PaaS Platform Securityenhancement Using Fuzzy and Trust Based Signature

PATHAKAMURI SRINIVASULU (✉ srinivas.nlr2004@gmail.com)

JNTUA: Jawaharlal Nehru Technological University Anantapur

B.V. Ramana Reddy

KSRM College of Engineering, Kadapa, A.P., India.

A.P. Siva Kumar

JNTUA CEA: Jawaharlal Nehru Technological University Anantapur College of Engineering
Ananthapuramu

Research Article

Keywords: PaaS platform, security enhancement, fuzzy, trust based signature, CDO

Posted Date: July 30th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-742860/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

PaaS PLATFORM SECURITY ENHANCEMENT USING FUZZY AND TRUST BASED SIGNATURE

Pathakamuri Srinivasulu^{1*}, B.V. Ramana Reddy², A.P. Siva Kumar³

¹ Research Scholar, Department of CSE, JNTUA, Anantapur, Andhra Pradesh, India

² Professor, Department of CSE, KSRM College of Engineering, Kadapa, A.P., India

³ Associate Professor, Department of CSE, JNTUA, Anantapur, Andhra Pradesh, India

*Corresponding Author: srinivas.nlr2004@gmail.com

ABSTRACT

The study of PaaS platform security enhancement has occupied scholars from a number of disciplines, in previous works has so many security issues like Security problems can be a big barrier to cloud computing. System servers require trustworthy security measures to different data domains according to the system servers own operating mechanism. Problem is constructed by filtering out those cloud providers not conforming to high-level security requirements. By including low-level security requirements to be used for filtering the cloud provider space and formulating the optimisation function. To overcome all the above drawbacks our proposed work mainly focused on the security of Platform-as-a-Service (PaaS) as well as the most critical security issues that were documented regarding PaaS infrastructure. This work has two main aspects: First, suitable access control on user personal data, VMs and platform services and Second planning and adapting application deployments based on security requirements. In Fuzzy based access control to information sources is mainly realised by exploiting the CDO security feature. In Security feature code was modified to map the class and packet filter for any specific permission to our own class. If the Identity Provider (IdP) has included public security information on the two main parts in the small token on which Trust based Signature elements are placed, i.e., the whole token or the assertions included, this public key is used to validate the respective signature. The experimental results will show that our proposed method outperforms the traditional methods. Our proposed methodology was implemented in the platform of JAVA.

Keywords: PaaS platform, security enhancement, fuzzy, trust based signature, CDO

1. INTRODUCTION

Cloud computing can be seen as network-enabled services that provide scalable, QoS guaranteed services on demand that can be accessed over the Internet [1, 2]. Cloud computing is a developing area, where three types of services are provided to customers. They are Infrastructure as a Service (IaaS): is most prevalent and developed market segments of cloud that deliver customized infrastructure on demand, Platform as a Service (PaaS): that provides platform and environment to the developers that build cloud services and application on the web and that services are stored in the cloud and accessed by cloud users using web browser, Software as a Service (SaaS): that provides its own application running on a cloud infrastructure[3, 4].

PaaS cloud service is focused here. PaaS clouds that host and run applications from several different users in the same platform in a safe manner. Platform as a service (PaaS) allows cloud developers and providers of higher-level services [i.e., software as a service (SaaS)] to build and deploy applications by hiding the complexity of lower level functions and services [5,6].

Security is a biggest challenge in cloud computing for various organizations that rely completely on CSP to store its data. Moreover, the PaaS platform enables resource sharing, which inevitably brings security issues. Furthermore, multi-tenancy is the key source of security concerns when multiple PaaS services reside on a single physical server. To achieve safe (or secure) multi-tenancy, PaaS solutions must have a way to isolate tenants from each other. Multi-tenancy refers to multiple cloud users running independent logical processes but sharing the same physical components, such as CPU, RAM and storage. Where, Multi Cloud Storage means the utilization of various cloud storage services using a single web interface rather than the defaults provided by the cloud storage vendors in a single heterogeneous architecture [7]. Multi-tenancy in cloud computing systems is enabled through virtualisation [8]. Virtualisation is the enabling technology for virtual machines, which emulates a physical server and is managed through software known as hypervisors[9].Based on the analysis of the multi-tenant access to each service on the PaaS platform, the system establishes the identity authentication and access control process in the PaaS platform.

Access control technology is an important tool. It allows legitimate users to gain access to information and system resources within legitimate time periods and prevent unauthorized users from accessing information and system resources by denying the access [10, 11].Access control can be considered from two aspects: authentication and authority [12, 13].Authentications services checks and confirm user identity through a secret code like password and verify whether the information of the user given is real or not. The key purpose of authentication is to enhance the security and get rid of the illegal access of data and also minimize the attacks like password stolen done by the attackers. Also, Authority refers to a kind of judgment and control of whether the subject of access allows access to a specific resource[14, 15].

2. RELATED WORKS

KyriakosKritikos et al [16] presented a novel model-driven approach and architecture which secures multi-cloud platforms, enables users to have their own private space and guarantees that application deployments are not only constructed based on but can also maintain a certain user-required security level. Such solutions exploits state-of-the art security standards, security software and secure model management technology. Moreover, it covers different access control scenarios involving external, web-based and programmatic user authentication.

Jinan Shen et al [17] presented a domain-divided security model in which different security policies are separately applied for three domains: the data storage domain, the data processing domain and the data transmission domain. In addition, security policies can be configured for upper-level applications based on their security requirements. Experimental results show that our security model is both practical and lightweight as it can provide differentiated security protection for cloud computing-based telecommunication service with a low overhead.

RobailYasrabet *al* [18] displayed a security of Platform-as-a-Service (PaaS) as well as the most critical security issues that were documented regarding PaaS infrastructure. The prime outcome of this study was a security model to mitigate security vulnerabilities of PaaS. This security model consists of a number of tools, techniques and guidelines to mitigate and neutralize security issues of PaaS. The security vulnerabilities along with mitigation strategies were discussed to offer a deep insight into PaaS security for both vendor and client that may facilitate future design to implement secure PaaS platforms.

Mohammad SaidurRahman *et al* [19] presented a privacy-preserving service selection framework for cloud-based service systems. In the cloud-based service system, a cloud provider selects the best service from a set of services based on their Quality-of-Service (QoS) information. The QoS information of services is sensitive from the service provider's point of view. A service provider can bribe a dishonest employee of the cloud provider for taking unfair advantage during a service selection process. Therefore, it is important to execute the service selection tasks keeping QoS information private. Fully Homomorphic Encryption (FHE) scheme is used in this paper for encrypting QoS values. Service selection task is performed by the cloud provider on encrypted QoS values to ensure privacy. In order to reduce computation overhead, MapReduce model is proposed for parallel execution. They conducted several experiments to evaluate the performance of their proposed privacy preserving service selection framework using synthetic QoS dataset.

Opara-Martins *et al* [20] displayed a critical analysis of the vendor lock-in problem, from a business perspective. A survey based on qualitative and quantitative approaches conducted in this study has identified the main risk factors that give rise to lock-in situations. The analysis of our survey of 114 participants shows that, as computing resources migrate from on-premise to the cloud, the vendor lock-in problem is exacerbated. Furthermore, the findings exemplify the importance of interoperability, portability and standards in cloud computing. A number of strategies are proposed on how to avoid and mitigate lock-in risks when migrating to cloud computing. The strategies relate to contracts, selection of vendors that support standardised formats and protocols regarding standard data structures and APIs, developing awareness of commonalities and dependencies among cloud-based solutions. They strongly believe that the implementation of these strategies has a great potential to reduce the risks of vendor lock-in.

3. PROBLEM STATEMENT

- Security problems can be a big barrier to cloud computing. System servers require trustworthy security measures to different data domains according to the system servers own operating mechanism.
- Problem is constructed by filtering out those cloud providers not conforming to high-level security requirements.
- By including low-level security requirements to be used for filtering the cloud provider space and formulating the optimisation function.

3.1. BACKGROUND OF AN EXISTING TECHNIQUE

PaaS follows a model-driven approach that relies on CAMEL [16] which extensively captures various aspects in multi-cloud application lifecycle. This approach is based on a particular architecture depicted in Figure 1 comprising two main modules: Upper ware, and ExecutionWare. The functionality of these modules along with details about other components important for the success of the security solution is analysed. Further improve the security enhancement of PaaS network platform our proposed methodology adding two modules like access control and security. These two modules gives a higher security comparing to the existing techniques.

Figure 1: Block diagram of existing background

As shown in figure 1, block diagram of existing background and its modules are specified further improve its security our proposed methodology adding two more modules like access control as well as security. The detail explanations of these modules are given as subsequent sections.

4. PROPOSED METHODOLOGY

In Existing, actually preventing users from moving to the cloud comes with respect to security, which becomes more complex in multi-cloud settings. In this work mainly focused on the security of Platform-as-a-Service (PaaS) as well as the most critical security issues that were documented regarding PaaS infrastructure. This work have two main aspects: First, suitable access control on user personal data, VMs and platform services and Second planning

and adapting application deployments based on security requirements. In Fuzzybased access control to information sources is mainly realised by exploiting the CDOsecurity feature. In Security feature code was modified to map the class and packet filter for any specific permission to our own class. If the Identity Provider (IdP) has included public security information on the two main parts in the small token on which Trust based Signature elements are placed, i.e., the whole token or the assertions included, this public key is used to validate the respective signature. The block diagram of the proposed PaaS platform security is given in figure 1.

Figure 2: Block diagram of the proposed PaaS platform security Enhancement

As shown in figure 2 there are two main entry points in the PaaS to enable user's exploiting its facilities. The Service Point (SP) is a REST service which offers platform services enabling users to either run standalone lifecycle tasks or the whole adaptive application provisioning workflow. In this respect, the SP is the orchestrator for the underlying platform modules and components to support this provisioning workflow. This service is not intended to be used directly by users but is suited for programmatic interactions. The next entry point builds on the SP to enable a more interactive experience with the platform users. This is realised via the Social Network (SN), while other types of external UIs or IDEs can be used for similar purposes or even for providing added-value user support to the user (e.g., billing, analysis or model editing services). The SN, apart from providing usual social network services, it allows users to initiate application deployment workflows, check the status of deployment tasks and browse application execution histories. It also enables the sharing of knowledge in terms of application models, metric specifications, scalability rule models or interesting deployment patterns for applications similar or equivalent to the application at hand. Further improve its security enhancement fuzzy as well as trust based signature are used. PaaS domain based on three four modules detailed explanation of this is discussed as follows; Upperware, Executeware, Access control using fuzzy and Security using Trust base signature

4.1 UPPER WARE AND EXECUTE WARE:

UPPER WARE MODULE:

The Upperware obtains user application and requirement models, constructs the respective application profiles and then maps user requirements to specific deployment plans by using the Reasoner. It is also responsible for performing global adaptation for the application deployment to close the design-time adaptation loop. The Executionware retrieves and enforces the application deployment plans in a multi-cloud manner, by being able to interact with and manage those individual heterogeneous private or public cloud infrastructures involved in the deployment plan thus being able to raise the abstraction level and deal with the respective heterogeneity, while also monitors respective resource and user-specific metrics. Via monitoring it is also able to evaluate scalability rules and adapt the application deployment, if needed, in a cloud-specific manner, thus closing the local adaptation loop. Figure 3 specifies the security solution of upper ware and execution ware which is represented as follows;

Figure 3: Upper ware and Middle ware Security Solution Architecture

Upperware and Executionware modules consist of some internal components which is explained as follows;

4.1.1 PROFILER:

The Profiler has been enhanced to process security requirements posed in user models and imprint them in the application profile as well as modify the way the deployment plan problem is constructed by filtering out those cloud providers not conforming to high-level security requirements. The latter maps well to the existing component's functionality as it has to perform a pre-filtering of the provider space to facilitate accelerating the subsequent deployment reasoning process.

4.1.2 REASONER:

The Reasoner was enhanced to extend the deployment plan problem by including low-level security requirements to be used for filtering (in case of hard global constraints) the cloud provider space and formulating the optimisation function (for security-based optimisation requirements such as maximise). The extensions to the Profiler and Reasoner components actually lead to the fulfilment of the PR2 requirement as actually high- and low-level requirements are enforced before and during the reasoning process for multi-cloud application deployment.

4.1.3 ADAPTER:

The Adapter takes care of global application adaptation by considering new possible plans proposed by the Reasoner for the current application deployment and selecting one for execution if the benefits for the respective transition are appropriate. In case of security, the Adapter's selection criteria were extended to account for security metrics such that the usual trade-off between performance, cost and security is considered. For instance, the selection criteria could regard that a new deployment is possible as new more secure and less costly cloud provider offerings have been discovered so as to increase the levels of security metrics. Once a new deployment is selected, the Adapter calculates the minimum number of adaptation actions to be performed in order to transit the application to the new deployment state and then sends this action set for execution to the Executionware module.

4.2 EXECUTION WARE MODULE:

In the case of the Execution ware module, only the Execution Engine has been updated,

4.2.1 Execution Engine:

Execution engine responsible for managing VM instances and application components, by enforcing the installation of: A security library in each application VM and the respective security probes for security monitoring. As the way security properties and metrics are specified is indifferent to the one for other non-functional terms, the module's Metrics Collector and Evaluation components were not modified and still serve the purpose of collecting metric measurements and evaluating them against any SLO type. To this end, /emphPR3 requirement is partially fulfilled as security-based libraries can actually be exploited for security-based monitoring and adaptation.

4.2.2 Adaptation Engine:

The Adaptation Engine is responsible for executing security or scaling adaptation strategy workflows specified in BPMN. Each security adaptation action is executed by calling a Management component which runs the respective software library functionality at the VM on which the problematic application component resides. Scaling actions are executed via calling the Execution Engine. As such, scaling and security actions can be mixed in an adaptation workflow and we foresee complex application provisioning adaptation scenarios requiring this mixture. For instance, one application VM can be both overloaded and have a security SLO violated. In this case, we could have a composite adaptation rule indicating that the VM will be scaled-out and that both VM instances old and new should be enhanced through executing an additional security software to increase the security like protection level on them. The open-source Activity Engine was used to realise the Adaptation Engine.

4.2.3 Metrics Collector and Evaluation Assesses:

In this, the security probes sent measurements to the Metrics Collector which aggregates them and reports them to the Evaluation for local adaptation as well as to the Reasoner for global. In Evaluation assesses the respective event conditions and checks which adaptation rules are enabled. In case of reaching scalability limits or local adaptation cycles detected, the Adapter is informed in order to select the best deployment plan from those derived by the Reasoner via the measurements received. Then, global adaptation is performed by calling the Executionware with the respective action set to execute. For each adaptation rule triggered, local adaptation continues with Evaluation calling BPMN Transformer with the rule adaptation strategy as input and then sending the resulting BPMN file to the Adaptation Engine for execution. This engine runs the BPMN file and calls the Execution Engine for scaling actions and the Management component for managing the respective underlying application or security components (e.g., to start/stop a security component or re-configure an application component).

After completing the above process, using fuzzy provide access control for users. The concept include access control using fuzzy is explained as follows;

5. FUZZY BASED ACCESS CONTROL (FBAC):

Access control based fuzzy is one of the security technique that regulates who or what can view or use resources in a computing environment. Physical access control limits access to campuses, buildings, rooms and physical IT assets. Logical access control limits connections to computer networks, system files and data. Fuzzy has the following four steps which is explained as follows; Fuzzy variables, If – then rules, Aggregation and Defuzzification.

These are some steps used in fuzzy. In order to improve FBAC, first define trust and trustworthiness concepts, then introduce user trustworthiness and role's required trustworthiness parameters and describe how to use these parameters to improve user assignment (UA) and role activation in FBAC. Finally, compute these two parameters using fuzzy relation equations.

5.1. Fuzzy variables:

In our approach, we define the trust and trust worthiness concept that affords good results. Here, trust worthiness and roles required trust worthiness parameters are used for intermediate variables. To improve the user assignment (UA) and role activation in FBAC the parameters are mathematically represented using the following equation;

$$\mu_A(y) = \begin{cases} 0, & y \leq a_1 \\ \frac{y - a_1}{b_1 - a_1}, & a_1 \leq y \leq b_1 \\ 1, & b_1 \leq y \leq c_1 \\ \frac{d_1 - y}{d_1 - c_1}, & c_1 \leq y \leq d_1 \\ 0, & d_1 \leq y \end{cases} \quad (1)$$

$$\mu_A(y) = \begin{cases} 0, & y \leq a_2 \\ \frac{y - a_2}{b_2 - a_2}, & a_2 \leq y \leq b_2 \\ \frac{c_2 - y}{c_2 - b_2}, & b_2 \leq y \leq c_2 \\ 0, & c_2 \leq y \end{cases} \quad (2)$$

Then these user assignment and role activation in fuzzy based access control parameters are applied to each fuzzy variable of a rule using the functions (3) and (4) respectively.

$$f(y ; a_1, b_1, c_1, d_1) = \max \left(\min \left(\frac{y - a_1}{b_1 - a_1}, 1, \frac{d_1 - y}{d_1 - c_1} \right), 0 \right) \quad (3)$$

$$f(y ; a_2, b_2, c_2) = \max \left(\min \left(\frac{y - a_2}{b_2 - a_2}, \frac{c_2 - y}{c_2 - b_2} \right), 0 \right) \quad (4)$$

5.2. If – then rules:

If-then rule is applied to the fuzzy variables to attain the fuzzy set out. These rules have multiple inputs and the fuzzy operator (AND). Using this operator, minimal of three membership values is selected for each rule. For an example, If-then rule is taken from the table and described as;

IF user trustworthiness (UT) **AND** role's required trustworthiness (RT) **Then** determines the amount of trust is required by a user to play the role in system

5.3. Aggregation:

This is a union of all the outputs attained from all If-then rules. A new aggregate fuzzy set is generated by choosing the maximal rule estimation values using the Fuzzy logic operator (OR). This maximum fuzzy output is given as input COG for Defuzzification.

5.4. Defuzzification:

Aggregated fuzzy output set is given as input to the process of Defuzzification provide role activation to the user. For Defuzzification, Centre of Gravity method (COG) is used. Using this method, we get the single crisp value as output from the input of fuzzy sets.

$$COG = \frac{\sum \mu_A(y) * y}{\sum \mu_A(y)} \quad (5)$$

Where, $\mu_A(y)$ represents the parameter function of fuzzy based access control (FBAC). After providing access control to the user/client security is a one of the most important thing because now a days unauthorized persons easily hack user information so avoid this problem in our work, security using trust based signature. The concept include security using trust based signature is explained as follows;

6. FUZZY BASED ACCESS CONTROL REQUIREMENTS:

The requirements and principles for access controls to platform facilities are the following:

IFAC1 – unique credentials per user for single sign-on access on all resource types across all platform entry points, components and modules.

IFAC2 – a default permission set must be associated to each basic role of an organisation. An organisation's administrator can then update this set based on organisation needs and policies. This accelerates specifying permissions as a common basic role subset exists across different organisations and can be mapped approximately to the same permission sets.

IFAC3 – an organisation must control the access that other organisations' users can have, indicated from now on as external users, and over information owned by this organisation. By fulfilling this requirement, the PaaS prototype will offer multi-tenancy facilities to its users and MDDB will be transformed into a multi-tenant store with private information spaces controlled by the organisations owning them.

IFAC4 – a super administrator is needed to deal with unforeseen security issues by having full write access to the whole MDDB space. The client organisations should trust the PaaS platform instance operator for this as it represents the most suitable measure to address potential vulnerabilities, especially in case where the access to a customer organisation's resources is completely taken out of its control.

IFAC5 – user identification can be performed via both internal and external identity providers. For external authentication, this creates the need to be able to process and exploit externally certified information so as to properly identify users and map them to the roles assigned to them.

IFAC6 – FUZZY security standards, such that state-of-the-art software available for them can be exploited to realise the required authentication and authorisation functionality.

IFAC7 – re-use existing technology to speed up the security functionality development. At MDDB-side, this translates to exploiting the CDO security feature to secure the access to the CDO repository. These are the requirements used in access control based on fuzzy.

7. SECURITY USING TRUST BASED SIGNATURE:

In this, the Security feature code was modified to map the class and packet filter for any specific permission to our own class. If the Identity Provider (IdP) has included public security information on the two main parts in the small token on which Trust based Signature elements are placed, i.e., the whole token or the assertions included, this public key is used to validate the respective signature.

In order to obtain accurate trust of each node in the network, we combine these two parts together as follows.

$$Trust = T1Trust^S + T2Trust^N \quad (6)$$

Where, $Trust^S$ is the trust value from direction interactions between a trust or and a trustee; $Trust^N$ is the trust value from recommendations. Here the trust based generation as well as the signature re-generations are represented as follows Algorithm 1 and Algorithm 2;

Step1: public key based random numbers $(g1, g2)$ and $G1$ is large prime number
Step2: Select random number $\alpha, \beta, \delta \in G1$
Step3: Compute helper values $T1 = g1 \oplus \alpha, T2 = g2 \oplus \beta$
Step4: Compute support value $S = (T1 + T2) / T1 * T2$
Step5: Compute a challenge value $C \in Z^*$ using $C \leftarrow H(T1Trust^S + T2Trust^N)$
Step6: Output the signature $\sigma \leftarrow (\alpha, \beta, C, T1, T2)$.

Algorithm 1. Signature generation

Step1: Re-derive $\check{T}1, \check{T}2$
Step2: Compute helper values $\check{T}1 = T1 \oplus \alpha, \check{T}2 = T2 \oplus \beta$
Step3: Compute support value $\hat{S} = (\check{T}1 + \check{T}2) / (\check{T}1 * \check{T}2)$
Step4: Compute a challenge $\hat{C} \leftarrow H(\hat{S}, \check{T}1, \check{T}2)$
Step5: Check that the challenge $[C = \hat{C}]$
Step6: If it is, accept. Otherwise, reject.

Algorithm 2. Signature Verification

7.1 SECURITY USING TRUST BASED SIGNATURE REQUIREMENTS:

The requirements to support the security using trust based application provisioning are as follows:

TBSR1 – ability to express both high-level and low-level security requirements.

TBSR2 – ability to match all types of security requirements and apply them to the deployment reasoning process.

TBSR3 – exploitation of state-of-the-art software libraries for trust based security monitoring.

TBSR4 – ability to model security rules to drive the trust based behaviour of the application.

TBSR5 – ability to enforce trust based actions on the infrastructure on which the application resides.

8. CDO SERVER, CDO REPOSITORY AND DATABASE:

8.1 CDO Server:

CDO Server is a service on top of CDO Repository accepting requests from CDOClients to perform model-based management actions. As such, the CDO Client can be an internal component of any PaaS module / component that needs to manage models over the CDORepository.

8.2 CDO Repository:

The main building block of a CDO repository is split into two layers, the generic repository layer that client applications interact with and the database integration layer that providers can hook into to integrate their data storage solutions with CDO. A number of such integrations already ship with CDO, making it possible to connect a repository to all sorts of databases.

Figure 4:Architecture of CDO Repository

The above diagram illustrates the major building blocks of a CDO repository. The CDO technology was used to construct the DB CDO Repository which enables EMF model management and persistence. This repository exhibits various features, including transactionality, lazy loading and fault-tolerance. It can also exploit a multitude of different underlying database management systems for model persistence.

9. RESULT & DISCUSSION:

Experimental Result:

Our experiments are conducted in a Cloud Sim with JAVA. They were performed on a PC with Windows XP Operating system at 2 GHZ dual core PC machine with 4 GB main memory running a 64-bit version of Windows 2007. In the experimental environment, the

hardware consists of six servers with the following configuration: a CPU(Intel Xeon 2.8 GHz, RAM 1 GB) and hard disk space of 1.2TB with RAID5 backup. The experiment result of the proposed work is described below:

Comparative Analysis:

The experiment is implemented for CAMEL [16] and ECCTS [17], respectively. To compare the performance in terms of service profit, user satisfaction and server utility, we consider factors including cost, waiting time and accuracy. So, the resource consumption (which is related to cost), operating time (which is related to waiting time) and correct communication percent (which is related to the correction rate) are analysed in the experiment. The service profit, user satisfaction and service utility in CAMEL and ECCTS are shown in Figs. 5 to 13 respectively.

Figure 5: Comparative analysis of service profit in Monday to Friday 1

As shown in figure 5 our proposed techniques are compared with CAMEL and ECCTS, compared to the above existing techniques our proposed methodology gives a high quality outcomes. Service profit of Monday to Friday 1's are specified in the above figure. From Fig. 5, we can discover that the service profit in Fuzzy&TBS obviously precedes the service profit in CAMEL and TETS, because the system in proposed methodology shares the common infrastructure, and the service in fuzzy&TBS is smoother than that in CAMEL and TETS. Meanwhile, we also find that the service profit in the period of holidays has a higher profit than in Monday and Friday period because, during workdays, a telecommunication service is busier.

Figure 6: Comparative analysis of user satisfaction in Monday to Friday 2

As shown in figure 6 our proposed techniques are compared with CAMEL and ECCTS, compared to the above existing techniques our proposed methodology gives a high quality outcomes. Comparative analysis of user satisfaction in Monday to Friday 2 are specified in the above figure. From Fig. 6, we also can find that the user satisfaction in fuzzy&TBS precedes the user satisfaction in CAMEL and TETS. This is because the applied service in CAMEL and TETS is busier than that in fuzzy&TBS. Moreover, there are more users using the telecommunication service in the Monday and Friday period than in holidays and weekend period. So, users have a higher degree of satisfaction during holidays than workdays.

Figure 7: Comparative analysis of service utility in Monday to Friday 3

From Fig. 7, we can similarly find that the service utility in fuzzy&TBS precedes the service utility in CAMEL and TETS. We also can find that the line in fuzzy&TBS is more

placid. So, the novel security model is more effective than a traditional telecommunication service. According to the above analysis of the security and the advantages, the novel security mechanism can ensure the effectiveness and safety of a telecommunication service. From analysing figure 7 our proposed methodology gives a high quality outcomes.

Figure 8: Comparative analysis of service profit in Saturday to Sunday 1

From Fig. 8, we can discover that the service profit in Fuzzy&TBS obviously precedes the service profit in CAMEL and TETS, because the system in proposed methodology shares the common infrastructure, and the service in fuzzy&TBS is smoother than that in CAMEL and TETS. Meanwhile, we also find that the service profit in the period of holidays has a higher profit than in Saturday and Sunday period because, during workdays, a telecommunication service is busier. From analysing figure 8 our proposed methodology gives a high quality outcomes.

Figure 9: Comparative analysis of user satisfaction in Saturday to Sunday 2

As shown in figure 9 our proposed techniques are compared with CAMEL and ECCTS, compared to the above existing techniques our proposed methodology gives a high quality outcomes. Comparative analysis of user satisfaction in Saturday to Sunday 2 are specified in the above figure. From Fig. 6, we also can find that the user satisfaction in fuzzy&TBS precedes the user satisfaction in CAMEL and TETS. This is because the applied service in CAMEL and

TETS is busier than that in fuzzy&TBS. Moreover, there are more users using the telecommunication service in the Saturday and Sunday period than in holidays and weekend period. So, users have a higher degree of satisfaction during holidays than workdays.

Figure 10: Comparative analysis of service utility in Saturday to Sunday 3

From Fig. 10, we can similarly find that the service utility in fuzzy&TBS precedes the service utility in CAMEL and TETS. We also can find that the line in fuzzy&TBS is more placid. So, the novel security model is more effective than a traditional telecommunication service. According to the above analysis of the security and the advantages, the novel security mechanism can ensure the effectiveness and safety of a telecommunication service. From analysing figure 10 our proposed methodology gives a high quality outcomes.

Figure 11: Comparative analysis of service profit in holiday 1

From Fig. 11, we can discover that the service profit in Fuzzy&TBS obviously precedes the service profit in CAMEL and TETS, because the system in proposed methodology shares the common infrastructure, and the service in fuzzy&TBS is smoother than that in CAMEL and TETS. From analysing figure 8 our proposed methodology gives a high quality outcomes.

Figure 12: Comparative analysis of user satisfaction in holiday 2

As shown in figure 12 our proposed techniques are compared with CAMEL and ECCTS, compared to the above existing techniques our proposed methodology gives a high quality outcomes. Comparative analysis of user satisfaction in holiday 2 are specified in the above figure. From Fig. 12, we also can find that the user satisfaction in fuzzy&TBS precedes the user satisfaction in CAMEL and TETS. This is because the applied service in CAMEL and TETS is busier than that in fuzzy&TBS.

Figure 13: Comparative analysis of service utility in holiday 3

From Fig. 10, we can similarly find that the service utility in fuzzy&TBS precedes the service utility in CAMEL and TETS. We also can find that the line in fuzzy&TBS is more placid. So, the novel security model is more effective than a traditional telecommunication service. From analysing figure 13 our proposed methodology gives a high quality outcomes.

CONCLUSION:

This paper presented a security enhancement for PaaS platform. To improve the platform security our proposed methodology uses fuzzy as well as trust based signature. Here, the access control using fuzzy and the same time security using trust based signature. In our work, there are two main entry points in the PaaS to enable user's exploiting its facilities. Service Point (SP) is a REST service which offers platform services enabling users to either run standalone lifecycle tasks or the whole adaptive application provisioning workflow. The next entry point builds on the SP to enable a more interactive experience with the platform users. This is realised via the Social Network (SN), while other types of external UIs or IDEs can be used for similar purposes or even for providing added-value user support to the user like billing, analysis or model editing services. Further improve its security enhancement fuzzy as well as trust based signature are used. Finally, the experimental results will show that our proposed method outperforms other existing methods in terms of various performance measures.

Authors Contributions: All Authors are equally Contributed and all Authors have read and agreed to the published version of the manuscript.

Funding: This Research has no funding by any Organization or Individual

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

REFERENCES

- [1] Odun-Ayo, Isaac, M. Ananya, Frank Agono, and Rowland Goddy-Worlu. "Cloud computing architecture: A critical analysis." In 2018 18th International Conference on Computational Science and Applications (ICCSA), pp. 1-7. IEEE, 2018.
- [2] Golani, Sukaran, and SurajSahal. "Technical survey on cloud computing and its security issues." (2019).
- [3] Kumar, Jaydip. "Cloud Computing Security Issues and Its Challenges: A Comprehensive Research."
- [4] Chauhan, Sameer Singh, Emmanuel S. Pilli, R. C. Joshi, Girdhari Singh, and M. C. Govil. "Brokering in interconnected cloud computing environments: A survey." *Journal of Parallel and Distributed Computing* 133 (2019): 193-209.
- [5] Kim, Donghoon, Henry E. Schaffer, and Mladen Alan Vouk. "About PaaS security." *International Journal of Cloud Computing* 6, no. 4 (2017): 325-341.
- [6] Gesvindr, David, and BarboraBuhnova. "PaaSArch: Quality Evaluation Tool for PaaS Cloud Applications Using Generated Prototypes." In 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), pp. 170-173. IEEE, 2019.
- [7] Dalvi, Mr Arvind, and S. D. Satav. "EDLC: Enhanced Data Leakage Controller for Secure Multi-Cloud Storage."

- [8]Almutairy, Nadiah M., Khalil HA Al-Shqeerat, and Husam Ahmed Al Hamad. "A Taxonomy of Virtualization Security Issues in Cloud Computing Environments." *Indian Journal of Science and Technology* 12 (2019): 3.
- [9] Mouratidis, Haralambos, Shaun Shei, and Aidan Delaney. "A security requirements modelling language for cloud computing environments." *Software and Systems Modeling* (2019): 1-25.
- [10] Cai, Fangbo, Nafei Zhu, Jingsha He, Pengyu Mu, Wenxin Li, and Yi Yu. "Survey of access control models and technologies for cloud computing." *Cluster Computing* 22, no. 3 (2019): 6111-6122.
- [11] Bhajantri, Lokesh B., and Tabassum N. Mujawar. "A Comprehensive Review of Access Control Mechanism Based on Attribute Based Encryption Scheme for Cloud Computing." *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)* 11, no. 3 (2019): 33-52.
- [12] Xu, Shuangshuang, and Hongliang Zhu. "A authentication and access authorization mechanism on the PaaS platform." (2019).
- [13] El Sibai, Rayane, Nader Gemayel, Jacques BouAbdo, and Jacques Demerjian. "A survey on access control mechanisms for cloud computing." *Transactions on Emerging Telecommunications Technologies* (2019): e3720.
- [14]Kumar, Sunny, AditiSinghal, and AnkurDumka. "Analysis of Cloud Security Framework Using Blockchain Technology." Available at SSRN 3383151 (2019).
- [15]Dey, Himel, Rifat Islam, and HossainArif. "An Integrated Model To Make Cloud Authentication And Multi-Tenancy More Secure." In 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 502-506. IEEE, 2019.
- [16] Kritikos, Kyriakos, Tom Kirkham, BartoszKryza, and Philippe Massonet. "Towards a security-enhanced PaaS platform for multi-cloud applications." *Future Generation Computer Systems* 67 (2017): 206-226.
- [17] Shen, Jinan, DeqingZou, Hai Jin, Bin Yuan, and Weiqi Dai. "A domain-divided configurable security model for cloud computing-based telecommunication services." *The Journal of Supercomputing* 75, no. 1 (2019): 109-122.
- [18] Yasrab, Robail. "MPSM: Multi-prospective PaaS Security Model." *arXiv preprint arXiv:1804.04731* (2018).
- [19] Rahman, Mohammad Saidur, Ibrahim Khalil, AbdulatifAlabdulatif, and Xun Yi. "Privacy preserving service selection using fully homomorphic encryption scheme on untrusted cloud service platform." *Knowledge-Based Systems* 180 (2019): 104-115.
- [20] Opara-Martins, Justice, Reza Sahandi, and FengTian. "Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective." *Journal of Cloud Computing* 5, no. 1 (2016): 4.