016.
22. F. Van Den Bergh et al., "HiTempo: a platform for time-series analysis of remote-sensing satellite data in a high-performance computing environment", International Journal of Remote Sensing, vol. 33, no. 15, pp. 4720-4740, 2012.
23. C. Petit, E. Lambin, "Integration of multi-source remote sensing data for land cover change detection", International Journal of Geographical Information Science, vol. 15, no. 8, pp. 785-803, 2001.
24. R.T. Fielding, "Architectural styles and the design of network-based software architectures", University of California, 2000.
25. C. Granell et al., "Enhancing integrated environmental modelling by designing resource-oriented interfaces", Environmental Modelling & Software, vol. 39, pp. 229-246, 2013.
26. Sarah E. Shukri, Rizik Al-Sayyed, AmjadHudaib, SeyedaliMirjalili, "Enhanced multi-verse optimizer for task scheduling in cloud computing environments", Expert Systems with Applications, 2020, 114230, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2020.114230>.
27. AdeyinkaTella, ScholasticaChizomaUkwoma, Adeniyi Isaiah Kayode, "A two models modification for determining cloud computing adoption for web-based services in academic libraries in Nigeria", The Journal of Academic Librarianship, Volume 46, Issue 6, 2020, 102255, ISSN 0099-1333, <https://doi.org/10.1016/j.acalib.2020.102255>.
28. Shuai Wang, Feng Zhu, Yiping Yao, Wenjie Tang, Yuhao Xiao, SiqiXiong, "A Computing Resources Prediction Approach Based on Ensemble Learning for Complex System Simulation in Cloud Environment", Simulation Modelling Practice and Theory, 2020, 102202, ISSN 1569-190X, <https://doi.org/10.1016/j.simpat.2020.102202>.