Access Control Mechanism for Prevention of Insider Threat in Distributed Cloud Environment

Thesis submitted in fulfilment for the requirement for the Degree of

DOCTOR OF PHILOSOPHY

By

GAURAV DEEP

Department of Computer Science Engineering and Information Technology,

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Waknaghat, Solan-173234, Himachal Pradesh, India

July 2023

DECLARATION OF SCHOLAR

I hereby declare that the work reported in the Ph.D. thesis entitled "**Access Control Mechanism for Prevention of Insider Threat in Distributed Cloud Environment**" submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of my work carried out under the supervision of **Dr. Jagpreet Sidhu** and **Dr. Rajni Mohana**. I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my Ph.D Thesis.

Gaurav Deep,

Department of Computer Science & Engineering and Information Technology,

Jaypee University of Information Technology,

Waknaghat -173234, India.

Date: 20th July 2023

SUPERVISOR'S CERTIFICATE

This is to certify that the work in the thesis entitled "**Access Control Mechanism for Prevention of Insider Threat in Distributed Cloud Environment**" submitted by **Gaurav Deep** is a record of an original research work carried out by him under our supervision and guidance in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering in the Department of Computer Science and Engineering, **Jaypee University of Information Technology, Waknaghat, India.** Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Date: 20th July 2023

Dr. Jagpreet Sidhu,

Associate Professor (Computer Engineering),

School of Technology Management & Engineering,

Narsee Monjee Institute of Management Studies(NMIMS),Chandigarh-160014, India.

Dr. Rajni Mohana,

Associate Professor,

Department of Computer Science & Engineering and Information Technology,

Jaypee University of Information Technology, Waknaghat -173234, India.

ACKNOWLEDGEMENT

First of all, I would like to express my deep respect and gratitude towards my guides Dr. Jagpreet Sidhu in School of Technology Management & Engineering , NMIMS, Chandigarh and Dr. Rajni Mohana in the Department of Computer Science & Engineering and Information Technology, JUIT, Solan for their support throughout this research work. Their technical knowledge and expertise in cloud security introduced me to the field of Access Control. At every step during this journey, they have helped me learn, correct, and upgrade my knowledge. Without their invaluable advice and assistance, it would not have been possible for me to complete this thesis. It is a great honour to work under their supervision.

I would like to express my gratitude to our Honorable Vice-Chancellor, Prof. (Dr.) Rajendra Kumar Sharma and Dean (Academic and Research) Prof. (Dr.) Ashok Kumar Gupta to promote the research and facilitate resources in the institution. I would also like to thank Prof. Dr. Vivek Kumar Sehgal, Head Department of CSE / IT, for constant guidance, research facilities and resources to carry out my research work. I would also like to thank my doctoral committee members, Dr. Rajinder Sandhu, Dr. Rakesh Kanji and Prof. Dr. Rajiv Kumar, for their valuable feedback, critical reviews during presentations, and time-to-time help.

I am also grateful for the endless support from the JUIT, Waknaghat. In particular, I thank all the teaching, library, and non-teaching staff at the Department of Computer Science & Engineering and Information Technology, JUIT, who has been extremely helpful on numerous occasions. I thank my fellow PhD friends for their consistent help and valuable discussion. Last but not least, I would like to thank my parents, my wife, my kids, and my beloved ones for encouraging me in the first place and supporting me spiritually throughout my life.

Needless to say, errors and omissions are solely mines.

Gaurav Deep

ABSTRACT

Technological advancements bring many benefits along with solutions to research problems. Cloud computing has revolutionized data storage and ensured availability of data and storage on demand. Its success and acceptance depend on the advantages it provides over its disadvantages. Employees in a cloud service environment manage data storage, movement, user authorisation policies, etc. Authority given as privileges for managing cloud data to employees has become an insider threat that directly impacts user confidence, company business and company reputation. This thesis aims to provide an access control mechanism to prevent insider threats at two levels in the distributed cloud environment. The issue of data protection in the cloud environment is also an important issue raised by various researchers in the related literature, which was also been handled at the architectural level in this thesis. Various techniques have been proposed in the literature to handle the insider threat, but they remain non-effective as they store insider's activity analysis in system logs, and the insider is aware of them. This thesis proposes a blockchain-based robust technique for authorisation of log files of insiders in the cloud environment. Insider authentication and activity details are stored in the blockchain. Robustness, distributed ledger, immutability and other benefits do not allow insiders to change these system logs. The proposed solution resolves the Insider threat issue by providing access control to system logs. This technique is tested and validated using a scyther formal system tool. The result ascertains that the proposed system is efficient and successfully mitigates various insider threats. The working of the protocol is also verified based on the four claims, and scyther proved that the proposed protocol is robust enough for real-time implementations. Its operational competence has also been tested in python by creating blockchain nodes for multiple users. The issue of achieving better Authorisation control at the architecture level in the cloud environment has also been addressed. Scalability remains the central issue in the existing work. As the number of PEPs increases with one PDP, its overall performance affects handling requests. A distributed architecture for better insider Authorisation control in the cloud environment with multiple PEP–PDP servers is proposed to achieve significantly better results in scalability and performance. These results have also been validated statistically in the ANOVA test, which proves that the proposed system is highly efficient and successful compared to the existing multiple PEP single PDP architecture. Insiders manage the PEP and PDP servers, so it is required to track their activity. Better Insider

Authorisation in the proposed distributed architecture is acheived in the cloud environment with the use of a Blockchain server / Blockchain Module to store all the messages between PEP and PDP. The proposed protocol's working is verified and tested based on the four claims, alive, nisynch, secret, and commitment. Alive means to achieve the intended communication with some events. Nisynch means non-injective synchronization, which ensures that the intended sender sends all the messages the receiver receives in a synchronized manner. Commitment is a promise made by one party to the other. Confidentiality of user data is achieved by using secret. Scyther proved that the proposed protocol is robust enough for real-time implementations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AA	Attribute Authority
ABAC	Attribute-Based Access Control
AC	Access Control
ACLs	Authorisation Checklists
ACMM	Access Control Management Model
ACP	Access Control Policy
AnA	Authentication and Authorisation
AnAA	Authentication and Authorisation Architecture
ANOVA	Analysis of Variance
APL	Authorisation Policy
APLs	Authorisation Policies
BBM	Biometric Behavior Masquerading
BBN	Bayesian Belief network
BN	Blockchain Node
BNM	Blockchain Node Mining
BS	Blockchain Server
CAC	Cloud Access Control
CC	Cloud Computing
CD	Cloud Database
CIA	Confidentiality, Integrity, and Availability
CM	Conceptual Model
CRA	Cloud Reference Architecture
CSP	Cloud Service Provider
CSPr	Cloud Service Privacy
CSS	Cloud Service Security
DA	Data Access
DB	Database
DC	Dependency Checker
DL	Deep Learning
DM	Deployment Models
DM	Decision-Maker
DSS	Decision Support System
ECA	Event – Condition - Action
ECC	Elliptic Curve Cryptography
GSDN	Gargoyle Software Defined Network
G-SIR	Geo-Social Insider Threat
HTTP	Hypertext Transfer Protocol
IA	Insider Attack
IACT	Insider Activity
IMDB	In-Memory Database
IoT	Internet of Things
ITDU	Internal Threat Detection Unit
ITH	Insider Threat
KBs	Knowledge Base Store
LPWAN	Low Power Wide Area Network
MACA	Multi-Factor Cloud Authentication

MAS	Multi-Agent System
MQTT	Message Queuing Telemetry Transport
MSN	Medical Smartphone Network
MSS	Message Security System
NIST	National Institute of Standards and Technology
PAP	Policy Administration Point
PBNM	Policy-Based Network Management
PDPaaS	PDP as a Service
PEP	Policy Enforcement Point
PEPaaS	PEP as a Service
PIP	Policy Information Point
RBAC	Role-Based Access Control
RDF	Resource Description Framework
SD	Service Deployment
SD	System Dynamics
SHA	Secure Hashing Algorithm
SI	Secure Information
SO	Service Orchestration
TAAC	Topology-Aware Access Control
TCP/IP	Transmission Control Protocol/ Internet Protocol
UB	User Behaviour
VANET	Vehicle Adhoc Network
VAR	Voltage Adaptation Ritual
WBAC	Work-Based Access Control
EACF	Extensible Access Control Framework
PDP	Policy Deciding Point

LIST OF NOTATIONS

Notation	**Description**
K_{Ai} ,$K^{-1}{}_{Ai}$	Public Key and Private Key of an insider entity A_i
(m) K_{Ai}	Message m is encrypted using the Public Key of entity A_i
(c) $K^{-1}{}_{Ai}$	Cypher text c is decrypted using the Private Key of entity A_i
H(m)	One way Hash of m
N_i	A unique random number called Nonce is generated by the entity N_i
BC_{iv}	Current index value in Blockchain list
BC_{liv}	Last index value in Blockchain list
H_v	Hash value in Blockchain
T_v	Timestamp Value in Blockchain
N_v	Nonce value in Blockchain list
I_{aa}	Insider authentication and activity

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The influence of Cloud Computing (CC) has reached every field of life. Everybody is interested in getting benefits from CC to reduce the initial investment cost attain; wider reachability, traffic management, and resource management all these hold significant benefits [1]. Many old and new technologies are becoming part of CC [2-3]. CC has revolutionized computing on demand. Businesses, medical, engineering, communication, entertainment, and research, are major areas now governed by CC [4-5]. Every new research area complements its benefits.

Cloud Service Provider (CSP) is responsible for every service they provide to their Cloud Service Client (CSC). U.S. National Institute of Standards and Technology (NIST) has provided a Cloud Reference Architecture (CRA) for CSP, according to which various services and Deployment Models (DM) are to be provided [6]. According to this CRA, CSP has to play a pixotel role in Service Deployment (SD), Service Orchestration (SO), Cloud Service Security (CSS), and Cloud Service Privacy (CSPr), as shown in Figure 1.1.

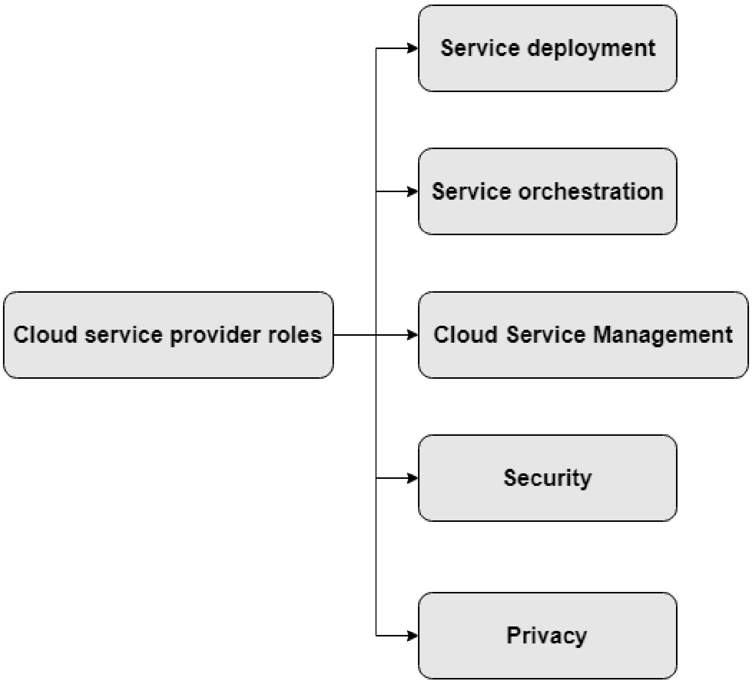


Figure 1.1: Cloud service provider roles according to NIST

The technological revolution has happened with the evolution of CC in every area, providing benefits in services and flexibilities. Moving a computerized database (DB) to a CC environment facilitates enhances more comprehensive access and availability [7]. The principle of Confidentiality, integrity, and availability (CIA) also applies to Cloud Database (CD) [8]. Confidentiality means Secure Information (SI) is only accessible to the authorised individual on the cloud; integrity means SI or system is also accurate and complete on the cloud. SI is always accessible on the cloud whenever needed [9]. There are many other issues associated with the CIA on the cloud, as shown in Figure 1.2.

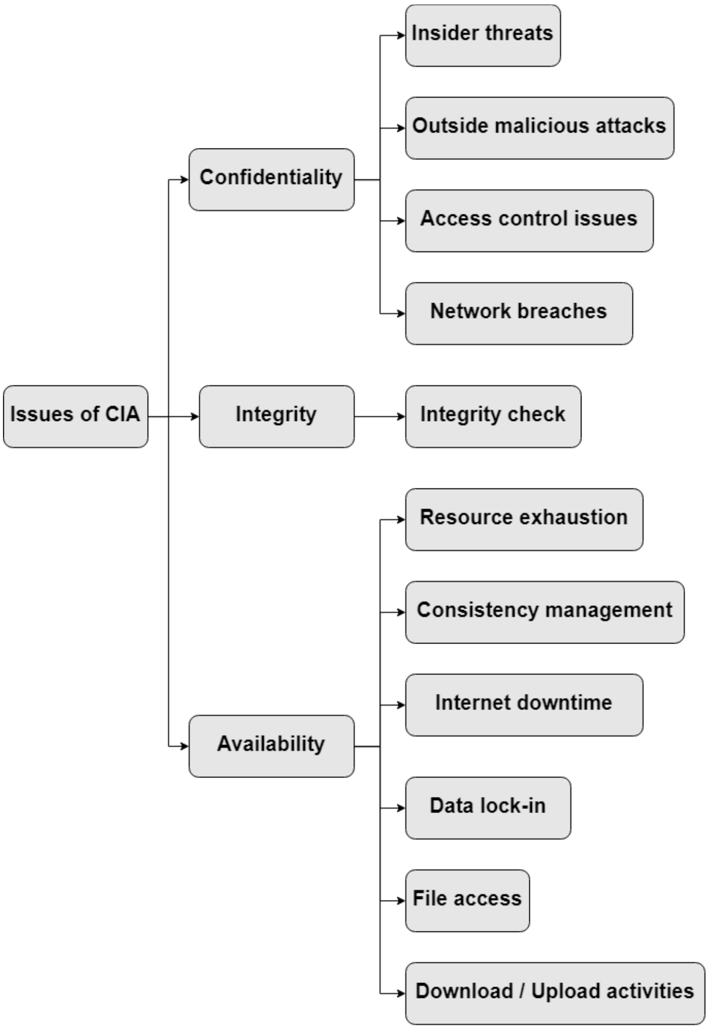


Figure 1.2: Issues of confidentiality, integrity, and availability in the CC

Insider Threat (ITH) is a threat from an insider (employee) of the CSP. Many companies didn’t report insider attacks due to fear of loss of reputation. Even then, there is a long history of Insider Attack (IA) [10]. Security and privacy are crucial concerns which affect other roles also. The users' data and information are in the safe custody of CSP under whose responsibility [11].

In distributed computing, a Single task is divided among multiple autonomous systems that may be located apart. In contrast, in Parallel computing, a single system may consist of multiple processing units running parallel .These issues can become critical on CC as several heterogeneous and homogenous hypervisors operate parallel [12]. Multiple hypervisors are parallel running on the cloud. User data migrates from one hypervisor to another as demand increases or decreases. Keeping data in a safe state becomes increasingly tricky. Here, the role of access control comes in as shown in Figure 1.3.

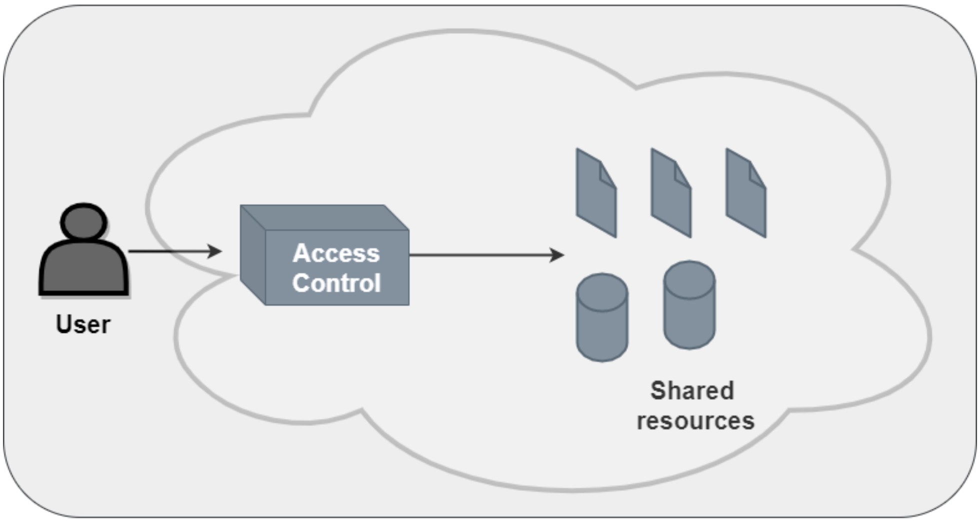


Figure 1.3: Role of access control

Access Control (AC) is a security function that protects shared resources against unauthorised access. The distinction between authorised and unauthorised access is made according to an Access Control Policy (ACP) [13].

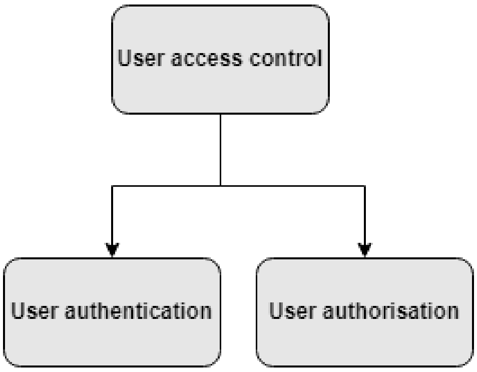


Figure 1.4: Access control functions

AC consists of authentication and authorisation, as shown in Figure 1.4. Authentication is verifying an entity's identity, given its credentials. The entity could be in the form of a person, a computer, a device, or a group of network computers [14]. An Authorisation represents the right granted to a user to exercise an action (e.g., read, write, create, delete, and execute) on particular objects [14].

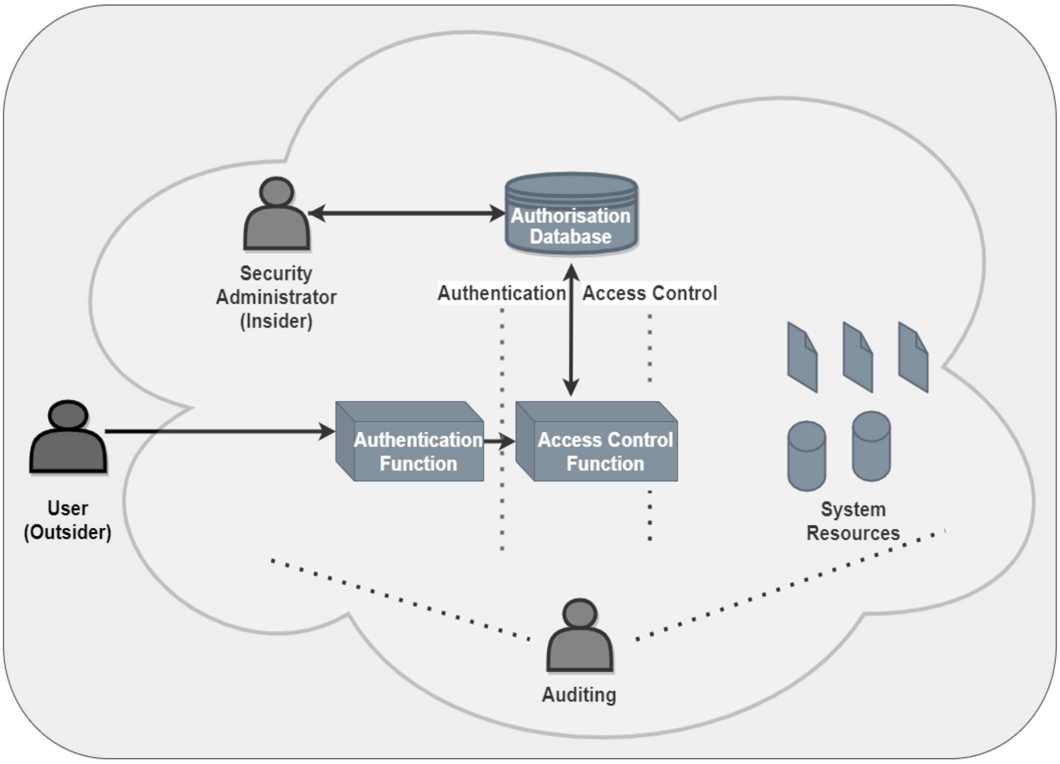


Figure 1.5: Stakeholders in CAC

Various Stakeholders in Cloud Access Control (CAC) are shown in Figure 1.5. CSP performs the Authentication and Authorisation (AnA) function to achieve better user data control. The security administrator keeps track of all activities going on their side. It has the highest level of privileges to apply the best mechanism to provide its customers with the best possible services and security [15]. When functions of AC are moved onto CC, these are controlled by employees of the CSP.

Attacks not only happen from outside the CSP only, but they can also occur from within. This type of attack is known as an IA. It is one of the most dangerous attacks where the CSP employee already knows the type of authentication technique, an encryption technique, and user authorisation employed [16]. The CAC mechanism is required to prevent IA and the CAC mechanism for its customers acting as outsiders to the system preventing outsider attacks.

1.2 MOTIVATION

According to the nucleus cyber 2019 ITH report [17], 70 % of attacks were IA. The U.S. state of cybercrime also surveyed in 2016 [18] that 27% of electronic crimes were suspected to be caused by insiders. This kind of attack is most challenging to detect as some insider plans of executes these attacks who is aware of company policies. By the time companies comes to know about IA, lots of damage has already been done. This may be why significant IA happened, as shown in Table 1.1.

Table 1.1: List of IA incidents and their effects on organisations

Year	Insider attack	Effect of attack
1979-2006	Boeing: The Nation-State Spy [19]	Critical data was compromised for many years.
2011	R.S.A.: Employees Fall for Phishing Attacks [19]	40 million employees' data was compromised.
2016	Sage: Unauthorised Employee Access [19]	250 Business customer's data compromised.
2016-2017	Anthem: Employee Data Exfiltration [19]	Record of 18000 Medicare members compromised.
2018-2020	Cisco Cloud data compromised [20]	15000 WebEx customers were affected.
2019	Microsoft customer support database [20]	200 million customers' data was compromised.
2019	Capital One Data Breach [20]	100 million customer details compromised.
2020	Marriott Guest details [20]	5 million guests’ details were compromised.
2020	General Electric Trade secret [20]	Millions of U.S. Dollars loss.
2020	Twitter accounts [20]	Several private and corporate accounts were compromised.

Critical data was compromised in the Boeing attack between 1979-2006. Data of forty million employees was compromised in the R.S.A attack that occurred in 2011. An IA compromised business customers’ (250) data in 2016. Record of 18000 Medicare members was compromised in 2016-2017 on the anthem. Cisco also lost 15000 WebEx customers’ data in 2018-2020. Microsoft's customer support DB also lost 200 million customer data in 2019.

In the same year, 100 million customer data was compromised in an attack on capital one data breach. Details of 5 million guests were compromised in the attack of Marriott DB in the year 2020. In the same year, two significant attacks also happened; millions of U.S. dollars were lost during the attack on General Electric Trade Secret, and in the second attack, several private and corporate account details were compromised during an attack on Twitter.

Insider Activity (IACT) analysis can be done to detect IA. Detection of this attack becomes problematic when there is no IACT record [21-25]. Various techniques are available, like BBM, physical traits, cyber behaviour information theft, communication behaviour collusion, and psychosocial behaviour sabotage. These techniques help in understanding IACT for future attempts to attack.

User APLs are designed to allow the window through which Data Access (DA) is allowed in a controlled manner over CC [26-28]. Once these APLs are framed and written in the computer language, they can be applied. Every CSP decides the techniques used for the user Authentication and Authorisation Architecture (AnAA). Applying the best available AnAA does not ensure that attacks do not happen on the user data.

AC architecture provides a straightforward way to prevent various attacks and data loss. Authorisation Policy (APL) clearly defines how much is allowed to access which user profile. Implementing these Authorisations requires Policy Deciding Point (PDP) and Policy Enforcement Point (PEP) servers [29-30]. PDP’s working capacity can be managed by associating the PEP server.

Two separate servers are used because PEP server accesses all APLs from the PDP Server for future access requests. Whenever a new request is received, it checks for existing copies of the set of policies. If the request matches with the policy, the corresponding action is taken; otherwise, it is forwarded to PDP for a new policy. It has been noticed that several shortcomings have been noticed in the detection of IA, such as:-

1. Insiders can access user cloud data.

2. Privileges are known to insiders to alter the authorisation record.

3. Existing cloud authorisation architecture cannot handle the growing number of insider requests.

4. Absence of storing inter-server message system in authorisation architecture for better insider activity control.

1.3 OBJECTIVES

This work aims to provide a complete CAC mechanism to prevent ITH, which can work in a distributed cloud environment. Objectives framed are given below.

- To propose a robust authorisation technique for log files of insiders in cloud environment.
- To develop and deploy distributed PEP-PDP architecture for access control in cloud environment.
- To design and develop a Message Security System (MSS) in distributed PEP-PDP architecture in cloud environment.

1.4 STRUCTURE OF THE THESIS

The entire work on the CAC mechanism to prevent ITH for distributed cloud environment is organized in the following chapters.

Chapter 1: This chapter contains an introduction on access control, IA on cloud, motivation and the objective of the work.

Chapter 2: Existing techniques and their limitations of CAC on handling IA are discussed in this literature chapter.

Chapter 3: This chapter addresses the first objective, proposing a robust authorisation technique for log files of insiders in a cloud environment using blockchain. The proposed technique is tested and validated in the scyther formal validation tool and implemented in python. Results prove its robustness.

Chapter 4: This chapter addresses the second objective, developing and deploying a distributed PEP-PDP architecture for CAC for cloud environment. The proposed architecture is tested in a simulation environment against the existing centralized PEP-PDP architecture on various parameters. Results show proposed architecture performs significantly better than the existing architecture and can handle large insider requests.

Chapter 5: This chapter address the third objective, designing and developing a Distributed message security system with PEP-PDP architecture for cloud environment. This proposed system stores all the communication between PEP-PDP as message data in the Blockchain Node (BN) at the Blockchain Server / Blockchain Module. This system also provides a tracking mechanism to detect IA.

Chapter 6: This chapter concludes the work and lists the direction toward future research work.

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

The growth of new communication technologies has helped achieve internet penetration of more than 63% by 2020, according to data [31]. Around 27000 new users get connected to the internet every hour [32]. Handling this increasing internet access is difficult; more cloud servers are required. Almost one-third of the cloud market share was occupied by amazon's web service [33].

Security issues can be seen from two aspects at CSP. Security of consumer data be provided from outside the service provider's premises, i.e. man-in-the-middle attacks, distributed denial-of-service attacks, IoT-based attacks, guessing attacks, etc. [34]. Another type of attack happens from within the premises known as IA, where employees of CSP perform data theft [35]. Various security mechanisms are available and applied by CSP to prevent these types of attacks, such as data encryption, packet filters, firewalls, intrusion detection, and prevention systems [36].

CAC covers user authentication and authorisation. User authentication means access to data is allowed to only its intended user, and Authorisation controls the amount of data allowed. User authentication consists of user identification details along with its password. This authentication type is known as single-factor authentication, where only a username and password are required [37].

When two components are required to verify identity, like an ATM transaction, the ATM data strip and user's identification number are verified as two-factor authentication [37]. More than two factors can be involved in verifying identities like voice, signatures, fingerprints, and location details; this type of authentication is known as multifactor authentication [38]. Authentication techniques and Authorisation architecture are discussed in this chapter.

2.2 AUTHENTICATION AND ACTIVITY TRACKING TECHNIQUES FOR INSIDER

Categorizing the user over the cloud into insider vs outsider is done by authentication. Related studies concerning authentication and activity tracking.

2.2.1 INSIDER ACTIVITY TRACKING BASED ON BIOMETRICS, PHYSICAL, CYBER, COMMUNICATION AND PSYCHOSOCIAL BEHAVIOUR

The activity of insiders can be tracked from system logs and other techniques. These techniques help in ITH detection, as shown in Figure 2.1. Biometric Behavior Masquerading (BBM) detection is based on an insider's cyber activity, recording their mouse strokes, keyboard strokes, application-level behaviour, system-level behaviour, graphic user interface interaction, and recording his activity on file search. Physical traits are based on eye colour, face detection, and thumbprints [39-40]. Cyber behaviour information theft detection is based on insider activity of printing, web browsing, device usage, login behaviour, file access, and download/upload activities. Communication behaviour collusion means his activities of email, instant messages, telephone, and file-sharing / transfers. Psychosocial behaviour towards his colleagues and authorities. These detection and prevention schemes can only work when an IACT record is available. Research work basis on these techniques is discussed below.

Harilal et al. [41] proposed a gamified competition to create an insider dataset based on actual behaviour. Each member participating in this competition acts as a company salesman. These teams were competing with each other. The task of these team members was to contact customers and collect maximum points. The proposed game ran in three predefined periods regular, wild card and score reporting. Insider behaviour dataset was made based on mouse traces, keystrokes, host monitor, network, emails, logon/logout activities and psychological questionnaire, which helped identify masquerade and traitors working in a company.

Voris et al. [42] proposed an active authentication using file system decoys and user behaviour (UB) modelling. Continuous authentication was done based on the UB model. Modelling was done based on the user's unique behaviour while interacting with a personalized computing environment. It also captured his way of file system access, process and networking access to decoy files. The cognitive fingerprint of each user was made, which helped detect the attacker with 95% accuracy.

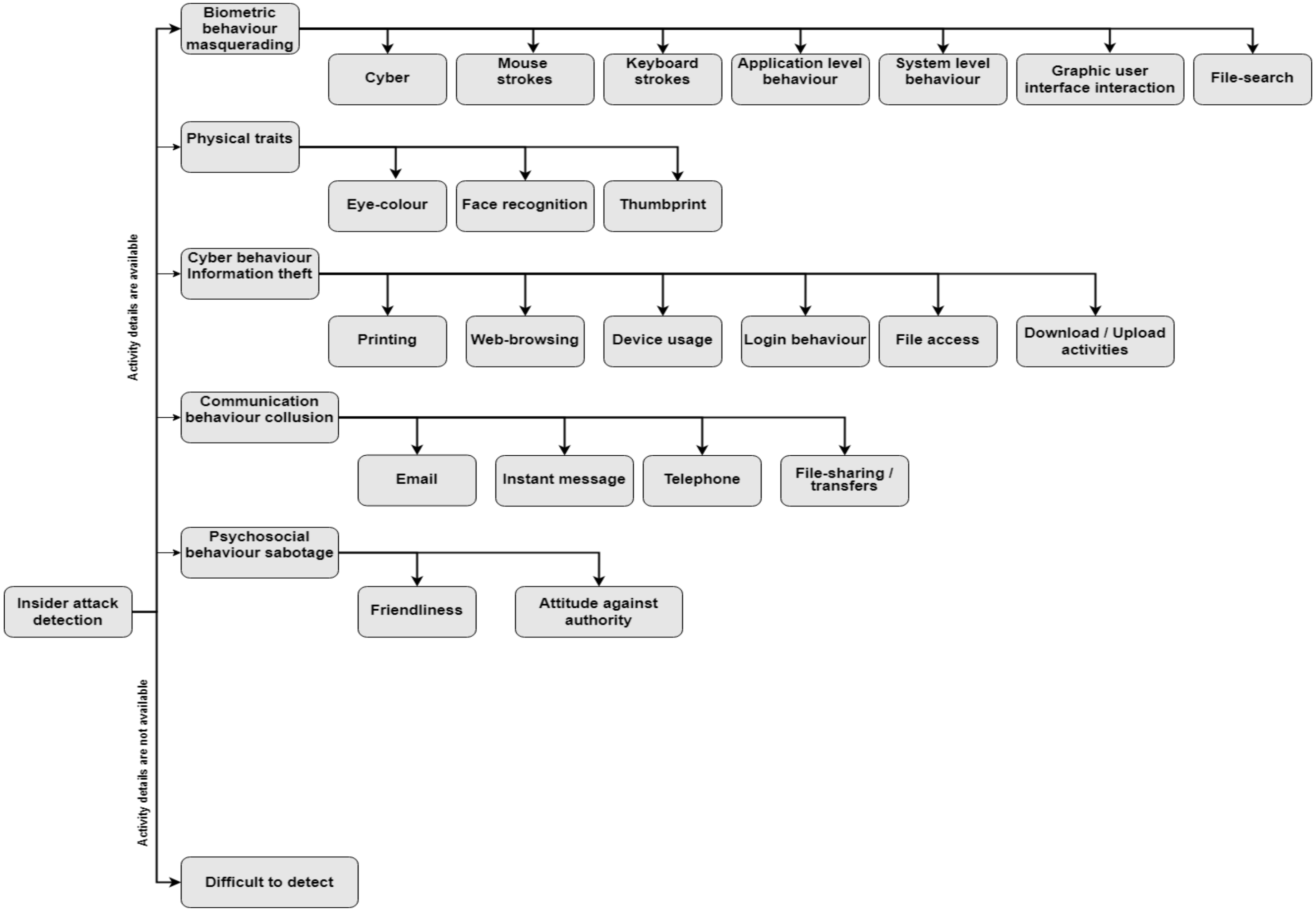

Insider attack detection
Activity details are available
Activity details are not available
Biometric behaviour masquerading
Cyber
Mouse strokes
Keyboard strokes
Application level behaviour
System level behaviour
Graphic user interface interaction
File-search
Physical traits
Eye-colour
Face recognition
Thumbprint
Cyber behaviour Information theft
Printing
Web-browsing
Device usage
Login behaviour
File access
Download / Upload activities
Communication behaviour collusion
Email
Instant message
Telephone
File-sharing / transfers
Psychosocial behaviour sabotage
Friendliness
Attitude against authority
Difficult to detect

Figure 2.1: Existing insider activity tracking techniques

Vidal and monge [43] proposed a machine learning model to detect masquerades. This model was based on windows users' and intruders' simulator logs as reference data sets. This proposed model used two sets of classifiers, in the first set, random forest, reducing error pruning the tree, and C.45 algorithms were applied, and in the second set of classifiers, bootstrap aggregation, Naïve Bayes and support vector machines were applied. Locality-based mimicry was able to detect legitimate users and attackers.

Hu et al. [44] proposed that ITH could be detected by understanding mouse Dynamic Behaviour and applying Deep Learning (DBDL). In this proposed work, all mouse movements and actions were mapped to images using tensor flow which further helped train the CNN network for creating a classification model. This way, a UB profile was created, which further helped to detect ITH. The performance of the proposed model was significantly better with a false acceptance rate of 2.94%, a false rejection rate of 2.28%, and an authentication time of 7.072 seconds (when m = 100) than the existing models in the literature.

Legg et al. [45] proposed a Conceptual Model (CM) and reasoning structure for detecting ITH. This CM consisted of three tiers: hypothesis, measurement and the real world. An analyst detected ITH by viewing through all three tiers. The reasoning component worked at each level and helped make a static profile of each user. Four elements worked under the real-world tier, enterprise, people, technology and information and physical. Operational level details were covered in the enterprise element, insider’s motivation and behaviour within the enterprise were covered in the people element, and digital activities in an organization were covered in the technology and information element. In the last, details of access control, location statics, security and physical destruction were covered in the physical element. On the working of these three tiers, CM detected ITH.

Agrafiotis et al. [46] proposed to detect ITH based on activity trees. These activity trees were based on attack trees, which monitored insider steps in normal working environments and malicious events. In these activity trees, the sequence of events was represented as nodes of a tree. Path in a tree followed by an insider in these activity trees helped to detect ITH, as at every moment parallel chain of nodes was available in case of a possible threat. The similarity index was calculated between these normal paths taken and in the case of the malicious path for IA.

Alsowail and al-shehari [47] proposed a multi-tiered framework to detect ITH. This framework consisted of pre-countermeasures, in-countermeasures, post-countermeasures, ITH prevention team and decision making. In pre-countermeasures, criminal background checks, personality traits tests, and security awareness training were conducted to make a user profile.

Countermeasures included technical behavioural and psychological profiling of a user. Post countermeasures included monitoring user behaviour and cyber activities in the notice period. The ITH prevention team evaluated and updated the profile in all these countermeasure steps, including the security analyst and psychologist. The final decision regarding the user according to its activities scores, whether a malicious or non-malicious user, was taken in decision making.

Sticha and axelrad [48] proposed a dynamic model for the detection of ITH. This model was based on two techniques system dynamics (SD) model and the bayesian belief network (BBN). In SD, monte carlo simulation and sensitivity analysis generate were used. These were used many times with specific distributions of input parameters.

Output generated from these techniques was used for estimating the conditional probability tables, which were applied to specify a BBN. The proposed model identified types of employees based on risk indicators: an average employee and a not satisfied malicious insider.

Tian et al. [49] had proposed a model for ITH detection based on the DL and dempster-shafer theory. This model worked on accidental and intentional ITH by filtering an organization's communication channels. Steps performed in the learning phase of this model were data cleansing, data sampling, data dimensionality, extraction and multi-feature learning.

The subsequent detection phase fed traffic data for pre-processing and feature extraction. The result was fed to the D-S fusion engine for analysing traffic identity into normal and malicious behaviour.

2.3 CLOUD ACCESS CONTROL MECHANISMS FOR INSIDERS

Various researchers have provided different approaches to CAC for security protection against CC employees, technicians and providers. Table 2.1 shows a comparison of many approaches that has worked on ITH.

Table 2.1: Insider authentication and activity tracking techniques available in the literature

Author	Work done	Trade-off
Wu et al. [50]	Encryption of user data was done before querying in the cloud.	The activity of insiders was not monitored, and authentication for Insider users is not available
Moon et al. [51]	The activity of insiders was monitored, and according to that, APL could be modified	Authentication for Insider users was not available.
Yaseen et al. [52-54]	The proposed architecture used multiple PEPs - a single PDP. In this dependency graph, and Knowledge base algorithms monitored the activity of insiders. APL could be modified.	Single PDP worked as a stressed member. It could handle a limited number of requests.
Dou et al. [55]	User-machine integrity dependency was available. The activity of insiders was monitored by which APL can be modified.	The working of the proposed architecture was machine-dependent only. The system was complex and challenging to implement for large sets of insiders.
Shaghaghi et al.[56]	The activity of insiders was monitored from network traffic, and according to that, APL could be modified	Authentication for insider users was not available.
Chattopadhyay et al. [57]	The activity of insiders was monitored by behavioural analysis through log files.	Authentication and change in APL for insider user was not available.
Baracaldo et al. [58]	Insiders' geo-social activities were monitored, and APL could be modified according to that.	Authentication for insider users was not available. The system was very complex in nature in practical implementation for a large set of insiders.
Meng et al. [59]	The medical smartphone could monitor the activity of insiders, which was used to make behavioural analyses.	Log file of activities of insiders was traceable, easily accessible and could be modified.
Babu et al.[60]	The activity of insiders was monitored by a Keystroke analyser which also helps in authentication and allows change in APL.	The system was very complex in practical implementation for a large set of insiders.

Eberz et al. [61]	The activity of insiders was monitored through eye movement Biometrics which also helped in the authentication.	The system was very complex in practical implementation for a large set of insiders.
Silva et al.[62]	Self-adaptive system-based MAPE-K framework, which consists of four stages: Monitor, Analyse, Plan and Execution, is used. This framework is integrated with the open stack system in detecting and preventing insider threats.	Response time in this system increases when the number of insiders increases.

Wu et al. [50] suggested encrypting user-provided data to prevent it from falling into the iniquitous user. Data was processed first by decrypted before applying the query to the data. And then, the data was encrypted again. An additional feature extraction algorithm was applied to the data before the encryption was proposed by the authors. In addition to other things, it made a query in the cloud more secure. Encryption was done by an index generator, a tool that took a user's data and compiled it into an index.

Moon et al. [51] proposed a primary method for detecting IA as a two-tier architecture. They had developed a very interesting In-Memory Database (IMDB), a proposed DB protection system. The work done by insiders was saved by taking notes in the change audit logs. A pre-processor processed data in the DB before being passed further for processes. This tool could detect an insider monitoring server's availability and predict any possible future attack. The cloud-based capability was also included in this system.

Yaseen et al. [52-54] proposed a model to detect and prevent IA. In its first work [52], the authors proposed a way to check for ITH because of insider knowledge. This approach used a knowledgebase algorithm that considered constraint dependency, hot cluster, safe cluster, and dependency matrix. A knowledge graph was generated, which warped users into limiting their access to sensitive data. The second work [53] proposed the threat prediction graph using probabilistic knowledge base analysis. Multiple PEP and single PDP were proposed in their third work [54] to detect ITH with the algorithms proposed. This system was suitable to work in situations where the number of PEP were insignificant.

Dou et al. [55] had compiled a set of standards and protocols for authenticating Hadoop systems with a trusted platform. This Protocol defined an authentication that removed user authentication's limitations and IA based on Kerberos. Authentication keys stored inside this protocol were not directly accessible. The protocol was bound to and had specific stipulations with specific systems. System computers were controlled, or certain software and hardware details were stored inside the system's internal registers for later use. This proposed protocol protected specific systems in achieving against additional and IA.

Shaghaghi et al. [56] proposed Gargoyle Software Defined Network (GSDN) architecture. GSDN commissioned the proposed work to detect and deter "must attacks" from ITH. It could also analyze traffic on passive network outlet paths and retrieve contextual information. First, three components were discussed: the network context analyser, risk management, and the advanced enforcement point. ITH had been detected. Based on this and other information, action plans could be created to prevent or mitigate attacks. The data was extracted by observing network traffic, and by doing so, the underlying goal was to monitor network traffic.

Chattopadhyay et al. [57] performed activity monitoring in time series. This study observed the usage of technical insider knowledge for a single day. The analysis was run at a single point in time, yet the samples were taken over a while as a way to detect ITH. Based on behavioural analysis, the statistics were taken to detect malicious or non-malicious insiders. The neural network was applied in the deep auto encoder classification technique, and the system improved the result.

Baracaldo et al. [58] suggested a Geo-Social Insider Threat Resilient access control framework (G-SIR). In this proposed work, insider movement activities were monitored. By monitoring employees' behaviour, specifically to activity in the workplace, the detection of enablers, inhibitors, or neutral employees was more comfortable. Inhibitors were people who potentially could harm. Users in the category of trust or extremely trusted could be placed in the category of enablers. Users in the category neutral could be placed in the neutral category. The framework was comprised of three modules: monitoring, context, inference, & access control. Role-based access control (RBAC) allowed for writing permissions and roles.

To prevent the Medical Smartphone Network (MSN) from being breached by an insider, a study by **Meng et al.** [59] analyzed potential behavioural patterns that could be established via the monitoring systems and setup to detect a malicious device. Nodes in the MSN were connected to the central server. The nodes sent the user's "statistics" to the central server. Each node in the network reported its "statistics" to the central server. A central server created each node (or piece) of the hierarchical relationship. As long as the malicious node behaved differently from others, its behaviour was detected by a difference in Euclidean distance. An evaluation was implemented in the real world using MSN and carried out by a practical healthcare institute.

Babu et al. [60] suggested a technique for preventing IA on CC by analyzing the insider's behaviour and associating risk-based access control systems. Behavioural analysis was done through the use of keystroke dynamics. Risk analyses could be conducted in an "offline" manner with the available resources. Every object had a numerical value of perceived risk. The DB called resource stores information about risks and safety. It utilized a support vector machine. Whenever a malicious user was detected in the system, the system effectively closes all privileges, segregating it from the rest of the internet.

Eberz et al. [61] discussed methods to detect an ITH using concurrent user eye movement. In this work, researchers had developed 20 features that a user was matched. These 20 features existed continuously, so the algorithm could identify the user. Video-based gaze tracking was applied to keep track of eye movement. Experiments were conducted controlled under a lab condition; 30 persons from the general public were used. People were asked to take part in various activities on the screen. Most important among them was the study of eye movement and other aspects. The open and closed set classes were applied to the retrieved data.

Silva et al. [62] have carried out research work on preventing insider threats. They have thoroughly studied keystone, in open stack used for identity management components to show various threat cases where the existing framework cannot detect and prevent insider threat. Authors have suggested using a self-adaptive system-based MAPE-K framework consisting of four stages: Monitor, Analyse, Plan and Execution.MAPE-K framework was integrated with the open stack system to enhance its capability in detecting and preventing insider threats. MAPE-K framework interacts with OpenStack via effects and probes. The controller in MAPE-K regularly takes feedback from OpenStack, monitors insider activity and takes action accordingly. As the number of users in this system, response time also increases, which is also its limitation.

2.4 CLOUD ACCESS CONTROL MECHANISMS FOR OUTSIDERS

Security and authentication ensure that only legitimate customers can have access to data and that hackers cannot attack their data. The purpose of authentication is to authenticate who has deemed the requestor (application /user). Over the past years, several researchers have studied the concept of authentication techniques available for outsiders, as shown in Table 2.2.

Table 2.2: Outsider authentication techniques available in the literature

Author	Work done	Tradeoff
Tsai et al.[63]	Three-factor authentication was proposed using elliptic curve cryptography. Single sign and mutual authentication were available.	Not suitable for resource constraint IOTs and multi-owners. Mutual authentication was not available.
Kalra et al.[64]	Two-factor authentication was proposed using elliptic curve cryptography, suitable for resource constraint IOTs. Mutual authentication was available.	Single sign-on was not available. Not suitable for multi owners authentication
Amin et al. [65]	Multi-factor authentication was proposed using bio hashing. Mutual authentication was available.	Single sign-on was not available. Not suitable for resource constraint IOTs and multi-owners.
Yang et al. [66]	Two-factor authentication was proposed using delffie - hellman. Single sign-on was available.	Not suitable for resource constraint IOTs and multi-owners. Mutual authentication was not available.
Kumari et al. [67]	Mutual authentication was available. Multi-factor authentication was proposed using Elliptic curve cryptography. It was suitable for resource constraint IOTs.	Single sign-on was not available. Not suitable for multi-owners.
Shajina and varalakshmi [68]	Two-factor authentication was proposed using triple D.E.S, Single sign-on, Mutual authentication and multi- owners authentication.	Not suitable for resource constraint IOTs.
Anakath et al. [69]	Multi-factor authentication was proposed using simple -homomorphic encryption.	Single sign-on was not available. Not suitable for resource constraint IOTs and multi-owners. Mutual authentication was not available.

Chaudhary et al. [70]	Three-factor authentication was proposed using elliptic curve cryptography. Single sign-on and mutual authentication were available.	Not suitable for resource constraint IOTs and multi-owners.
Kumar et al. [71]	Biometric-based authentication was proposed using elliptic curve cryptography. Mutual authentication was available.	Single sign-on was not available. Not suitable for resource constraint IOTs and multi-owners.
Chatterjee et al. [72]	Biometric-based authentication was proposed using clustering. Mutual authentication was available.	Single sign-on was not available. Not suitable for resource constraint IOTs and multi-owners.
Deebak et al.[73]	Single user sign in and authentication was proposed for IoT based medical devies to work in cloud of medical things.	Details of Network delay,throughput rate, routing overhead and energy consumption are not available.

Tsai et al. [63] proposed a system to authenticate mobile users. This method allowed individuals to access various cloud services using a singular private key. A smart card provider created a user and the service provider's public and private keys to authenticate each other. The smart card gave users access to access services from SP. This way, involved users remained anonymous, traceable to no one, and unable to be faked, signed with each other, verified, or exchanged.

Kalra et al. [64] proposed a mutual authentication scheme for Internet of Things (IoT) devices and cloud servers. This scheme used Hypertext Transfer Protocol (HTTP) cookies to implement the secure Elliptic Curve Cryptography (ECC). IoT devices used cloud support to improve their processing capacity. For embedded devices to authenticate themselves with the server, they should function as HTTP clients. The Transmission Control Protocol / Internet Protocol (TCP / IP) protocol stack was often used to customize embedded systems. Within this protocol, it had different phases that included the setting-up (the first phase), a system that could recognize the device (the second phase), and a system that could authenticate the device (the third phase). This protection scheme provided resistance against various forms of attack, such as brute force attacks, eavesdropping, man-in-the-middle attack, offline dictionary attack, cookie theft attack, and replay attacks, and provided forward secrecy, anonymity, confidentiality, and mutual authentication.

Amin et al. [65] proposed an authentication process for a multi-medical server system employing user name, password and biometric features (fingerprint & smart card). In the first part of the system, a user chose the desired identity, password, and facial recognition template (such as a fingerprint) and sent them via a secure channel for registering. After a user had submitted their details, a bio-hashed version of these details was stored on the user's smartcard and the medical registration server. Once a user had been authenticated on the medical server, he could fetch data from the desired medical server according to his evolving needs. This technique prevented the session key disclosure attack, the user impersonation attack, the replay attack, the initial wrong password identification, and the mutual authentication, which resists the offline password guessing attack.

A protocol to allow access to multimedia data on the multimedia cloud was proposed by **Yang et al.** [66]. This protocol used open ID two-factor authentication, which required smart cards and user login information. This scheme used various cloud models to authenticate smart cards and users. User APLs were formulated according to the RBAC paradigm. The work included analysing the Security, functionality, and efficiency of the application.

Kumari [67] proposed the authentication mechanism for a growing number of IoT devices and cloud servers. This system used a multi-factor authentication mechanism, including login details, cookies and device details, and tamper-resistant devices. This authentication protocol provided a suitable authentication process for resource constraint IOTs where the need for mutual authentication arose. The elliptical curve's cryptography helped prevent various types of attacks, such as the absence of device anonymity, IA, guessing of offline passwords, and computing of no session key.

Shajina and varalakshmi [68] proposed a multi-owner authentication protocol, which worked on different owners, group managers, and service managers in a cloud for authentication and improvement of the security requirements of single sign-on. The proposed mechanism allowed the primary owner to create a group and allowed them to build more members. Owners were given a valid token after the certification body had verified it with all required parameters. The token consisted of all the information of the user credentials, the token expiration time, the services it was used for, etc. The session tokens from the session manager enabled the access that the service required.

A trust model was proposed by **Anakath et al.** [69] for authentication in a device. The device's identity was identified, and a protocol was selected to verify the device's identity. Knowledge, possession, and inherence factors come into play to confirm the act itself. This protocol applied possession factors, one-time passwords, and passcodes that no one else knew. As part of a user account on big data Multi-Factor Cloud Authentication (MACA), a profile was established with user details and parameters in an encrypted format.

Chaudhary et al. [70] presented an improved authentication scheme. The paper allowed the authentication server to block a forged request/reply at the time of authentication. An approach that used a single private key to authenticate mobile users allowed them to access all cloud services from multiple CSP. This research was an enhancement work of **Tsai et al.** [62]. The proposed research was more promising, mature, and verified in proverif.

Kumar et al. [71] suggested using biometric information to authenticate users within the cloud. In a nutshell, the way to the biometric password was a facial feature. Biometric DB stored the facial features of cloud users, which were encrypted. Facial image features were extracted from preprocessed facial images. Facial recognition helped in identifying users. Hundreds of facial features were calculated and matched with the stored similarity scores of facial features that would be stored and retrieved similar to the currently measured facial features.

Chatterjee et al. [72] suggested a re-authentication system biometrics-based. This system was secure, and its security level was enhanced by using keystroke dynamics (detecting the tendency of keystrokes). In this scheme, they were asked to enter login information to verify that they were legitimate customers like anyone else. When the user typed his credential, the keystroke dynamics were logged into the DB. These specifics helped identify and verify the details obtained by a k-means clustering algorithm. Feature sets were tested with a heterogeneous, homogeneous, and aggregate class of methods.

Deebak et al. [73] have highlighted the role played by the cloud of medical things in developing smart healthcare systems. In this research work, Authors highlighted that numerous authentication mechanism are still persuable to security threats. Researchers in this work proposed sensor/ sensor-tag based smart healthcare environment which uses single user sign in to mitigate various threats. Single-user sign-in was based on the Chebyshov chaotic map in COMT. Formal verification was done with BAN Logic and found suitable for working in cloud-based IoT devices in healthcare environments.

2.5 AUTHORISATION TECHNIQUES FOR USERS

To allow a user to access data is decided under the authentication policy. A policy about authorisation and authentication plays a vast, significant role in the modern world. Table 2.3 shows APL frameworks used in distributed environments and their applications.

Table 2.3: APLs framework used in the distributed environment available in the literature.

Author	Work done	Trade-off
Abomhara et al. [74]	Authors had proposed a work-based access control model for cooperative healthcare environments	The proposed model performance validity was not evaluated regarding resource consumption, e.g., time and computational capability.
Alam et al. [75]	Authors had proposed garbled role-based access control for cloud	Practical implementation of the FHE scheme on GRBAC and duty conflicts was not done
Habiba et al. [76]	Authors had proposed icancloud simulation platform for cloud	A comprehensive audit trail for filtering and reasoning over the audit trail information and manifest potential security threat was not included.
Sun-Moon Jo [77]	Authors had proposed an access control model for a dynamic XML data environment in mobile computing	A safe access policy of the system could be improved. The proposed system should handle extensive capacity XML data.
Chen et al. [78]	Authors had proposed an AC model using RBAC for community medical internet of things	The proposed model did not consider the multiple identities of medical staff.
Shin et al. [79]	The proposed Proposed anonymous AnA architecture was based on traceable signatures. It included four protocols and 12 inner algorithms.	The proposed architecture was complex to implement in real-world scenarios.
Gabillon et al. [80]	Authors had proposed an Authorisation model using ABAC for pub – sub network For IoT	The proposed model was not suitable for bridged brokers and Low Power Wide Area Network (LPWAN) hosting the sensors.
Rathore et al. [81]	Authors had proposed an Authorisation model using answer set programming for online social network	The proposed model was complex in nature in terms of implementation for preventing user privacy.

Abomhara et al. [74] proposed a work-based access control (WBAC) model wherein team member roles were classified based on the belbin team role theory. In work-based assessment models, teams were segregated based on the type of work they produced and what type of work they contributed in collaborative work. A suggested work plan had been planned for potential collaboration in healthcare environments. Essential in a situation where a patient worked with multiple doctors from various departments and multiple hospitals that worked together to save the patient's life. Doctors might not divulge sensitive information to unauthorised persons with the patient's healthcare records, as this data was maintained confidential. The authors had first formalized the system, consisting of essential elements and relations, defining various AC functions, and many authorisation constraints. This reduced the complexity of updating permission reviews over other models such as RBAC and Attribute-Based Access Control (ABAC).

Alam et al. [75] proposed that users could access resources dynamically based on their role. These were essential computing techniques used in the garbled circuit and fully homomorphic encryption. This garbled mode of computation was applied in Role-Based Access Control (RBAC). Markings were associated with the specific job roles of the users on the system. All the relevant information about the roles was stored in the RBAC server. In a scenario where the GRBAC was hacked, it would be impossible to know roles. It provided firm security in CC.

Habiba et al. [76] proposed a dynamic access control system in the cloud. This proposed system consists of mainly four models (i) Data Access Right Model (DARM), (ii) Policy Model (PM), (iii) Access Control Management Model (ACMM), and (iv) Authorisation model. The DARM consisted of access rights trees constructed from a collection of access rights trees. It consisted of a hierarchical relationship between a set of access rights, such that one access right pointed to another.

A PM consisted of obligations, conditions, primary rules, deadlines, and user preferences. Policies must represent subjects' rights, resources, rules, and preferences. Because every policy must be expressed in the format of 8 tuples (S, A, Rs, R, C, O, D, F), the ACMM should have many sub-models to control access in the company. The data Authorisation model was made up of three stages: the pre-Authorisation stage, Authorisation stage, and Authorisation's post-stage.

Sun-moon jo [77] devised a safe access practice for XML data. It worked on a resource-efficient secure access policy that would control access as information was controlled according to access privileges, role in the environment, permission to view the data, and how to pass the data (e.g., to ease the use of data across data boundaries). The policy allowed limited access to parts of the target document, allowing every policy to work on the whole document. In this paper, the target document focused only on the areas easily accessible as central units in the target document, and minimal access policies are applied to those central units.

Chen et al. [78] declared that the Community Medical Internet of Things (CMIoT) would play a vital role in the medical data. The protection of patient medical data was controlled by various safeguards, including transmission, storage, and access control. The security of the transmission was achieved via asymmetric encryption. The storage security was achieved by symmetric encryption, and the security of accessing was achieved by dynamic Authorisation based on role.

In CMIoT, data was collected from various IoT devices, fragmented, encrypted, and sent to the cloud for storage. The user could access specified cloud data based on the level of his role. Cloud-based data management allowed data to be retrieved only from the community connected to medical IoT from the third-party cloud of the community.

The paper on AAnA was proposed by **Shin et al.** [79]; it used short traceable signatures. Three authorities were working at the same time. The first authority was the group manager, and the second authority was the Authorisation manager. Both of the following features were accomplished using the two distinct managers for group membership and Authorisation. The group manager's role was to verify who belonged to the group using a short traceable short signature.

Authorizing manager was an application which provided privileges to users based on their real identities. This Authorisation list of all users and their privileges was forwarded to the service provider. The service provider asked for a signature from the user whenever it detected illegal activity to help make the account secure. This signature was passed to the group manager and the Authorisation manager to authorize further actions.

Gabillon et al. [80] proposed a very expressive ABAC security model. This model is communicated via the Message Queuing Telemetry Transport (MQTT) platform. The MQTT protocol allowed devices to send data to other devices to communicate and coordinate. For each topic, a message about that topic was published. The messages were published under the topic for which the messages were being published. It was assumed that only one MQTT broker was chosen for this discussion. It assumed working on TLS / SSL at the Transport layer between all nodes of the IoT network to make IoT network nodes encrypted. It used first-order logic with equality to describe the proposed model. This AC enforcement model consisted of PEP, PDP, Policy Information Point (PIP), and Policy Administration Point (PAP). PEP intercepted all of the MQTT messages and forwarded them to PDP; PDP got help from PIP in deciding the access and saved the results in PIP. A logical security policy for this model was defined in the Resource Description Framework (RDF).

People's online privacy had been compromised on social networks when information was shared multiple times, as the information had not been given consent. The paper of **Rathore et al.** [81] proposed that a social media site be explicitly designed and protected by an AC style. This control model worked under which conditions multiple or single participants were subscribed to the social network (web sites). In this model, trust was specified as calculated among each resource owner. AC policy varied between family members and friends, with the trust level more prevalent in family members. The proposed model was represented by answer set programming, which relied on logical propositions.

2.6 AUTHORISATION ARCHITECTURES FOR USERS

Policy development plays an essential role in every PDP architecture because they make every final decision regarding the policies. An architecture built using the PEP-PDP is found to be an essential factor in the APLs. This requirement is commonly called managing user Authorisation. It has always been a fundamental part of user management.

The PEP-PDP architecture application is found in almost every field where the final policy decision has already been taken. A single PEP-PDP architecture with its applications is discussed in section 2.6.1, and multiple PEP - single PDP with its application is discussed in Section 2.6.2.

2.6.1 SINGLE PEP-PDP ARCHITECTURE

A single PEP-PDP architecture includes one PEP and one PDP server. However, they become inefficient when they have to handle a large number of requests. There are many places where their applications are found. Some of them are shown in Table 2.4.

Table 2.4: Single PEP-PDP architecture applications available in the literature.

Applications/ Authors	**Decentralized multi-agent systems**	**Electronic patient health record**	**Access control in cyber-physical space**	**Increase the agility of the electricity grid.**	**Cloud-based healthcare recommender service**	**Managing android permissions**	**Fall detection system for hospitals**
Nyrkov et al. [82]	X						
Son et al. [83]		X					
Cao et al. [84]			X				
Ryan et al. [85]				X			
Elmisery et al. [86]					X		
Oglaza et al. [87]						X	
Krempel et al.[88]							X

Nyrkov et al. [82] proposed encrypting protocols in a decentralized multi-agent system. An agent was a particular entity independent in collecting and processing data. The final product was sent on the channels. Multi-agent systems were systems of several agents working on a common goal. The advantage of using a multi-agent system was that it could be used even after some instances were gone. A Multi-Agent System (MAS) was configured to work in a decentralized manner. The system proposed required trusted centres to register with the authentication servers by sharing a generated (or randomly generated) secret key. After that, all of the AnA procedures between them took place. The procedures followed between Attribute authority and agent corresponded with each other—access Policies in XACML.PDP allowed declarative APLs and contingent access policies to implement meta policies and decide whether access was granted or denied. In the PAP phase, new policies were created; in the PIP phase, the required data was given to the PDP; and in the PDP phase, the policy was applied by the PEP. The MAS was improving, but it lagged behind the synchronization speed of the multiple agents.

Son et al. [83] attempted to offer solutions for the privacy control of patient records in the various locations where they were made available. The patient log could be viewed by doctors, nurses, insurance companies, pharmacies, and relatives. Authors had suggested using electronic patient health records in the public cloud to better control access to the records in a more secure and controlled mode. The two main challenges were: denying access to the patient's private data to attackers to protect it against improper uses by leakers and protecting shared data against loss and misuse that might harm the patients. The author proposed a privacy protection model in which the Authorisation of cloud access was based on the healthcare system's AC model. This new system required accessing the data from PEP first and then transmitting it to access so that PEP got information and the access requester got to know more about the subject, action, and environment.

This request was transferred to the PDP, who decided on an access request through the appropriate policy. While creating the access policies from the repository, the PAP platform acted to store them. PAP also made sure to make available policies about privacy and use. The limitations of this proposal would be to deal with an emergency.

Cao et al. [84] proposed a cyber-physical AC framework for cyber-physical space. The TAAC (Topology-aware access control) model was proposed to allow better access control. (This was an extension of) RBAC model was like the TAAC model. The TAAC model was a combination of physical and cyber access controls. This type of security was adaptive, adjusting or limiting the user's privileges according to their behaviours. This paper also suggested secure policy enforcement, which should help to mitigate ITH. The risk value was calculated from the data on the account holder's previous activities and the recent access request. Through such facilities, undesired users could be restricted from accessing the system.

The AC and framework enforcement modules functioned correctly in the cyber-physical space. In the first module, PEP received a coverage request, which was sent to the PDP for a response. PDP stored data according to the policy contained in PAP. Topology attributes were stored in the PIP, and risk attributes were stored in the risk module, both of which helped the PDP make decisions. In administration, trust policies were applied according to the trust repository for each user. Policies were managed through policy constraints in the policy constraint management module. This proposal did not cope with the fact that multiple cyber access spaces were available in a smart city.

Ryan et al. [85] suggested a new type of control to increase grid agility. As the authors suggested, better power and energy network management could be achieved by using policy-based network management (PBNM). For the PBNM system to indeed be able to perform well, authors had proposed using a text mining technique at the LV level to derive connection parameters. It also performed a Voltage Adaptation Ritual (VAR) on all the DER networks to arrive at the settings that would do the best job for them. In another paper, the authors proposed using PBNM Voltage Control Validation using PEP-PDP architecture. Policies that must be used across multiple systems were stored in a policy repository, one of which was fetched by PDP. The Power factor checked the applied policy against the calculated value. When PDP determined capacity limits in the violated policy, it raised the allowed VVC.

Elmisery et al. [86] presented a cloud-based healthcare recommendation scheme. The healthcare service discussed and processed data obtained from the internet. This service kept data safe by two times encrypting and obscuring the data before storage in the cloud. A personal gateway would be placed at the patient's side to provide the first concealment level. Data were collected to a fog node and then transferred. This security sensor applied attribute-based encryption to the encrypted portion of the health profile before relaying. Every group had multiple users and multiple profiles. The security authority centre generated the certificates for both fog nodes and gateways.

The privacy of patients' health data had been protected, which had increased trust. Patients' privacy policies were built according to the IOHT records. A PIP unit was a privacy preference unit. PEP analysed privacy policies to determine actions on in-out behalf. Policy agent acted as PDP that controls the flow of user-health data to external medical information assets. An agent performed a first-level concealment procedure, and a global concealment agent completed a second-level concealment procedure. The trust agent computed confidence.

Oglaza et al. [87] addressed the concept of a user's confidentiality and information in an android platform based on a smartphone. Android applications, on average, demanded 11.4 access permission, of which 5.12 directly impacted the device's privacy. This amount grew as the number of android apps resulted in more privacy breaches. Fix this issue, as researchers had suggested a Decision Support System (DSS) for writing elevated policies, where traditional non-technical users could write policies. This DSS was based upon the recommendations in their entirety. In this system, the characteristics of each user were stored locally. Depending on the local "cloud" storage characteristics, a new object was instantly suggested. The system was helpful if the user's behaviour did not change. It took time for the system to learn. The proposed framework "Kapur" would comprise DSS, Policies, PEP, and PDP.

An app received a request to access one of their private details. PEP received this request and decided whether or not the request qualifies for approval and whether the user had given consent. PEP transformed and transferred the request to PDP in the XACML V3 layout. Which supported or refused controls for available policy choices. If PDP rejected the request, most of the system's information about user interests was absent.

Even further, this requirement sent to DSS asks users to allow or deny the requirement. If permitted by the user, the new XACML V3 feature was generated and modified in the DB policy. The DSS became sufficiently mature when the requirement score exceeded a predetermined threshold value. Among other uses, the proposed framework handled user privacy more relaxed and better.

Krempel et al. [88] proposed a system for fall detection in patients. In their work, 12 essential requirements were analysed to fulfil the stakeholders' demands, i.e., hospital operators, medical personnel, patients, and legal stakeholders, thereby supporting the system to compete with the existing systems. These 12 requirements under the nursing process consisted of (a) detecting collapses, (b) intimating Nearest Nurses, (c) preventing misuse of details, and (d) confirming urgent situations to provide two-way interaction between patients and concerned nursing staff members.

This system could work in three modes; default mode, assessment mode, and investigation mode. In the first mode, all the cameras continue to work. In this mode, fall detection algorithms also process video data for fall detection. In the second mode, no person may interfere with the operation; in the second mode, the algorithm tracks any fall, and the system enters into a second mode, an assessment mode.

In this second mode, as quickly as the algorithm simultaneously identified the fall in the camera's video, a message was broadcasted to nearby nursing staff indicating an alarm was triggered. When the alarm did not get an acknowledgement, the notification was broadened. If any nursing staff unlocked the alarm, all other alarm messages were forced to cancel.

In the last mode of the emergency response, the investigation started after the nurse performs an anonymous video assessment of the patient. The anonymised video was played out to the nurse, who then conforms to the inside of the emergency. In case an emergency happens, an on-screen camera image was optimised for the nurse to view it to help in decisions regarding medical equipment. Communication between patients and nurses could be done in both directions, thus better handling the situation. The system was controlled and managed by PEP, PIP, and PDP. PEP monitors and logs all events and then forwarded them to PDP to ensure they meet the safety requirements of PIP. This system was designed to detect a fall. It is the best possible solution to do so.

2.6.2 MULTIPLE PEP - SINGLE PDP ARCHITECTURE

Yaseen et al. [54] highlighted the adverse effects of ITH. The ITH was the most vulnerable threat than the outsider threat; In contrast, insiders knew various procedures and policies of their system; outsiders could steal from the windows they could access. Cloud usage had been increasing dramatically, which required the insider group to manage it.

A PDP with some PEP support, one that the PEP had given governing judgment. With PEP, the decisions taken copies keep on getting stored inside of caching, ultimately improving the functionality of making decisions. A query received at PEP used a similar query already in the PEP side caching. The decision to that request was enforced on the received request.

This paper provided evidence that insiders had an advantage in attacking data theft of a cloud-based relational DB. Insiders had the expertise to be able to have a better idea of what the cloud data would look like based on various data dependencies. Researchers had also demonstrated that cloud relational DB and cloud distributed systems fall short when detecting attack attempts based on various inferences. Authors had also shown that keeping track of what is happening in the cloud is not easy.

In this proposed architecture, multiple PEP with side caching was interconnected by a single PDP. An insider request to access data was received by PEP. If PEP had a similar request, then PEP could request a decision copied from the cache, or PEP could ask from one of its neighbours at the next level. Requests to PDP were sent in case no previous decisions were found in PEP or neighbouring PEP.

On receiving the request at PDP, the risk of granting access was assessed in Internal Threat Detection Unit (ITDU) according to the algorithms [52][53]. After checking the risk level of the request, ITDU provided the entry or denied the request. Results showed the performance of this multiple PEP with a single PDP architecture was better in handling requests than a single PEP-PDP architecture.

2.6.3 PEP - PDP ARCHITECTURE-RELATED ISSUES

In the past, researchers have used different applications of the PEP-PDP architecture. Issues related to them are discussed below.

Baldini et al. [89] explored the risks of unauthorised access to personal data made available via IoT devices. If users did not want their data shared, the authors had proposed using an ethical data protection protocol design. Regardless of their ethical choices, the ethical design models they proposed also dealt with multiple challenges applicable to mitigating privacy risks.

This paper also suggested that the model-based security toolkit was the best architecture toolkit. This kit allowed for the implementation of a set of predefined policies on user's profiles with a set of defined conditions Event – Condition - Action (ECA.) The structure used regulation and set up a policy to manage the environment. This architecture was centred on the PEP and the PDP. PEP sent automatic updates to the PDP. On the other hand, PDP responded to the behaviour executor with a reactive/preventive compliance message. A proposed model to maintain users' privacy had been suggested.

Ghazi et al. [90] suggested that DB security should be taken care of as a core service for document-oriented NOSQL DB. For coping with numerous security-related issues applicable in NoSQL DB over the cloud, the proposed system covered various authentication issues, authorisation, and encryption of the DB objects. Within this scheme, the authentication service consisted of three principal parts. Namely, a safe authenticate service, an identity management service, and a certification authority service.

The following section combined fine-grained authentication services such as PEP, PDP, and PAP. In this fine-grained access layer, policies were developed by PAP and archived in the PAP cloud DB. PEP received and responded to access requests from users before sending them on to the PDP. Based on the currently stored policies, the PDP decided whether to provide access or not. Confidentiality comprises service key distribution and encryption.

The symmetric keys were also obtainable in key distribution sets. At each user, separate keys were created and maintained independently. Cryptographic service data was stored on the cloud and encrypted and decrypted using an advanced encryption algorithm after the

KD. Service. DB-SECaaS ensures the safety of NoSQL data-store applications through advanced patterns.

Context-aware security systems allowed the middleware to manage policies being enacted. In the paper of **Mehak et al.** [91] concentrated on the security issue associated with CC. The authors highlighted various difficulties and dire circumstances, such as data leakage, loss, account hijacking, and insecure API usage. This paper also had a proposed framework known as PaaSword, which helped maintain the persistence layer.

The framework could work in various ways to meet the CSS requirements and took on many challenges. Due to various challenges in the distributed cloud architecture, existing AC Authorisations were unsuitable for the cloud environment. In some cloud environment cases, once AC Authorisations is applied on the user, access request during and after the resource usage is not evaluated.

In some cases, continuous resource usage monitoring and a dynamic update of attribute values are evaluated. These types of diverse AC Authorisations are not fit for every Cloud enviornment. Authors have suggested an extensible AC framework for the cloud environment in order to handle them.

The framework that had been proposed was applied to a variety of cloud environments in a more generalized way. Some EACF features used a standard access-control policy specification format, common access layer for cloud applications, a generic framework for cloud-hosted applications, framework extensibility, development, and support for third-party plugins.

The architecture of EACF was composed of PEP as a Service (PEPaaS), PDP as a Service (PDPaaS), PAPas a Service (PAPaaS), Attribute Authority (AA), policy repository and attribute repository.

The primary function of the PAPaaS was to manage and create AC policies. In the PEPaaS architecture, PEPaaS received the user's request and converted it into an XACML decision query that then gets forwarded to the PDPaaS. The query was sent to PAPaaS, which provided the requested policy and considered whether or not there is a need for it in a given situation.

Through this proposed framework, various cloud services would be able to provide authorised access to cloud services. This framework fulfiled different properties such as extensibility, generality, consumer-driven, Authorisation function, and standard policy language format.

2.7 REQUIREMENT OF MESSAGE SECURITY SYSTEM / DATA PROTECTION SYSTEM IN AUTHORISATION ARCHITECTURE

Many researchers have focused their research on the importance of data protection concerning the Indian scenario, as shown in Table 2.5. Their work mainly related to the role of existing sections in Indian IT acts, cybercrimes, cyber defence mechanisms, the impact of the European union's general data protection regulation on Indian data privacy laws, and steps towards providing concrete solutions with effectiveness in the form of laws.

Table 2.5: Importance of data protection

	Importance of data protection raised by various researchers
Anja Kovacs[92]	The author highlighted the issue of user data protection, cyber security, and data protection regulations in India. It also tried to provide a clear picture of actors involved in protecting personal data.
Akshaya S[93]	The author systematically analysed the data protection laws and raised their requirement in India. The author also discussed the judgments of the Supreme Court of India regarding personal data protection.
Bhadade et al.[94]	The authors discussed the impact of the European union's general data protection regulation on Indian data privacy laws and highlighted its importance in the presence of various attacks such as the Facebook privacy scandal.
Gada and Aghav[95]	Authors had done a detailed study on the law reforms needed in the data protection laws of India by taking very vital issues such as the processing of personal data, responsibilities of company directors, the role of data protection officer, steps for preventing fake news, the powers of investigating police officers, and also raised the importance of their work
Ashit Kumar Srivastwa [96]	The authors discussed the impact of existing data protection and its concerns laws in India in the current digital scenario. The authors stressed providing a concrete solution with effectiveness in the form of laws

Anja kovacs [92] focused their work on the cyber security issue of personal data protection. The author, in his work, had also discussed the draft of the personal data protection bill, which described steps for personal processing data collected by Indian companies, states, and bodies. Various sections of the Indian IT (amendment) act was also discussed regarding protecting personal data.

The author had mainly focused his work on describing a draft of a personal data protection bill, which focused on explaining various categories of personal data, the Role of the data controller, and the data processor/data operator. It also defined data protection in the digital sphere, obligations of the controller, processors/operators, steps involved in the processing of personal data, security requirements for collecting and processing personal data, jurisdiction area regarding data transfer agreements, administration structure and powers of actors involved in enforcing data protection law. The author emphasized the need for personal data protection law in India.

Akshaya S [93] focused their work on the issue of expansion of existing personal data protection laws and their requirement in India. The authors discussed the personal data protection bill, 2018. It also focused on the judgment of the Supreme Court of India, which said that privacy is the fundamental right under the purview of article 21 of the Indian constitution as a part of the right to “life” and “personal liberty”. Information privacy was also part of privacy.

The author also discussed vital principles for data protection in the Indian context. According to these principles, personal data protection law must be focused on the dynamics of technological change. It should cover private entities and the government. Consent must be achieved from citizens. Personal data must go for minimal processing. The unlawful data processing means penalties, and the data controller was accountable. The author hopes the data protection bill could achieve security at its optimum level.

Bhadade et al. [94] laid the importance of personal data protection. They had given the examples of the Facebook privacy scandal in the united states of America's presidential elections in 2016, in which 50 million users' data was studied without their consent, knowing their political interests. This type of scandal was also evident in elections held in India in 2019 by a major political party. In 2018, the European Union general data protection and regulation implemented a law in this regard. This law provided a guideline for other countries and helped them frame personal data protection laws. It also helped India in framing the personal data protection bill in 2018.

The authors highlighted various components of the personal data protection bill, such as its applicability, processing of data under various sections, principal data rights and its sections, transferability and accountability issues under various sections, transfer of personal data outside India, and applicable penalties. The author also discussed the details of the data protection authority in India given in the bill.

Gada and Aghav [95] highlighted many issues in the draft of the personal data protection bill, 2018. In their work authors tried to gave a clear definition of personal data, the meaning along with the steps involved in the processing of personal data, the responsibilities of company directors, the role of the data protection officer, steps for preventing fake news, the powers of investigating police officers which helped in the investigation of cases, and also tried to explain the different classification of intermediaries. The authors tried to explain various sections of this bill for better understanding. Their work had taken up key issues related to personal data protection and helped a common man understand them. The authors had taken up other issues like the powers of intermediaries in data collection.

Ashit Kumar Srivastwa [96] highlighted the steps for protecting personal data, the existing IT act 2000, the information technology rules 2011, and the Supreme Court’s judgment. With this judgment government of India has appointed the B Krishna committee to look into the data protection regime in India. The committee, in this report, suggested creating a concrete personal data protection bill in 2018. The committee had observed in this report that the existing Indian IT act could not match the dynamics of the digital world, where it is much more important to protect personal data. The proposed personal data protection bill as a draft by the committee supplemented the efforts of existing Indian IT acts and regulations.

2.8 RESEARCH SOLUTIONS PROVIDED BY BLOCKCHAIN

Blockchain has helped in numerous applications in achieving the desired security level.
Table 2.6 shows various solutions proposed by blockchain in various research areas.

Table 2.6: Blockchain solutions in various research areas available in the literature.

	Research area	**Solution proposed**
Chen et al. [97]	Fraud in education	Student education detail in blockchain
Li et al. [98]	Bottleneck and compromising issue in centralized management server in VANET	Decentralized VANET with all data in the blockchain

Chung et al. [99]	Difficulty in maintaining customized product process management in cognitive manufacturing	Blockchain was used to maintain data of process management in cognitive manufacturing
Sun et al. [100]	Trust-building issues in sharing-based smart cities	Blockchain could be used for trust-building in sharing-based smart cities
B.Vinod [101]	Difficulty in maintaining details of interline charges, agent bonus on airline bookings and tracking of property booking when multiple sites were used for bookings	Blockchain could help maintain a single record for interline charges, agent bonuses and property bookings.
Han et al. [102]	Cyber attacks issue in patient health data in a centralized system.	Patient health data was stored in a hybrid blockchain, private blockchain at the local hospital, and consortium blockchain at the upper level. Hybrid blockchain makes things difficult for an attacker.
Ryu et al. [103]	Digital forensics involved a very lengthy and challenging procedure in IoT for sent messages	IoT digital forensics was made simpler with the use of blockchain

Chen et al. [97] proposed the use of blockchain in the education sector. Blockchain could be used as learning as earning; digital currency could be rewarded for intelligent contracts between students and teachers. All academic details of a student were stored in the blockchain, including assignments, exam results and degree details to prevent fraud in education, which could be accessed by student ID. The same could be applied to teachers and schools where teachers were rewarded with digital currency based on their performance and teaching activities.

Li et al. [98] proposed using blockchain in Vehicle Adhoc Network (VANET). Traditionally, VANET worked in a centralised system controlled by a single management authority. The centralized system threatened its members once the attacker compromised it. Also, a centralized system was prone to a single point of failure due to excessive load and bottleneck problems. To prevent centralized systems from excessive load and bottleneck problems, the authors proposed using a decentralized system with blockchain. Vehicles were moving on the road in groups; their parameters like speed, location etc., are communicated to the roadside unit by the onboard unit installed in the vehicle. The roadside unit transfers these real-time vehicle parameters to the certification authority and other core network servers. All data in the core network was stored in a private blockchain to make it more secure.

Chung et al. [99] proposed the use of blockchain in cognitive manufacturing. Day by day, competition in the market was increasing; companies had started to attract customers by offering personalized product customisation. These customizations on products increased raw materials variety, so many changes in process management. All data from product customization to manufacturing to delivery is stored in blockchain. It helped in understanding customer trends and demand. Sensors used in the manufacturing process were used for monitoring purposes. Data generated by these sensors were stored in the blockchain and could be accessed to detect any deviation in required parameters.

Sun et al. [100] proposed blockchain in sharing-based smart cities. Blockchain could be used by humans, organizations and technology to build trust in smart cities. For building trust in sharing transactions, blockchain played an important role. Data received from various IOTs in smart cities could be stored with the help of blockchain, building trust in sharing-based services among businesses. Security provided by blockchain builded trust in decentralized nodes, which might be used for transactions, IOT's or services sharing-based smart cities.

B.vinod [101] proposed the use of blockchain in business related to travelling. Loyalty bonuses for an airline could use digital tokens, which could be accessed by using cryptocurrency. Interline charges were converted to cryptocurrency, which could be taken by the following airline. A private blockchain could be used for contracts between airlines and agents to track sales records, which helped secure payments. The issue was raised when multiple sites booked a property. To eliminate this problem, blockchain could be used, which helped in tracking the booking record of a property. Smart contracts could be generated using machine learning and stored in blockchain.

Han et al. [102] proposed the use of blockchain to store patients' medical records in a hospital. Every hospital chain stored patient health data in their centralized server, a soft target for cyber-attacks like wannacry ransomware attacks. For preventing Cyber-attacks, patient data could be stored in a de-centralized form using a hybrid blockchain. Patient health data was stored in a private blockchain at the hospital level; if the patient allowed it to share among other entities of the hospital chain, it is further stored in a consortium blockchain. Two blockchains were working, one at the hospital level and the other at the hospital chain level providing more security in the de-centralized form.

Ryu et al. [103] proposed using blockchain for IoT digital forensics. With the technological advancement over time, the exponential growth of IoT had happened. IoT could communicate as per requirement, in the cloud, on the network or directly. For digital forensics, all three areas of cloud, network and devices could be explored. Diversification of IoT's type and usage had made digital forensics difficult. Authors had proposed blockchain to store communication details of IOTs by which digital forensics could be refined. Blockchain could be accessed by any of the participants' device users, device manufacturers, service providers and investigators for digital forensics.

2.9 RESEARCH GAPS AND PROBLEM FORMULATION

After extensive research review following research gaps were found:

- Due to privileges being given to insiders, existing authorisation techniques are not fully capable of protecting ITH.
- Insiders are capable of deleting or corrupting system logs to remain hidden.
- Insider is aware of threat prevention and detection techniques.
- Applications of existing Authorisation architectures are found in many areas and can handle a limited number of requests.
- Scalability is affected when the number of PEP increases in existing architecture, the performance of PDP starts decreasing in handling growing number of PEPs.
- No MSS exists to store and monitor messages passing between PEP and PDP servers, as insiders manage these servers .

A new CAC mechanism is required, which can monitor insider's activity even after the insider comes to know about the threat prevention mechanism. CAC mechanism does not allow changing, deleting, or corrupting his system logs. A MSS is required to monitor messages passing between PEP and PDP servers; insiders cannot change, delete or corrupt these messages. As seen from the literature, Blockchain proves to be robust. Capabilities of Existing architectures should be enhanced to handle more requests and prevent the non-responsiveness nature/ failure of PDP in case of more requests.

CHAPTER-3

SYSTEM LOG PROTECTION MECHANISM

3.1 INTRODUCTION

Employees working in CSP acts as an insider to the system. Responsibility of insiders has been increased with the wider reach of cloud computing in many areas [104-107]. To access the system, these employees need to authenticate themselves so that system recognizes them, as seen in the literature. Employee activity and authentication details are managed by a system analyst in a centralized server location in a company, as shown in Figure 3.1. This information is crucial to a company or organization. Access management plays a vital role in the management of this information.

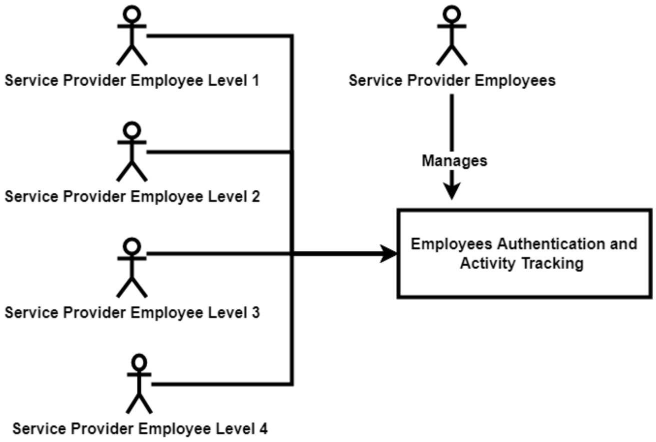

Service Provider Employee Level 1
Service Provider Employee Level 2
Service Provider Employee Level 3
Service Provider Employee Level 4
Service Provider Employees
Manages
Employees Authentication and Activity Tracking

Figure 3.1: Employee’s authentication and activity tracking at service provider end.

During the last decade, many IA were linked with an unethical insurrection by the employees. This information is again managed by employees of a company, as these employees enjoy various privileges given to them.These privilages may be used to misuse information. These insiders either delete or corrupt log files to remain hidden, misusing their privileges. They may become a potential threat related to information misuse of data breaches in provisions to detect IA are not in place.

Many IA detection techniques require log files or generate new log files based on new parameters such as biometric, geo-social activities or physical traits; if an insider is also aware of this, he can delete/corrupt these files, making the task more difficult as seen in Figure 3.2.

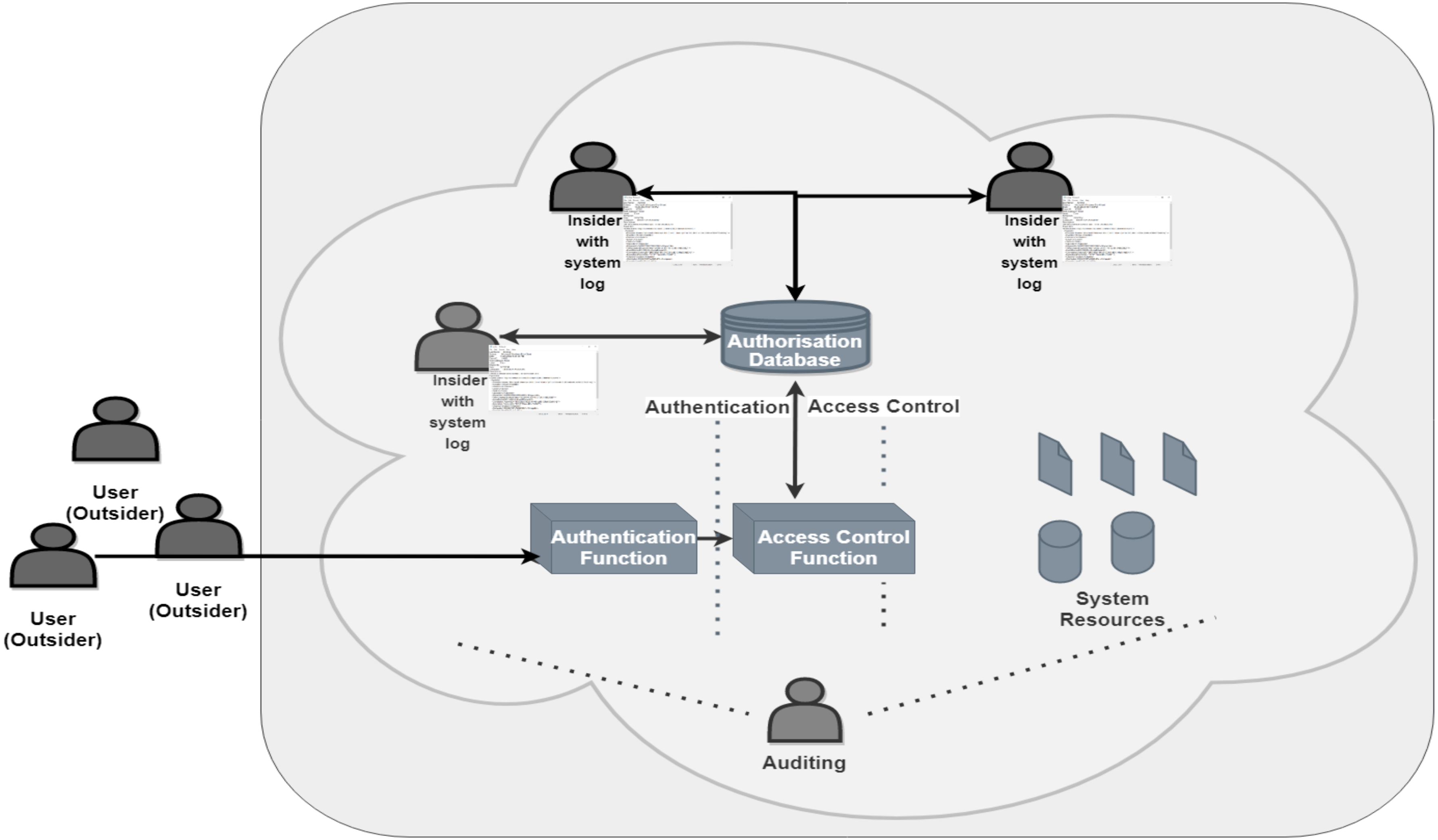
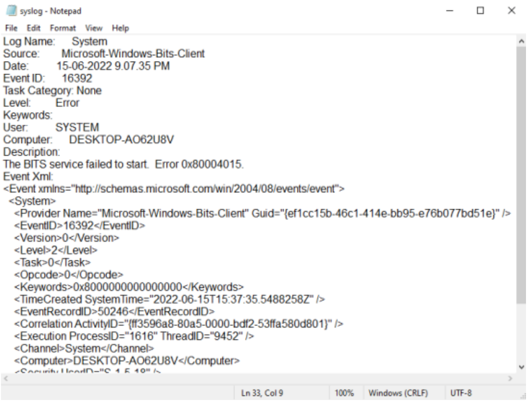


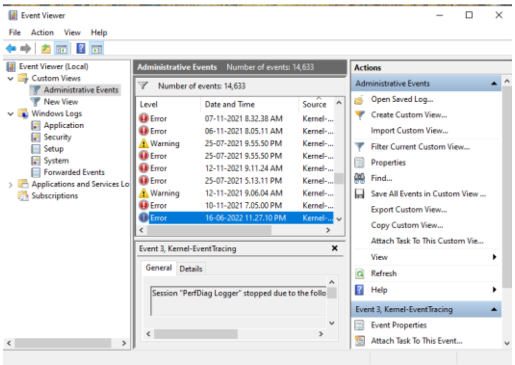
Figure 3.2: Insiders with accessible system logs



```
Log Name:      System
Source:        Microsoft-Windows-Bits-Client
Date:          15-06-2022 9.07.35 PM
Event ID:      16392
Task Category: None
Level:         Error
Keywords:
User:          SYSTEM
Computer:      DESKTOP-AO62U8V
Description:
The BITS service failed to start.  Error 0x80004015.
Event Xml:
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Bits-Client" Guid="{ef1cc15b-46c1-414e-bb95-e76b077bd51e}" />
    <EventID>16392</EventID>
    <Version>0</Version>
    <Level>2</Level>
    <Task>0</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8000000000000000</Keywords>
    <TimeCreated SystemTime="2022-06-15T15:37:35.5488258Z" />
    <EventRecordID>50246</EventRecordID>
    <Correlation ActivityID="{ff3596a8-80a5-0000-bdf2-53ffa580d801}" />
    <Execution ProcessID="1616" ThreadID="9452" />
    <Channel>System</Channel>
    <Computer>DESKTOP-AO62U8V</Computer>
```

Figure 3.3: A text file containing system log

Figure 3.3 shows data related to insider activities is stored in log files. Log files can be accessed from the event viewer in the windows operating system, as shown in Figure 3.4.



Level	Date and Time	Source
Error	07-11-2021 8.32.38 AM	Kernel-...
Error	06-11-2021 8.05.11 AM	Kernel-...
Warning	25-07-2021 9.55.50 PM	Kernel-...
Error	25-07-2021 9.55.50 PM	Kernel-...
Error	12-11-2021 9.11.24 AM	Kernel-...
Error	25-07-2021 5.13.11 PM	Kernel-...
Warning	12-11-2021 9.06.04 AM	Kernel-...
Error	10-11-2021 7.05.00 PM	Kernel-...
Error	16-06-2022 11.27.10 PM	Kernel-...

Session "PerfDiag Logger" stopped due to the follo

Figure 3.4: Event viewer in windows operating system

Privileges given to insider makes the task of accessing these log files easy. The main limitation of existing ITH detection techniques is that they can work only on the availability of log files; in contrast, non-availability makes the task difficult and impossible.

3.2 PROPOSED BLOCKCHAIN-BASED AUTHORISATION MECHANISM

Blockchain provides numerous benefits in terms of security; every specialized area wants to take advantage of it. Blockchain ensures immutability, forgery resistance, democratic, double-spend resistance, consistent state of the ledger, resilience and auditable process [108-111]. Immutability means that once a transaction is done on the blockchain, it cannot be altered. Every node in the blockchain cryptographic hash and digital signatures are applied to build it forgery resistant. Every node in the blockchain has equal rights like in a democratic structure, and no one is more powerful than others. All nodes in the blockchain are auditable; all previous nodes are accessible via a hash function. Preventing double-spending in the blockchain allows access to every transaction up to the genesis block.

Blockchain is of three types public, private and consortium blockchain. In a public blockchain, anybody can participate, whereas, in a private blockchain, an authorised user is allowed to participate in a controlled manner by a centralized authority. As in a private blockchain, the number of users is finite; it is less complex than the public blockchain. A pre-selected set of nodes controls the consortium blockchain consensus process; these selected nodes control the Authorisation of nodes. Each node in the blockchain is connected to its previous node, backward to the first node (Genesis node) in the distributed network. Each node stores the hash value of the previous node, which checks the membership of the blockchain. Various parameters are stored in each Blockchain Node (BN), like index value, the hash value of the previous node, timestamp value, merkle tree root hash, data, and nonce value, as shown in Figure 3.5.

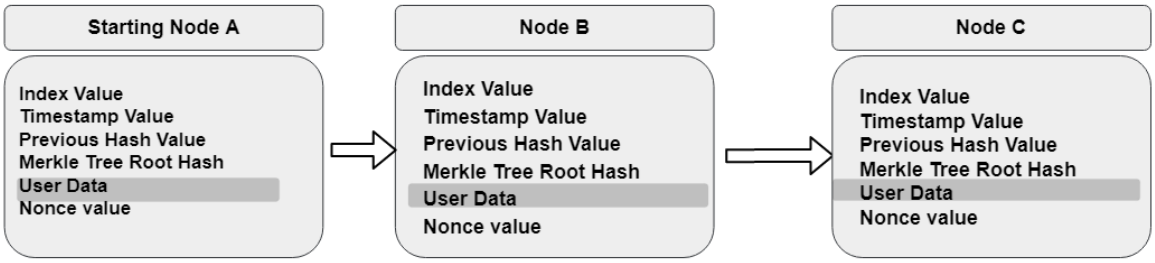


Figure 3.5: BN with user data

It becomes tedious task to change BN as it grows. Robustness can be understood from the requirement that at least 51 per cent of the nodes in a distributed ledger are to be accessed and changed to update any node. Various research articles in the literature recommend blockchain instead of traditional encryption techniques [112-113]. In the proposed blockchain-based authorisation, insider authentication and activity details are stored in a new BN whenever insiders enter the system after they get authenticated. Table 3.1 shows the test bed configuration for the proposed model.

Table 3.1: Blockchain test bed configuration

Sr.No	Configuration	Number
1	Octa Core, 8 GB RAM	1

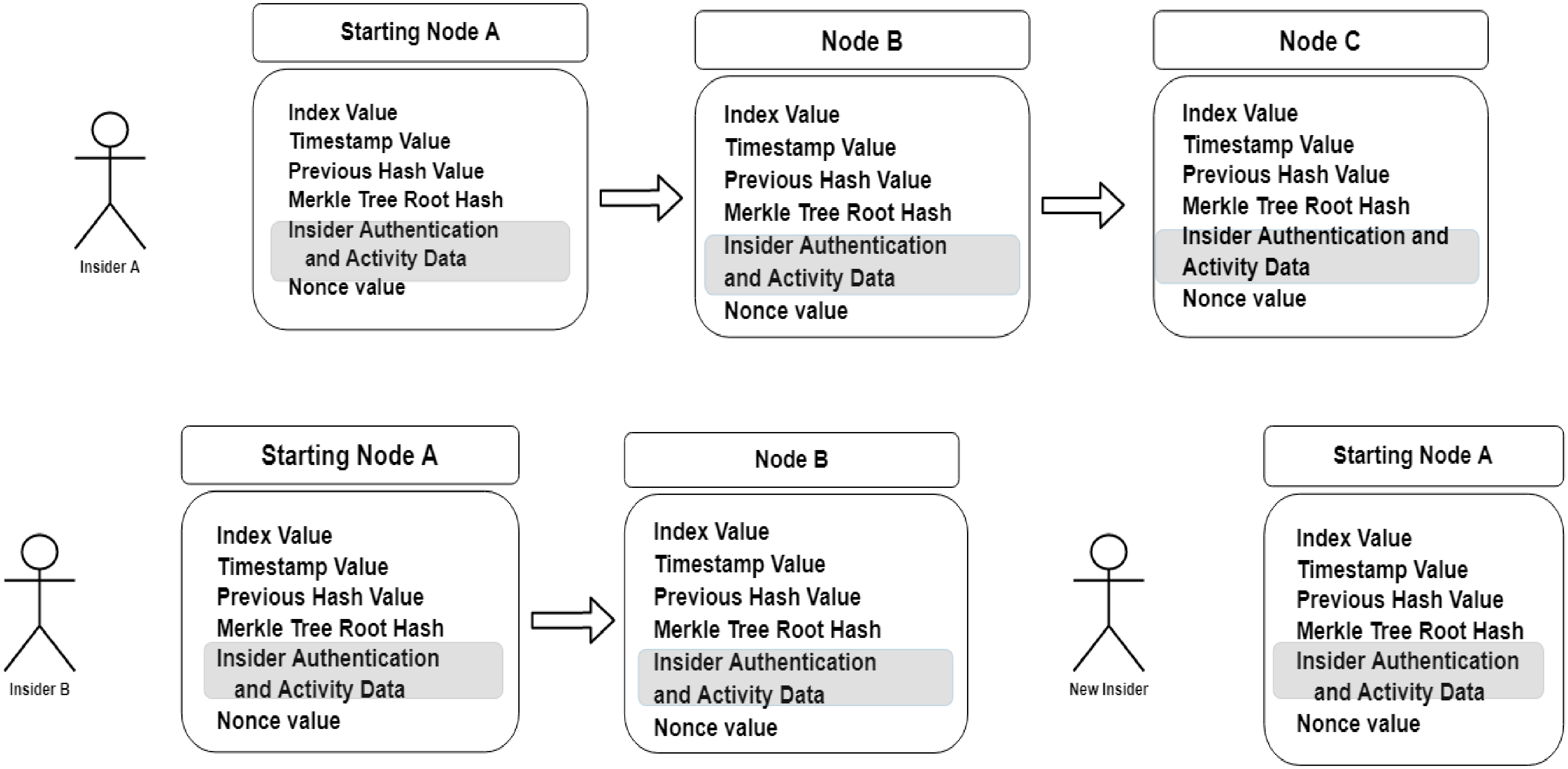


Figure 3.6: Blockchain-based mechanism for various type of insiders

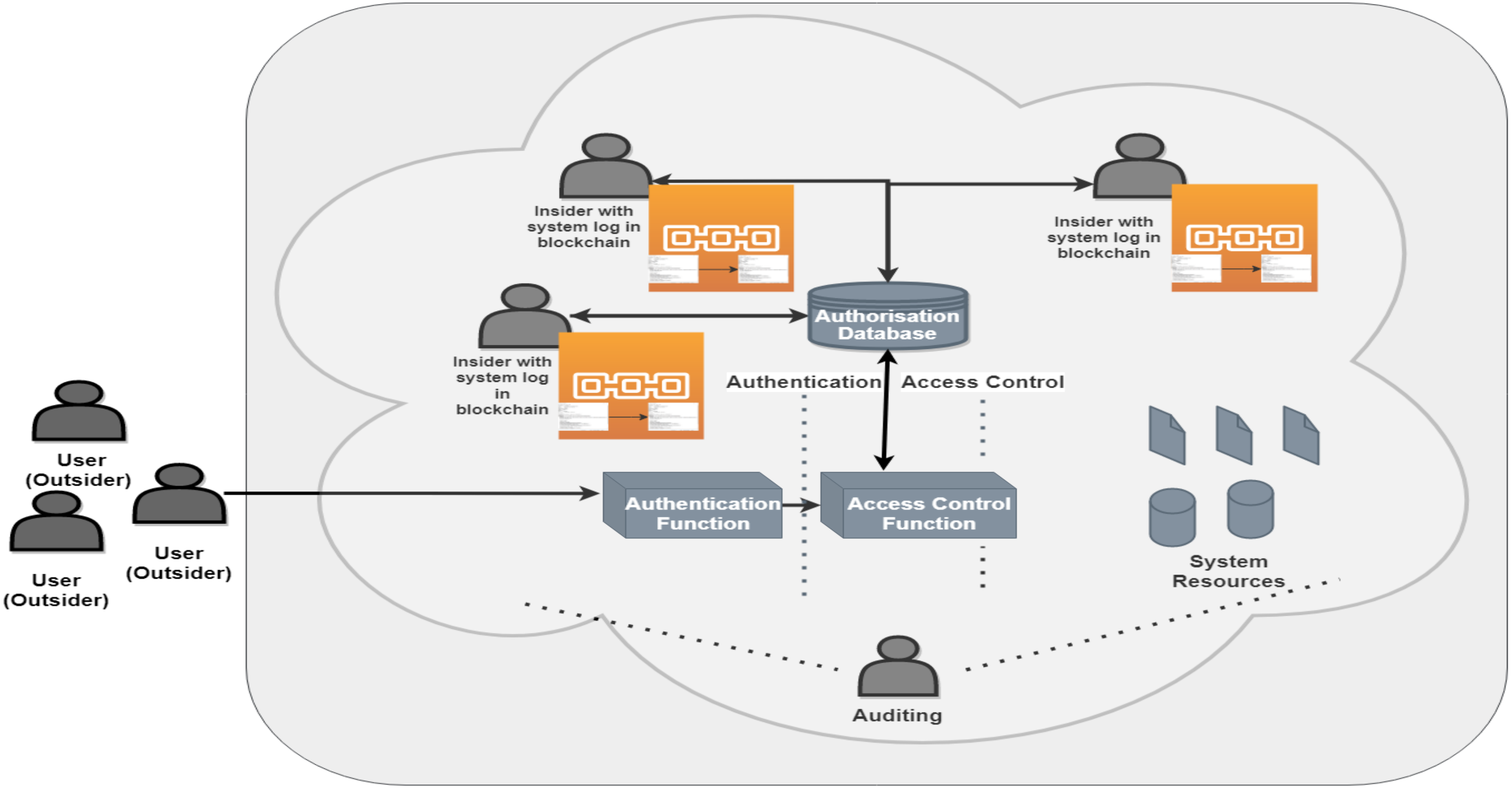


Figure 3.7: Proposed architecture of BN based system log management

It can be seen from Figure 3.6 that insider A has been accessed three times, so three BN have been created, and insider B has been accessed two times, and two BN are created. A single new BN is created, known as the genesis block for new insiders. It becomes challenging for insiders to change their authentication and activity details in a distributed BN, as shown in Figure 3.7.

3.2.1 PROPOSED ALGORITHM FOR BLOCKCHAIN-BASED AUTHORISATION MECHANISM

The proposed algorithm for the blockchain-based authorisation mechanism works in three steps. In the first step, authentication and parameters of the existing blockchain are checked before creating a new BN in Blockchain Server / Blockchain Module. In the second step, a new user profile is created in DB, and a new BN known as the genesis node is created. In case of any discrepancy in Blockchain parameters third step is followed, which gives errors and exit.

Algorithm 1: Blockchain-based authorisation mechanism.

Input: Request Q received at Blockchain Server (BS) in the cloud.

Output: Access granted or rejected.

```
Step 1:  If Login ID & User Signature== valid then continue this step
                 If current index value > last stored index ^hash value ^ timestamp value ^ nonce
                 value == valid then
                         Create a new BN and grant authorisation to store the log file.
                 Else go to step 3.
         Else go to Step 2
Step 2:  If User !=  valid then
                 Add new user node (Genesis block)
                 Initialize index value
                 Allocate current time stamp value
                 Store predefined value in current hash value
                 Store data value
                 Allocate valid nonce value
                 Update user record in blockchain database, grant authorisation to store log file.
Step 3:  Give Error message and Exit
```

In this algorithm, Step 2 is applicable for a new user; authentication for him was done and checked in the previous step. The flow of information can be well understood from the flow chart given in Figure 3.8.

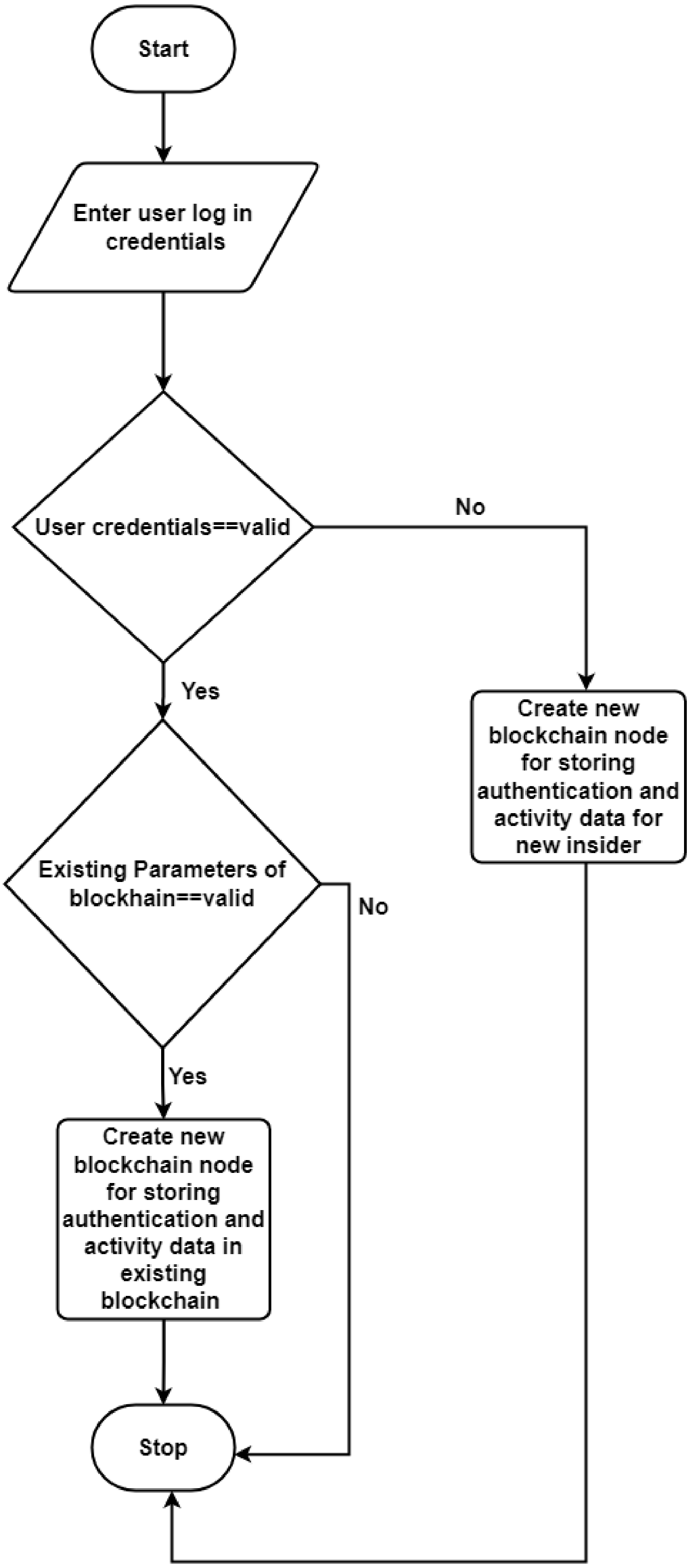

Start
Enter user log in credentials
User credentials==valid
No
Yes
Create new blockchain node for storing authentication and activity data for new insider
Existing Parameters of blockhain==valid
No
Yes
Create new blockchain node for storing authentication and activity data in existing blockchain
Stop

Figure 3.8: Flowchart for blockchain-based authorisation mechanism

3.2.2.1 CRYPTOGRAPHIC REPRESENTATION OF PROPOSED WORK

Insider I sends user log-in info as message m, one-way hash function H(m) is applied on this message, public-key cryptography with nonce value N_i is done and sent to BS as $N_i(H(m)\ K_{Ai})$. After receiving this, BS applies $N_i(c\ )\ K^{-1}{}_{Ai}$ to decrypt. After successfully validating its details new node is created in the BS only if $BC_{iv} > BC_{liv} \wedge H_v \wedge T_v \wedge N_v ==$ valid store I_{aa} details; otherwise, it generates an error in the blockchain. When $N_i(c\ )\ K^{-1}{}_{Ai}$!= valid, create a new node with new insider log-in info details on user request.

3.2.2.2 SECURITY ANALYSIS

Security analysis is carried out to analyze the resistance of the protocol to various attacks. The proposed protocol is secure against the following attacks.

1) Security against replay attack: A replay attack involves capturing the messages exchanged between a valid user and a server and replaying the same later. For successfully launching a replay attack, an adversary should be able to replay a valid login request message $N_i(H(m)\ K_{Ai})$, later. Hence the proposed scheme uses nonce values to resist replay attacks. However, the server verifies the nonce values' freshness before accepting the request and response. Random nonce values used in the proposed scheme viz. N_i is generated independently, and its values are session dependent. Hence attackers cannot gain access to the system by replaying messages previously transmitted by legal users.

2) Security against guessing attack: The data is never transmitted in the plain-text form in the proposed scheme. Moreover, the data is modified into $N_i(H(m)\ K_{Ai})$ before transmitting to the BS. Hence even if the attacker needs to verify the guessed password, it needs to decipher the $N_i(\ c\ )\ K^{-1}{}_{Ai,}$ which is impossible as the private key cryptographic value is available only at authentication and BS.

3) Security against denial–of–service attack: A denial–of–service attack can be launched by an adversary by creating invalid login request messages and bombarding the server with the same or by modifying the current password, which prevents a valid user from accessing resources; he is authorised to access. The adversary cannot create valid login request messages without knowing the appropriate credentials. The password's validity is also checked before creating a login request and allowing a user to modify the current password.

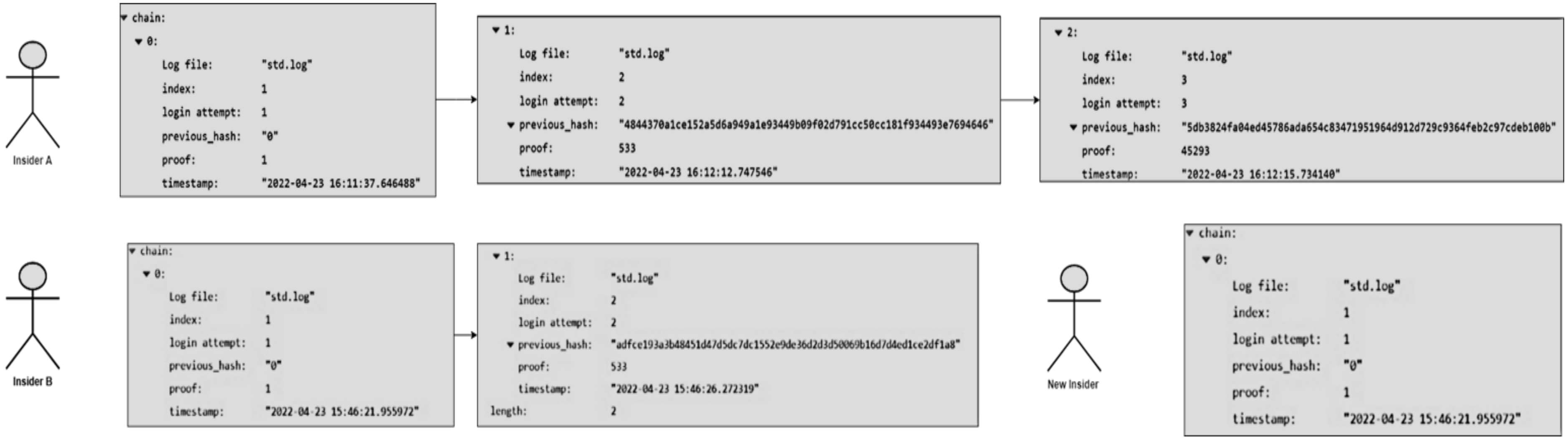
4) Security against server impersonation attack: In a server impersonation attack, an unauthorised server tries to masquerade as a valid server and attempts to obtain the credentials of a valid user. Assume that an adversary intercepts earlier transmissions between the user and the server. For spoofing the server, the adversary should be able to generate the response messages. To compute the public key, the adversary should have knowledge of the server's secret key, which is unknown to the adversary. Also, he cannot calculate the nonce value, which is never transmitted across the communication channel during any session.

3.3 EXPERIMENTAL RESULTS AND DISCUSSION

Blockchain implementation was done in python. Starting node of a blockchain is known as the genesis node. Blockchain uses hashing functions, which are used to convert input data of any length into a fixed-size string. For creating a genesis node, details of various parameters are required, such as an index value refers to the position of the node in a blockchain, timestamp value refers to the time of the creation of a node, address of the previous hash value, in the case of genesis block it is zero, merkle tree root hash is the hashes of all the transactions in a block, insider authentication and activity log data is stored as user data, nonce works as a variable used to find the valid hash.

The python SHA (Secure Hashing Algorithm 256) algorithm is used to create hashing values. This SHA algorithm works as a one-way cryptographic function. As shown in Figure 3.9, three BN have been created in python for insider A as it has accessed the CSP thrice. Similarly is the case for insider B; two BN are created. A new insider genesis node in a blockchain is created. Blockchain Node Mining (BNM) is performed to find valid hash values and the validity of the proposed work in python. Figure 3.10 shows BNM achieved during the implementation.

The proposed work is also tested and validated in scyther's formal method tool. Its adversary model was based on the dolev–yao model [114-115], which is used to analyse security protocols against the capabilities of the adversary, such as an adversary can obtain any message passing through the network, it can send messages to any principal by impersonating another principal, and it can alter the messages.



```
▼ chain:
  ▼ 0:
      Log file:        "std.log"
      index:           1
      login attempt:   1
      previous_hash:   "0"
      proof:           1
      timestamp:       "2022-04-23 16:11:37.646488"
```

```
▼ 1:
      Log file:        "std.log"
      index:           2
      login attempt:   2
    ▼ previous_hash:   "4844370a1ce152a5d6a949a1e93449b09f02d791cc50cc181f934493e7694646"
      proof:           533
      timestamp:       "2022-04-23 16:12:12.747546"
```

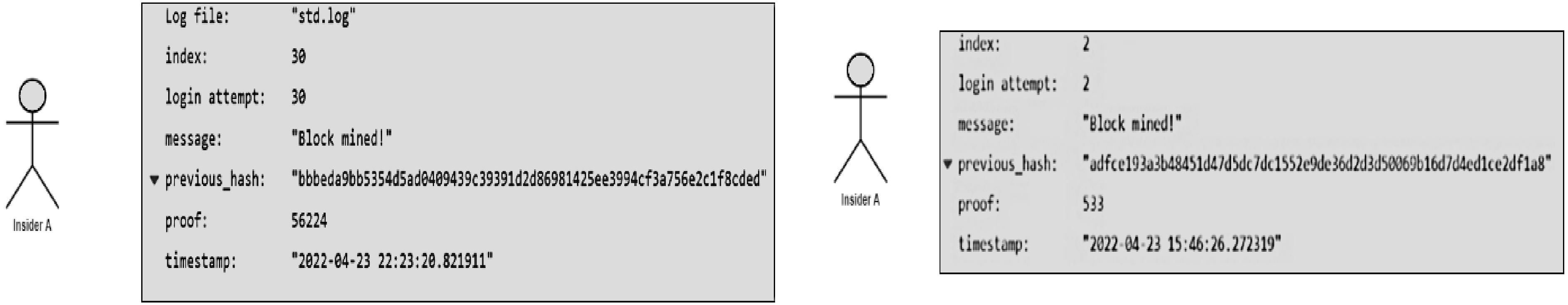
```
▼ 2:
      Log file:        "std.log"
      index:           3
      login attempt:   3
    ▼ previous_hash:   "5db3824fa04ed45786ada654c83471951964d912d729c9364feb2c97cdeb100b"
      proof:           45293
      timestamp:       "2022-04-23 16:12:15.734140"
```

```
▼ chain:
  ▼ 0:
      Log file:        "std.log"
      index:           1
      login attempt:   1
      previous_hash:   "0"
      proof:           1
      timestamp:       "2022-04-23 15:46:21.955972"
```

```
▼ 1:
      Log file:        "std.log"
      index:           2
      login attempt:   2
    ▼ previous_hash:   "adfce193a3b48451d47d5dc7dc1552e9de36d2d3d50069b16d7d4ed1ce2df1a8"
      proof:           533
      timestamp:       "2022-04-23 15:46:26.272319"
length:                2
```

```
▼ chain:
  ▼ 0:
      Log file:        "std.log"
      index:           1
      login attempt:   1
      previous_hash:   "0"
      proof:           1
      timestamp:       "2022-04-23 15:46:21.955972"
```

Figure 3.9: BN for insider A, B and new Insider in python



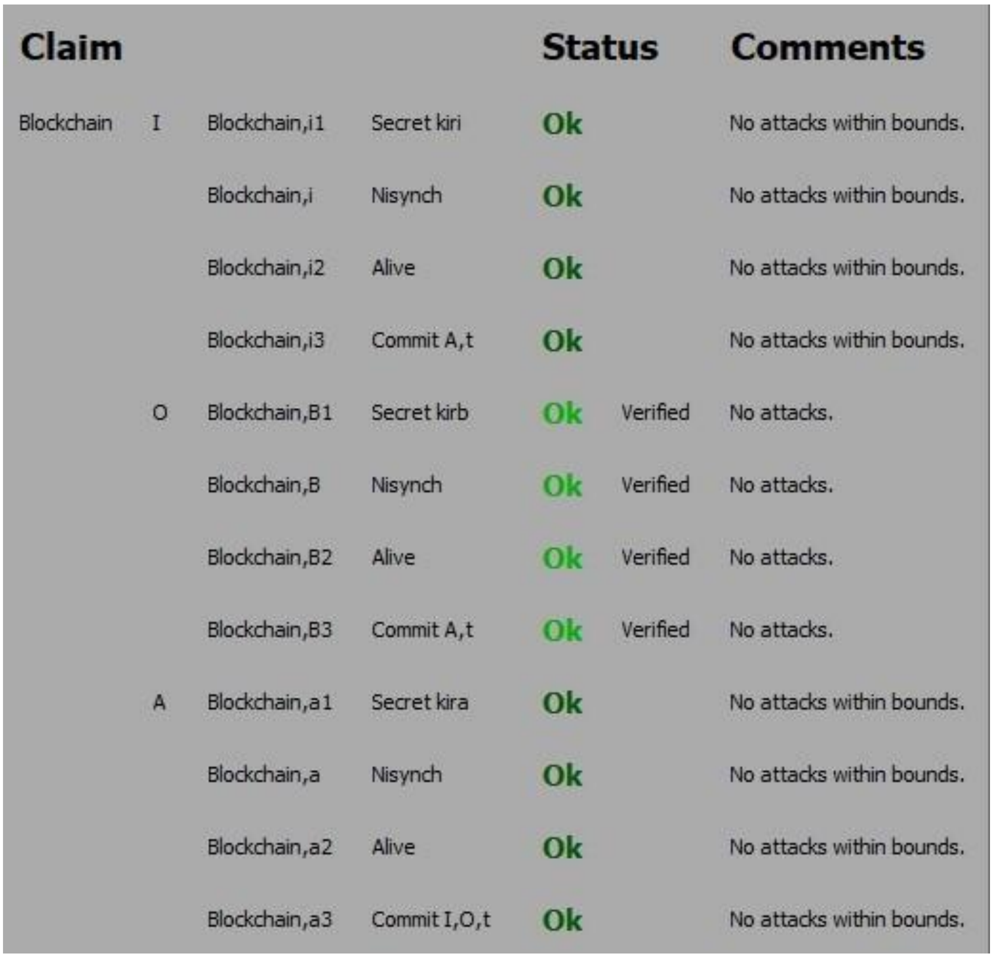
```
  Log file:        "std.log"
  index:           30
  login attempt:   30
  message:         "Block mined!"
▼ previous_hash:   "bbbeda9bb5354d5ad0409439c39391d2d86981425ee3994cf3a756e2c1f8cded"
  proof:           56224
  timestamp:       "2022-04-23 22:23:20.821911"
```

```
  index:           2
  login attempt:   2
  message:         "Block mined!"
▼ previous_hash:   "adfce193a3b48451d47d5dc7dc1552e9de36d2d3d50069b16d7d4ed1ce2df1a8"
  proof:           533
  timestamp:       "2022-04-23 15:46:26.272319"
```

Figure 3.10: BNM for insider

Scyther automatically verifies all the security protocols. It also creates an attack graph for detecting an attack. Claim events were used to represent all the security requirements [116-119], containing four claims: alive, nisynch, secret, and commitment [120-121]. Alive means to achieve the intended communication with some events. Nisynch means non-injective synchronization, which ensures that the intended sender sends all the messages the receiver receives in a synchronized manner. Commitment is a promise made by one party to the other. Confidentiality of user data is achieved by using secret.

The results of the experimentation are shown in Table 3.2. Status Ok means no attacks existed within bounds. The nonce is a session variable, ensuring no old value is reused. Results ensure that all four claims have been achieved and are verified.



Claim				Status		Comments
Blockchain	I	Blockchain,i1	Secret kiri	Ok		No attacks within bounds.
		Blockchain,i	Nisynch	Ok		No attacks within bounds.
		Blockchain,i2	Alive	Ok		No attacks within bounds.
		Blockchain,i3	Commit A,t	Ok		No attacks within bounds.
	O	Blockchain,B1	Secret kirb	Ok	Verified	No attacks.
		Blockchain,B	Nisynch	Ok	Verified	No attacks.
		Blockchain,B2	Alive	Ok	Verified	No attacks.
		Blockchain,B3	Commit A,t	Ok	Verified	No attacks.
	A	Blockchain,a1	Secret kira	Ok		No attacks within bounds.
		Blockchain,a	Nisynch	Ok		No attacks within bounds.
		Blockchain,a2	Alive	Ok		No attacks within bounds.
		Blockchain,a3	Commit I,O,t	Ok		No attacks within bounds.

Figure 3.11: Scyther claim test for robust authorisation mechanism

It can be concluded from the results that the proposed solution resists all the well-known primary attacks. The comparisons between the proposed blockchain-based insider authentication mechanism and other existing authentication mechanisms are presented in Table 3.3.

Table 3.2: Security comparison of the proposed mechanism with literature

	Of-line password guessing attack	**Replay attack**	**DoS attack**	**Insider attack**	**Impersonation attack**
Proposed blockchain authorisation mechanism	Yes	Yes	Yes	Yes	Yes
Silva et al. [62]	No	No	Yes	Yes	Yes
Tsai et al. [63]	Yes	Yes	Yes	No	No
Yang et al. [66]	Yes	No	Yes	No	Yes
Shajina and Varalakshmi [68]	No	Yes	Yes	No	No
Anakath et al. [69]	Yes	Yes	No	No	Yes
Chaudhary et al. [70]	No	Yes	No	No	Yes
Deebak et al. [73]	Yes	Yes	Yes	Yes	Yes

The results show that the proposed mechanism for user authentication withstands all possible attacks and is found suitable to be employed in a real environment for insiders.

3.4 SUMMARY

This chapter addresses the issue of ITH, where insiders with privileges try to remain hidden. They can delete or corrupt system logs, making identifying ITH difficult. Various techniques have been proposed in the literature to tackle this critical issue, but they remain non-effective as they store insider's activity analysis in system logs, and the insider is aware of them. This chapter proposes a blockchain-based robust technique for authorisation of log files of insiders in the cloud environment.

Insider authentication and activity details are stored in the blockchain. Robustness, distributed ledger, immutability and other benefits do not allow insiders to change these system logs. This technique is tested and validated using a scyther formal system tool. The result ascertains that the proposed system is highly significant and efficient and successfully mitigates various ITH. The working of the protocol is also verified based on the four claims, and scyther proved that the proposed protocol is robust enough for real-time implementations. Its working efficiency has also been tested in python by creating BN for multiple users.

The literature shows that no significant work has been done to protect system logs from ITH. The proposed solution greatly solves the ITH issue by providing access control to system logs. The next chapter proposes an improved Authorisation control at the architecture level using multiple PEP-PDP.

CHAPTER-4

AUTHORISATION ARCHITECTURE

4.1 INTRODUCTION

Issues of distributed architectures need to be addressed in cloud environments [122-123].CAC is vital in preventing unauthorised access to users' data stored in the cloud. The first step is to recognize legitimate insiders to a CSP by applying authentication. In the second step, authorisation policies on the insider are fetched and applied. Authorisation policies are implemented on insiders by implementing PEP-PDP architecture. PEP server is also known as a policy enforcer server; it works nearer to the insider, and the PDP server is known as a policy decider which decides policies.

The user sends a Data Access (DA) request to the PEP-PDP architecture, as shown in Figure 4.1; this DA request is received by the PEP server, which applies the authorisation policy to the insider. The PDP server provides these authorisation policies. The main task of the PDP server is to create authorisation policies or ACLs (Authorisation checklists) based on APL written in authorisation programming languages.

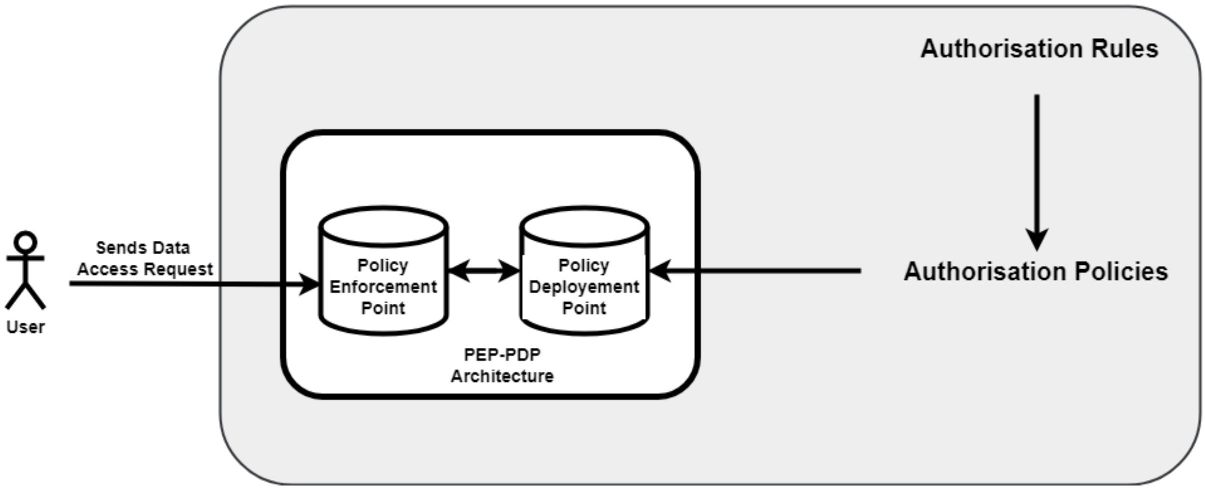


Figure 4.1 DA request in PEP-PDP architecture

PEP-PDP architecture can be categorised into two types, without cache memory and with cache memory. In PEP-PDP architecture without cache memory, as shown in Figure 4.2, when any insider sends a DA request, this request is received by the PEP server. PEP server forwards the request to the PDP server asking for the APL applicable on the insider. Once the PEP server receives APL, it checks against insider request; if it allows, then DA is allowed; otherwise, it is denied.

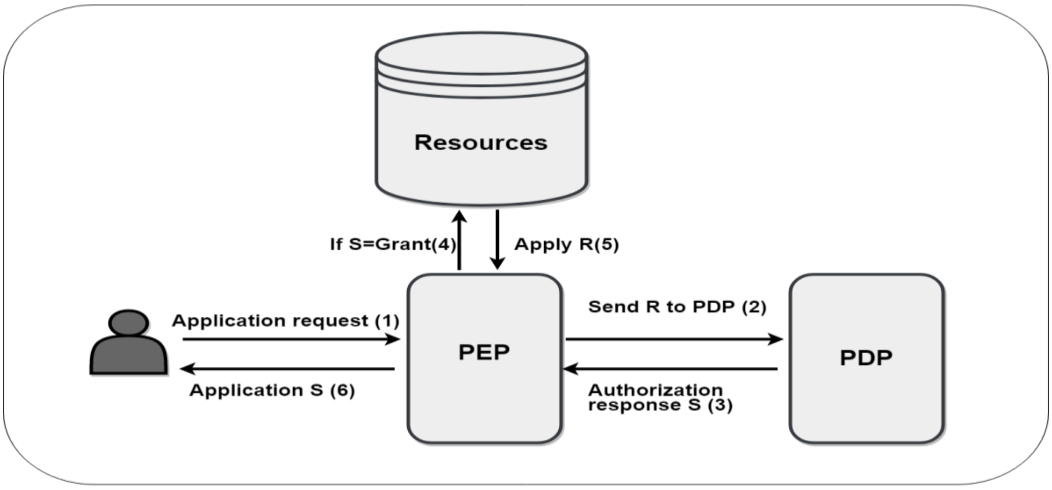


Figure 4.2: Working of PEP-PDP architecture without cache response

As in PEP-PDP architecture with cache memory, as shown in Figure 4.3, when any insider sends a DA request, this request is received by the PEP server. PEP server checks for available APL applicable on the insider request in its cache memory. If it is available and allows data access request, data is fetched from the resources and provided to an insider. If authorisation is unavailable in local cache memory, PEP forwards the request to the PDP server asking for the APL. Once the PEP server receives APL, it checks against insider request; if it allows, then DA is allowed; otherwise, it is denied.

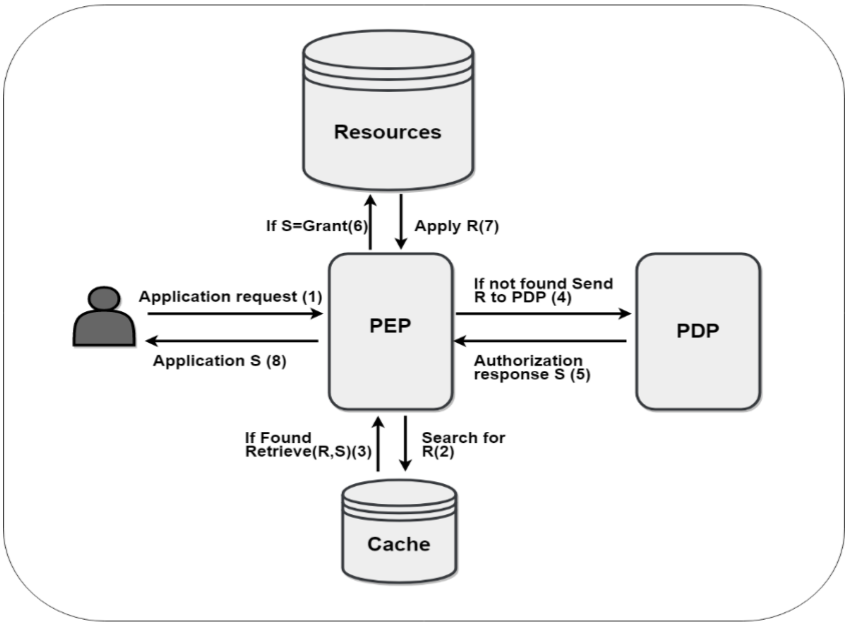


Figure 4.3: Working of PEP-PDP architecture with cache response

PEP-PDP architecture with cache works in two scenarios shown as cases I and II in Figure 4.4. PEP handles insider requests with APL available with local cache in case I. In case II, PEP forwards request to PDP when it does not have APL.PDP provides APL to PEP, which further acts accordingly.

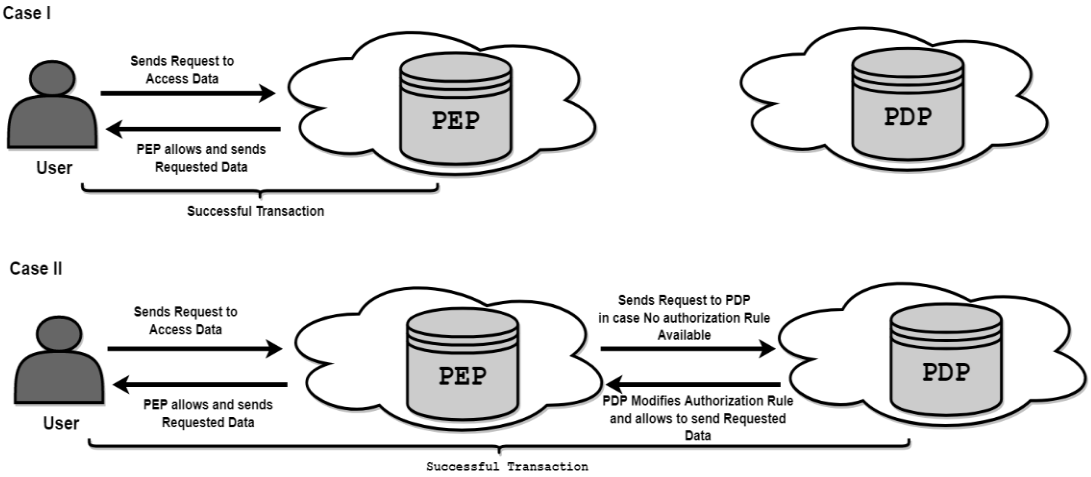

Case I
Sends Request to Access Data
PEP allows and sends Requested Data
User
PEP
PDP
Successful Transaction
Case II
Sends Request to Access Data
PEP allows and sends Requested Data
User
PEP
Sends Request to PDP in case No authorization Rule Available
PDP Modifies Authorization Rule and allows to send Requested Data
PDP
Successful Transaction

Figure 4.4: (Case I) Successful transaction at PEP level (Case II) Successful transaction at PDP level

Yaseen et al. [54] enhanced the existing PEP-PDP architecture with cache, which includes multiple PEPs and a single PDP server. This architecture can handle more requests than existing single PEP-PDP architecture, as shown in Figure 4.5, and a single PDP handles all insider requests received at multiple PEPs.

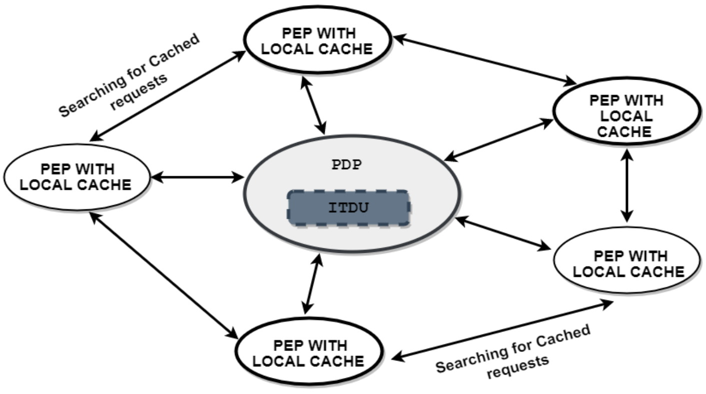


Figure 4.5: ITH aware PEP-Side caching

In this architecture, when an insider sends a DA request to any of the PEP, it checks for the availability of the concerned APL with the local cache. If authorisation policy is unavailable at request receiving PEP, the request is broadcast to all the neighbouring PEPs requesting the concerned APL in their respective local cache. In either case, APL is not available at the PEP level, or neighbouring PEPs level request is forwarded to PDP asking for the concerned APL.

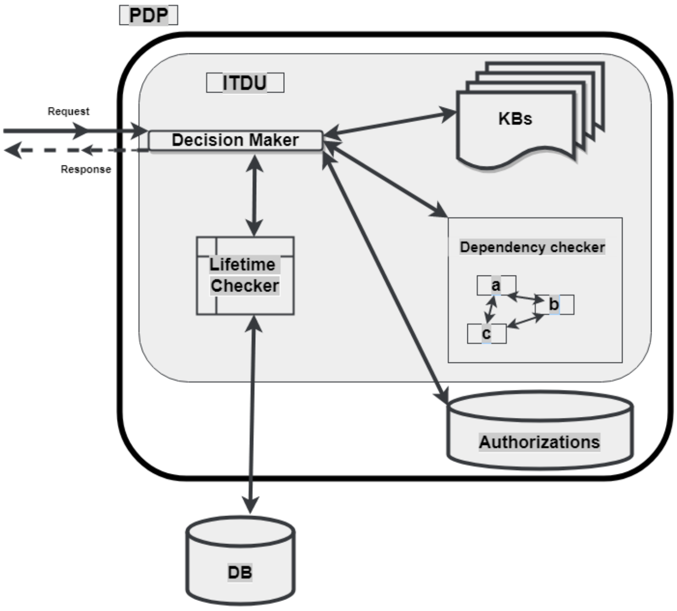


Figure 4.6: ITDU at the PDP side

PDP checks the insider request against any threat in its ITDU as shown in Figure 4.6. This ITDU consist of various components working under it, such as Decision-Maker (DM), Knowledge Base store (KBs), lifetime checker, Dependency Checker (DC), DB, and Authorisations. DM extracts a knowledge base from the KBs regarding the DA request received. The lifetime of the data is checked in DB, and it is updated in KBs. DM contacts the DC to check if there is any dependency between already provided data and requested data. DM takes the final decision after the result is received from DC; in case the result poses a threat, then the request is rejected; otherwise, it is accepted. The decision is communicated to requesting PEP and to all other corresponding PEPs. The information flow can be well understood from the flow chart given in Figure 4.7.

Yaseen et al. [54] proposed multiple PEPs working with a single PDP. This type of architecture is also known as centralized architecture. Single PDP handles requests coming from all the PEPs. As the number of PEPs increases, it degrades the performance of PDP for handling requests. This architecture with a single PDP can handle a maximum 1500 number of requests when the number of insiders is 50, and each insider generates 30 requests. So there is a limit to increasing the number of PEPs, which has become a limitation of this architecture. Failure or non-responsiveness of PDP fails the whole architecture.

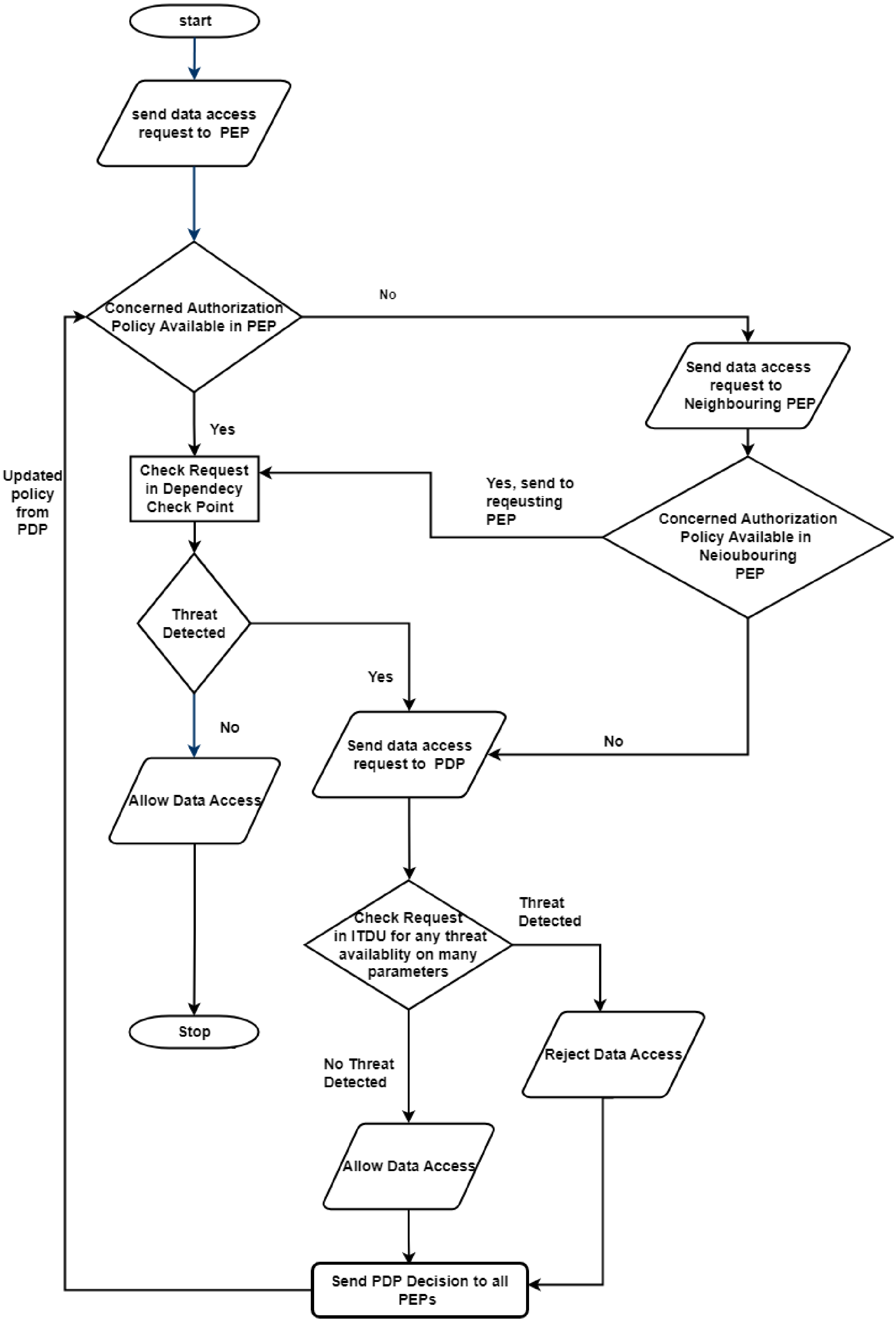

start
send data access request to PEP
Concerned Authorization Policy Available in PEP
No
Send data access request to Neighbouring PEP
Yes
Updated policy from PDP
Check Request in Dependecy Check Point
Yes, send to reqeusting PEP
Concerned Authorization Policy Available in Neioubouring PEP
Threat Detected
Yes
No
Send data access request to PDP
No
Allow Data Access
Check Request in ITDU for any threat availablity on many parameters
Threat Detected
Stop
Reject Data Access
No Threat Detected
Allow Data Access
Send PDP Decision to all PEPs

Figure 4.7: Flow Chart of multiple PEP-single PDP architecture

4.2 PROPOSED DISTRIBUTED AUTHORISATION ARCHITECTURE

The scalability issue of the architecture of **Yaseen et al.** [54] is handled in the proposed distributed authorisation architecture. The proposed architecture is an extension of **Yaseen et al.** [54]. In distributed authorisation architecture, it is scaled up with multiple PEPs and PDPs to work together. Simulation work was carried out on multiple computer systems where 4 computer systems were PDP, and 24 computer systems were PEP, as listed in Table 4.1.

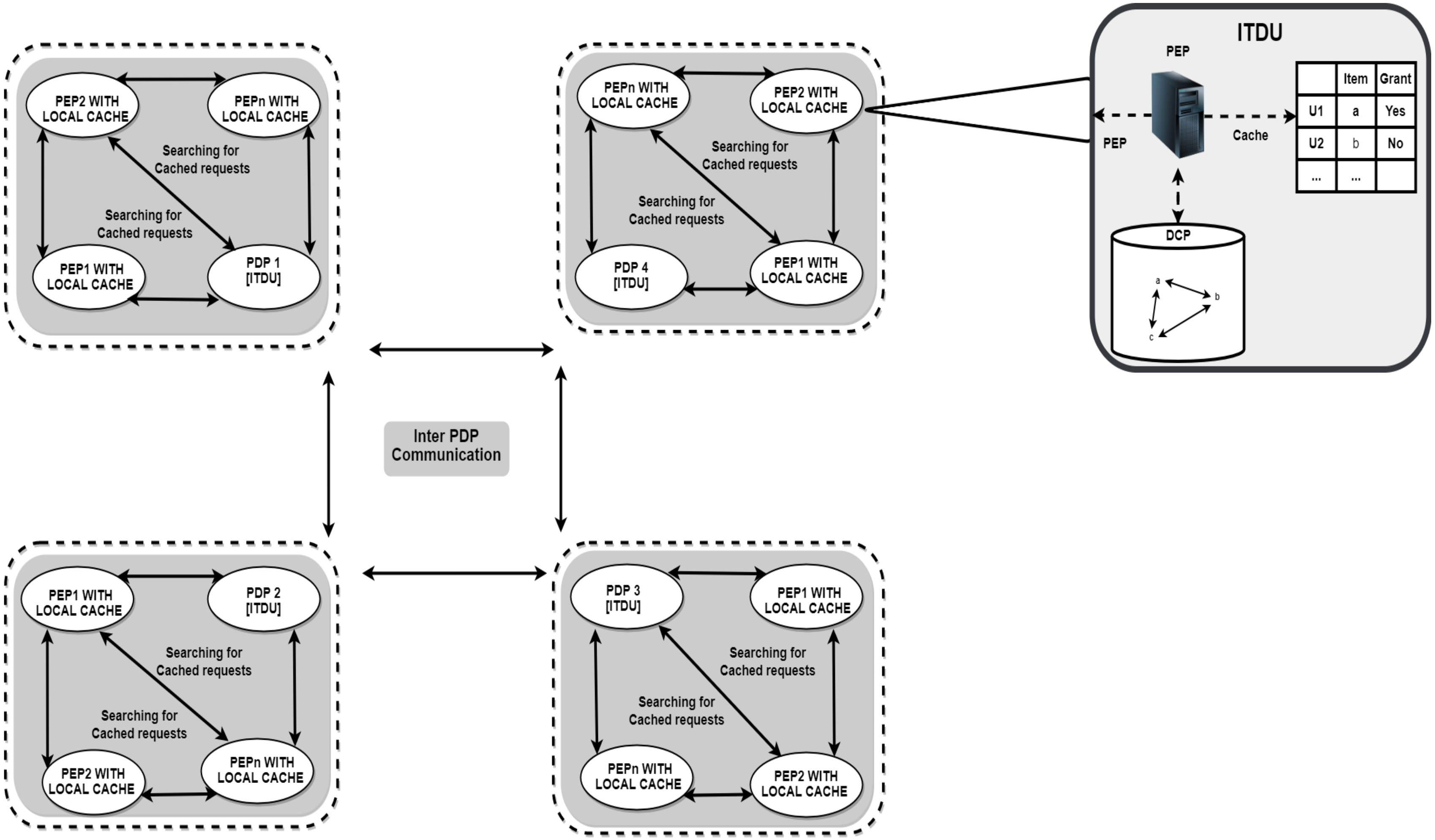
Table 4.1: Distributed architecture test bed configuration

Sr. no	Role	Configuration	Number
1	PDP	Octa Core, 8 GB RAM	4
2	PEP	Octa Core, 8 GB RAM	24

Multiple blocks are working in parallel to each other, each consisting of a single PDP and multiple PEP, as shown in Figure 4.8. Multiple PDP can communicate with each other via inter PDP communication network. One block consists of multiple PEP and a single PDP. The working of this block is replicated.

In a block, when an insider sends a DA request to any PEP, the request receiving PEP checks for the availability of APL in its local cache. When it is found, dependencies between the user's already-provided data and the requested data are checked in the dependency checkpoint at PEP for threat. If no threat is found, APL allows data to be fetched from resources and provided to the insider.

In another case, when authorisation policy is unavailable at insider request receiving PEP, request for required APL is broadcast to all the neighbouring PEPs asking them to look into their respective local caches. When any PEP provides APL to requesting PEP, it is again checked in the dependency checkpoint. In either case, if APL is unavailable at the PEP level or the neighbouring PEPs level request is forwarded to PDP asking for the concerned APL. The flow of information in all blocks consisting of PDP and multiple PEPs can be well understood from the flow chart in Figure 4.9. As the number of insider requests increases, this architecture can be scaled up.



	Item	Grant
U1	a	Yes
U2	b	No
...	...	

Figure 4.8: Proposed distributed PEP- PDP architecture

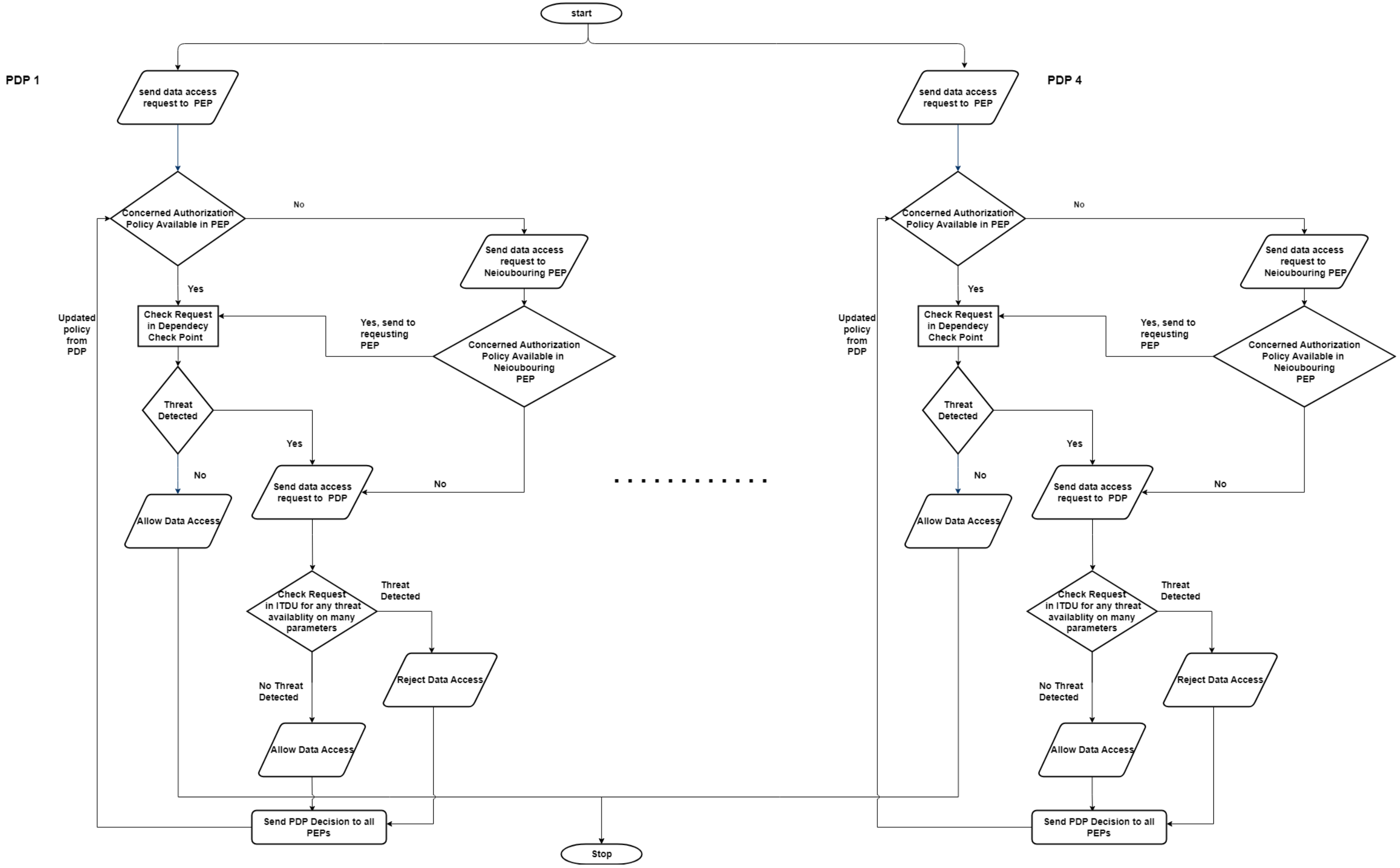

start
PDP 1
send data access request to PEP
Concerned Authorization Policy Available in PEP
No
Send data access request to Neioubouring PEP
Yes
Updated policy from PDP
Check Request in Dependecy Check Point
Yes, send to reqeusting PEP
Concerned Authorization Policy Available in Neioubouring PEP
Threat Detected
Yes
No
No
Send data access request to PDP
Allow Data Access
Check Request in ITDU for any threat availablity on many parameters
Threat Detected
Reject Data Access
No Threat Detected
Allow Data Access
Send PDP Decision to all PEPs
Stop
PDP 4
send data access request to PEP
Concerned Authorization Policy Available in PEP
No
Send data access request to Neioubouring PEP
Yes
Updated policy from PDP
Check Request in Dependecy Check Point
Yes, send to reqeusting PEP
Concerned Authorization Policy Available in Neioubouring PEP
Threat Detected
Yes
No
No
Send data access request to PDP
Allow Data Access
Check Request in ITDU for any threat availablity on many parameters
Threat Detected
Reject Data Access
No Threat Detected
Allow Data Access
Send PDP Decision to all PEPs

Figure 4.9: Flow Chart of 4 PDPs working parallel to each other

4.2.1PROPOSED ALGORITHM FOR DISTRIBUTED AUTHORISATION ARCHITECTURE

The proposed algorithm works in each block consisting of multiple PEP and a single PDP. This algorithm mainly consists of two steps; in step 1, the availability of authorisation policy is checked in insider request receiving PEP and in all neighbouring PEPs. In the next step, APL is checked and provided by PDP in case APL is unavailable in step 1.

```
Algorithm: Distributed authorisation architecture.
Input: An insider alice access request Q for a data item D, received by PEP in one of the blocks.
Output: Access decision (Grant or Reject)
STEP 1: If Q (alice, D) does exist in PEP caches, then
            Send a request to DCP to check dependencies
                If D can be combined with K to infer information, then
                    If  Alice has a cached value of K, then
                    Forward alice request to the associated PDP to check the possible threat
                    Else
                        No threat found, re-issue the cache response for alice request to D
                    End If
                End If
        Else
            If Q (Alice, D) does  not exist in PEP caches, then
                Send Q (Alice, D) to all neighbouring PEP respective cache,  heuristic decision copy
                 when found, send it to PEPs DCP for threat evaluation
            Else
                No decision copy, send Q (alice D) to the associated PDP
            End if
        End if
STEP 2: If alice request is received at the associated PDP, then
            Send a request to ITDU
                If ITDU decides that there is no threat exists, then
                    Alice is allowed to get their requested D; all CPEPs receive associated PDP
                    decision and corresponding  CPEP allows the user to get his requested D.
                Else
                Alice is not allowed to get their requested D; the associated PDP rejects the
                request.
                The response gets updated in all CPEPs
                End If
            End If
```

4.3 EXPERIMENTAL RESULTS AND DISCUSSION

4.3.1 SIMULATION RESULTS

The proposed architecture consists of multiple blocks working parallel to each other, where each block consists of multiple PEP and a single PDP. As the number of blocks has increased which results in speedup in handling the same number of requests as compared to the architecture proposed by **Yaseen et al.** [54].

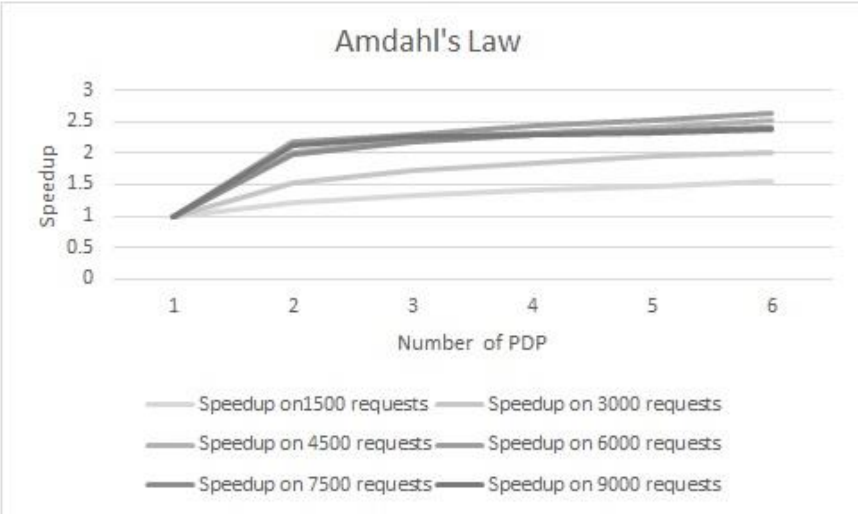


Figure 4.10: Result of Amdahl's Law on the Proposed and Existing Architecture

It can be observed from Figure 4.10 that the proposed architecture reaches speedup factor 2.5 with 6 number of PDPs on handling 9000 requests. The improvement in scalability has resulted in other parameters also, such as:-

1. Requests handled w.r.t data dependency.

Figure 4.11 shows the number of requests handled in the proposed architecture in contrast to [54] as the number of dependencies (%) increases. Existing architecture can handle at the most 1500 requests. In contrast, the proposed architecture performs much better in handling requests. 2 PDP architecture handles 2500 requests, 3 PDP architecture handles 4000 requests, and 4 PDP architecture handles 5000 requests when the number of dependencies increases.

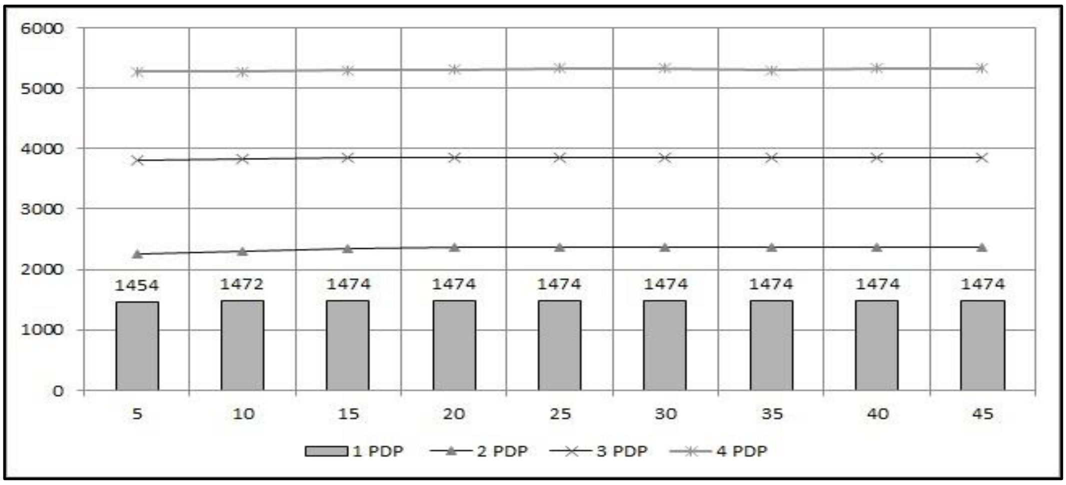


Figure 4.11: Requests handled w.r.t data dependency

2. Possible threats detected w.r.t data dependency.

Figure 4.12 shows the number of possible threats detected at PEP as the number of dependencies (%) increases. Existing architecture can detect 78 threats, whereas proposed architecture performs much better in detecting threats. 2 PDP architecture detects 85 threats, 3 PDP architecture detects 85 threats and 4 PDP architecture 90 threats when the number of dependencies increases.

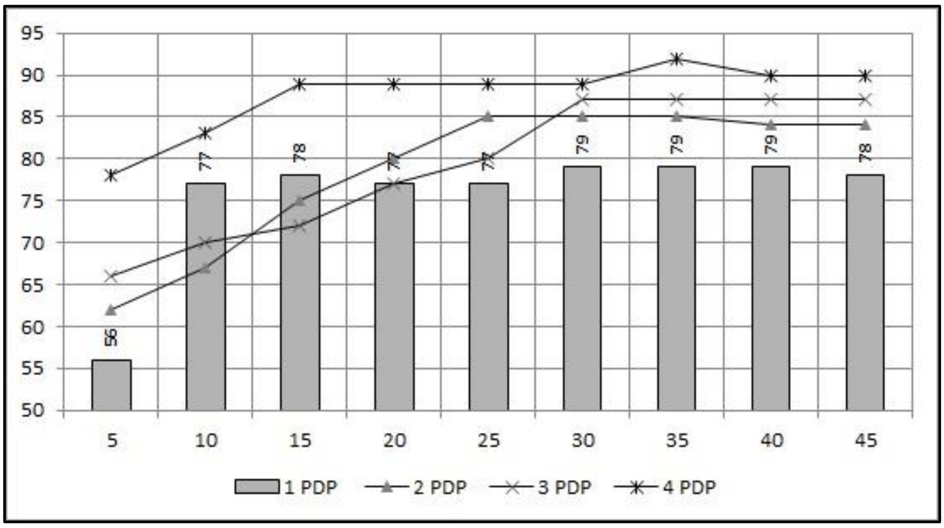


Figure 4.12: Possible threats detected in PEP w.r.t data dependency

3. Requests passed to PDP w.r.t requests per insider.

Figure 4.13 shows the number of requests passed to PDP in the proposed and existing architecture. As the number of requests increases, the probability of finding APL in PEP and neighbouring PEP local cache decreases by which requests are forwarded to PDP. Existing architecture can pass 5050 requests when the number of requests per insider reaches 110, whereas proposed architecture performs significantly better in passing requests to PDP. 2 PDP architecture can pass 8000 requests, 3 PDP architecture can pass 14000 requests, and 4 PDP architecture can reach 20000 requests when the number of requests per insider reaches at 110.

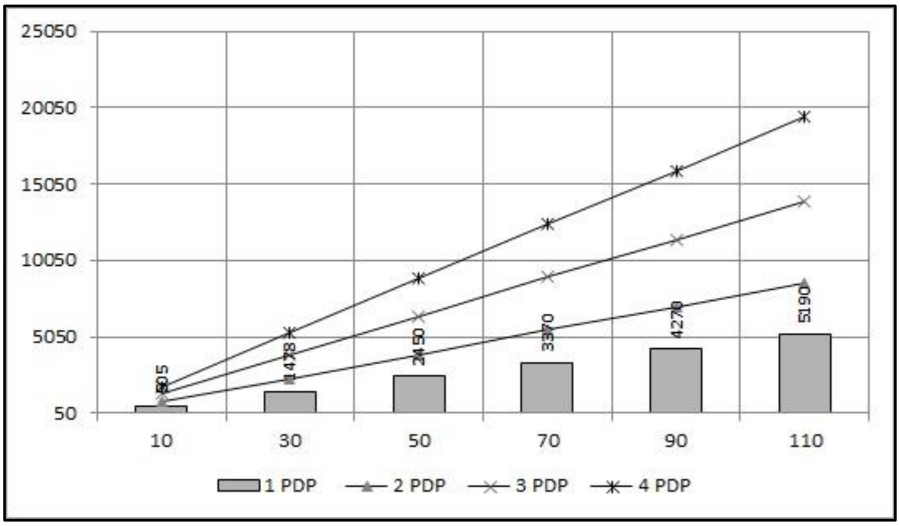


Figure 4.13: Requests passed to PDP w.r.t requests per insider

4. Requests passed to PDP w.r.t number of data items available.

Figure 4.14 shows the number of requests passed to PDP in proposed and existing architecture. As the number of data items increases, the probability of finding APL for data items in PEP and neighbouring PEP local cache decreases by which requests are forwarded to PDP.

Existing architecture can handle 1500 requests when the number of data items available is 550. In contrast, the proposed architecture performs significantly better in passing requests to PDP. 2 PDP architecture can pass 2200 requests, 3 PDP architecture can pass 3800 requests, and 4 PDP architecture can pass 5050 requests when the number of data items available is 550.

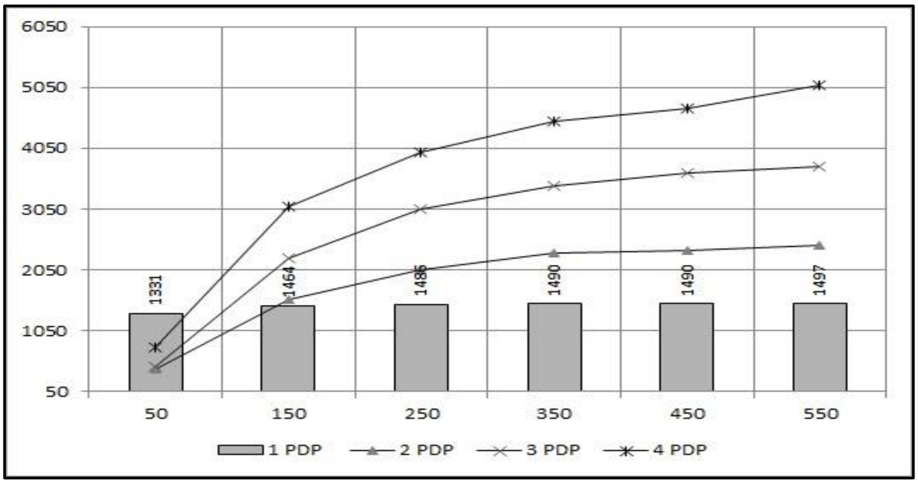


Figure 4.14: Requests passed to PDP w.r.t number of data items available

5. First hits of various PDP architectures w.r.t number of PEPs.

Insider sends DA request to PEP, PEP searches for APL in its cache. When PEP founds APL in its cache, it is known as the first hit. It can be observed from Figure 4.15 that the proposed architecture performs significantly better than the existing architecture in finding authorisation policies at the PEP level only.

As the number of PEP are increased at the same number of requests, the probability of finding authorisation policies decreases in both existing and proposed architectures.

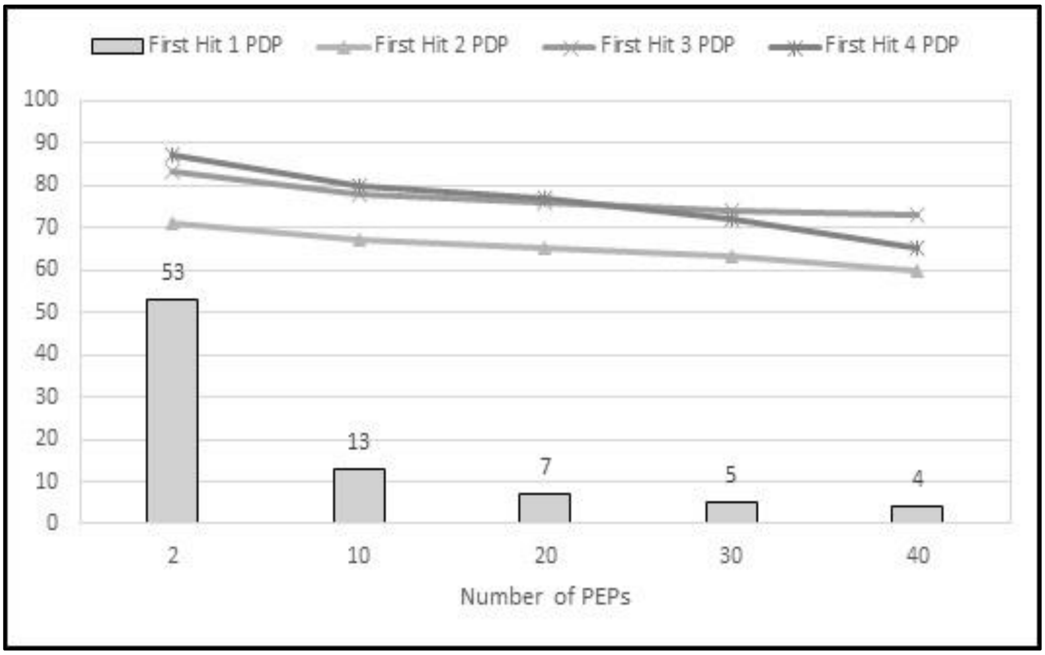


Figure 4.15: First hits of various PDP architectures w.r.t number of PEPs

6. Collaborative hits of various PDP architectures w.r.t number of PEPs.

Insider sends DA request to PEP, PEP searches for APL in its cache. When PEP founds APL in its cache, it is known as the first hit. At the next level, it searches for APL at neighbouring PEPs. If APL is not found at the PEP level and neighbouring PEP level, the request is handled by PDP. Collaborative hits refer to the collection of all the hits for finding APL.

It can be observed from Figure 4.16 that the proposed architecture performs significantly better in contrast to existing architecture in finding APLs.

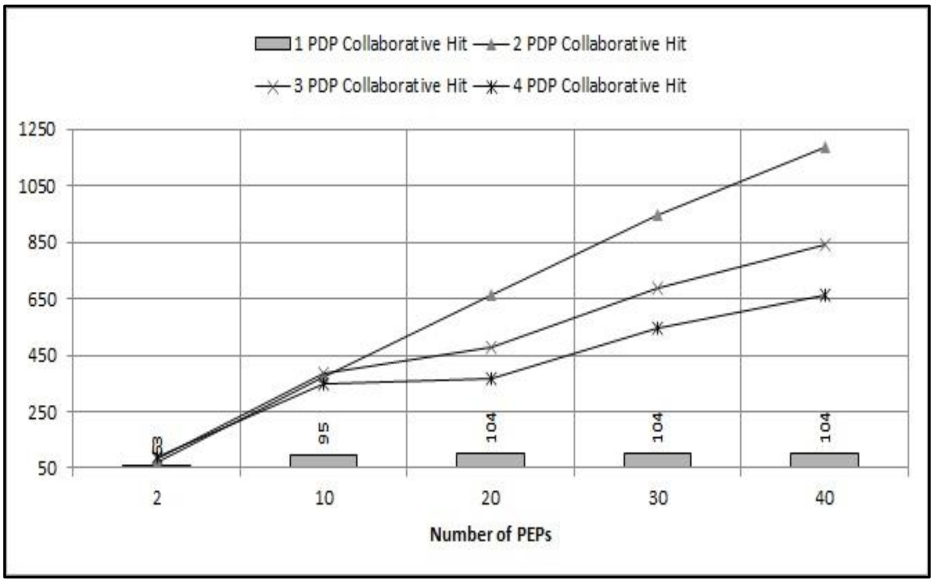


Figure 4.16: Collaborative hits of various PDP architectures w.r.t number of PEPs

4.3.2 STATISTICAL TEST WITH ANALYSIS OF VARIANCE TEST (ANOVA)

Various static methods, like the Z and T-test, are applied only to two group values. The Chi-square test finds the expected value between three or more groups. These tests are not valid here as we are interested in analyzing the variance between the group values [54] and the values of the proposed work. Therefore, the variance test (ANOVA) analysis is applied to determine whether there is any statistical difference between the means of three or more independent groups [124-126].

Null and alternate hypotheses have been chosen for different proposed model variants for each measured point, as shown in Table 4.2. ANOVA tests were conducted at a significance level of 5%.

Table 4.2: Null and alternate hypotheses for ANOVA test

	1. Requests handled w.r.t data dependency.	**2. Possible threats detected w.r.t data dependency.**	**3. Requests passed to PDP w.r.t requests per insider.**	**4. Requests passed to PDP w.r.t number of data items available.**	**5 First hit and collaborative hits w.r.t number of PEPs.**

Null hypothesis	For the Same percentage of dependencies existing and proposed architecture are behaving same.	For the same percentage of dependencies existing and proposed architecture are behaving same.	For the same set of requests/insider existing and proposed architecture are behaving same.	For the same set of the number of data items for the existing and proposed architecture are behaving same.	For the same set of the first hit for a number of PEPs for the existing and proposed architecture are behaving same.	For the same set of the collaborative hit for a number of PEPs for the existing and proposed architecture are behaving same.
Alternate hypothesis	For the Same percentage of dependencies existing and proposed architecture are behaving differently because with the increase in number of PDP performance is improving.	For the same percentage of dependencies existing and proposed architecture are behaving differently because with the increase in number of PDP performance is improving.	For the same set of requests/insider existing and proposed architecture are behaving differently because with the increase in number of PDP, performance is improving.	For the same set of the number of data items for the existing and proposed architecture are behaving differently because with the increase in number of PDP performance is improving.	For the same set of the first hit for number of PEPs for the existing and proposed architecture are behaving differently because with the increase in number of PDPs performance is improving.	For the same set of the collaborative hit for number of PEPs for the existing and proposed architecture are behaving differently because with the increase in number of PDPs, performance is improving.

ANOVA test results are shown in Table 4.3.

Table 4.3: ANOVA test results

Sr. no.			Sum of Squares	df	Mean Square	F	Sig.
1)	Requests handled w.r.t data dependency.	Between groups	76991092.53	3	25663697.5093	43300.04735	*p*< .05
		Within groups	18966.2222	32	592.6944		
		Total	77010058.75	35			

2)	Possible threats detected w.r.t data dependency.	Between groups	728.75	3	242.9167	4.43178	*p*< .05
		Within groups	1754	32	54.8125		
		Total	2482.75	35			
3)	Requests passed to PDP w.r.t requests per insider.	Between groups	207846098.8	3	69282032.93	3.56436	*p*< .05
		Within groups	388748813.8	20	19437440.69		
		Total	596594912.6	23			
4)	Requests passed to PDP w.r.t number of data items available.	Between groups	17789333.13	3	5929777.708	5.05083	*p*< .05
		Within groups	23480400.83	20	1174020.042		
		Total	41269733.96	23			
5)	First hit w.r.t number of PEPs.	Between groups	12326.95	3	4108.9833	30.8714	*p*< .05
		Within groups	2129.6	16	133.1		
		Total	14456.55	19			
6)	Collaborative hits w.r.t number of PEPs.	Between groups	826759.75	3	275586.5833	3.33553	*p*< .05
		Within groups	1321943.2	16	82621.45		
		Total	2148702.95	19			

It can be analysed from Table 4.3 that the F-ratio value for requests handled w.r.t data dependencies is 43300.04735. The *p*-value is < .00001. The result is significant at p< .05. It has been interpreted from ANOVA test results that the alternate hypothesis has been accepted which state that with the increase in the number of PDP, the system's performance improves for the same set of dependency percentage.

The f-ratio value for possible threats detected w.r.t data dependency is 4.43178. The p-value is .010278. The result is significant at p < .05.It has been interpreted from ANOVA test results that the alternate hypothesis has been accepted which state that with the increase in the number of PDP, the system's performance improves for the same set of dependency percentages. F-ratio value for requests passed to PDP w.r.t requests per insider is 3.56436. The p-value is

.032536. The result is significant at p<.05. It has been interpreted from ANOVA test results that the alternate hypothesis has been accepted which state that with the increase in the number of PDP, the system's performance is improving for the same set of requests/insider.

F-ratio value for requests passed to PDP w.r.t number of data items available is 5.05083. The p-value is .009127. The result is significant at p<.05. It has been interpreted from ANOVA test results that the alternate hypothesis has been accepted which state that with the increase in the number of PDP, the system's performance is improving for the same set of the number of data items.

Similarly, the F-ratio value for the first hit w.r.t number of PEPs is 30.8714. The p-value is<.00001. The result is significant at p<.05. It has been interpreted from ANOVA test results that the alternate hypothesis has been accepted which state that with the increase in the number of PDP, the system's performance improves for the same set of first hits as the increase in the number of PEPs.

F-ratio value for collaborative hits w.r.t number of PEPs is 3.33553. The p-value is .046029. The result is significant at p<.05. It has been interpreted from ANOVA test results that the alternate hypothesis has been accepted which state that with the increase in the number of PDP, the system's performance improves for the same set of the collaborative hits as the increase in the number of PEPs.The proposed work has been compared to all the most similar work found in literature, as shown in Table 4.4.

Table 4.4: Comparison of the proposed architecture with literature

	Clustering-based request travels from one PDP to another PDP. **(PEP without Cache)**	**Different PDP work for different requirements in coordination with each other.** **(PEP without Cache)**	**Multiple PDP works independently to each other, although connected, can be scaled up or down easily for the same purpose.** **(PEP with Cache)**
Fan Deng et al. [127]	X		
Carvalho et al. [128]		X	
Proposed architecture			X

Fan deng et al. [127] proposed a two-stage clustering approach with PDPs working sequentially on authorisation policies. In this approach, when a request is received and transferred by the request dispatcher, it is further sequentially received by multiple sub-PDPs, at each sub-PDP request; it was matched with the APL. This architecture was based on PEP without cache.

Carvalho et al. [128] proposed a 4PDP4E toolset to protect users' data travelling online. This architecture was based on PEP without cache. This toolset was proposed for data protection directives given by the European Union. In this toolset, 4 PDPs were used for risk management, requirement engineering, model-driven design, and system assurance in the systems development lifecycle. In this, different PDPs work for different requirements in coordination.

In the proposed architecture, multiple PEPs and 4 PDPs are used in simulation in 4 blocks. Even though each block works parallelly in this proposed architecture, they are connected via inter-PDP communication. The proposed architecture was tested statistically using the ANOVA test, and the value of p is less than 0.05, which says that with the increase in the number of PDPs, the system's performance improves for the same set of parameters. The proposed architecture is best suitable for access management in a distributed architecture, where requests migrate on different clouds.

4.4 SUMMARY

This chapter addresses the issue of achieving better Authorisation control at the architecture level in the cloud environment. Work available in literature tried to handle this issue either at the PEP level or at the PDP level. **Yaseen et al.[54]** proposed increasing the number of PEPs; PEPs are based on cache architecture. **Fan deng et al.**[127] and **Carvalho et al.** [128] proposed multiple PDPs based on PEP without cache architecture. Scalability always remains the predominant issue in these proposed architectures; performance suffers significantly while achieving scalability.This chapter proposes a distributed architecture for better insider Authorisation control in the cloud environment. It is based on multiple PEP–PDP servers and is successful in achieving significantly better results in scalability and performance. Significant results have been achieved at multiple parameters of performance; in handling the number of insider requests and detecting possible threats with respect to data dependency, requests passed to PDP with respect to requests per insider and data items available in the case of the first hit and collaborative hits it outperformed than the existing multiple PEP- single PDP architecture. Results have also been validated statically in the ANOVA test, which proves that the proposed system is highly efficient and successful compared to the existing multiple PEP single PDP architecture. The next chapter takes care of the issue of better control and data protection in the proposed distributed architecture for better authorisation control in the cloud environment.

CHAPTER-5

A DISTRIBUTED MESSAGE SECURITY SYSTEM

5.1 INTRODUCTION

The importance of data security in the cloud environment has been raised by various researchers in the literature due to various attacks [129-133]. Data protection in CC is an important area for research, as marked by the European union's general data protection regulations [134-138]. Various cyber-attack detection mechanisms require some characteristics to trace attack activity. Insiders working in CSP can access and modify user authorisation policies. IA detection techniques require system log files or creating log files based on cyber activity, biometrics, etc. Insider is aware of these attack detection techniques and tries to corrupt or delete these files to remain hidden in case of IA [139-145]. In our proposed CAC mechanism for ITH in a distributed cloud environment, we have proposed protection of system logs in the third chapter, enhanced capabilities of existing PEP-PDP architecture in the fourth chapter and a MSS is proposed to store all the messages communicated between PEP and PDP in Blockchain Server (Blockchain Module) to have better control on tracking insider activities in this chapter.

5.2 PROPOSED MESSAGE SECURITY SYSTEM IN DISTRIBUTED AUTHORISATION ARCHITECTURE

Preventing and detecting malicious insider activities at PEP-PDP architecture requires an insider activity storage mechanism. In this chapter, a MSS is proposed in distributed authorisation architecture, which stores messages communicated between PEP and PDP in a BS, as shown in Figure 5.1.

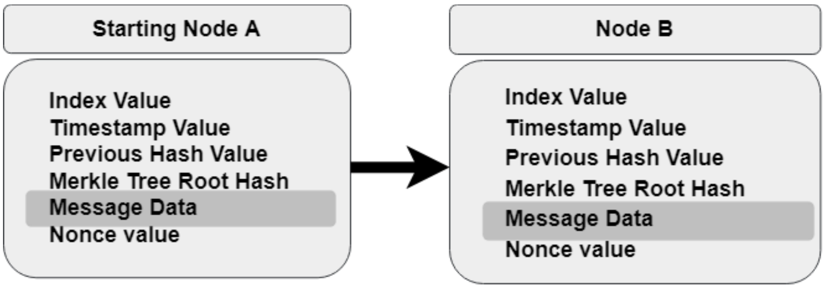


Figure 5.1: Message data in BN

In distributed authorisation architecture, it is scaled up with multiple PEPs, PDPs and BS to work together. Table 5.1 illustrate the a MSS test bed configuration.

Table 5.1: A MSS test bed configuration

Sr. no.	Role	Configuration	Number
1	PDP	Octa Core, 8 GB RAM	3
2	PEP	Octa Core, 8 GB RAM	24
3	BS	Octa Core, 8 GB RAM	3

The working of the proposed architecture can be understood from the working of one block consisting of PEP, BS and PDP, as shown in Figure 5.2. In this block, when any of the PEP receives an insider DA request, it checks for the availability of the concerned APL in its local cache. If available, dependencies between the user's already-provided data and the requested data are checked in the dependency checkpoint available at PEP for any threat. If no threat is found, APL allows data to be fetched from resources and provides back to the insider.

In another case, the authorisation policy is unavailable at the previous step; requests regarding required APL are communicated to all neighbouring PEPs. In case the availability of APL happens, it is provided to requesting PEP and is checked in the dependency checkpoint. When APL is not available at the PEP level or neighbouring PEPs level, the request is forwarded to PDP via BS; this BS first checks request is coming from a legitimate sender; if it is found to be a valid message, BS creates a new BN after checking all the parameters of an existing blockchain.

After storing message details in BN, it forwards them to PDP, asking for the concerned APL. When PDP sends its decision along with APL, it is received by BS. It checks whether it is coming from a legitimate PDP, stores details in already created BN, and forwards it to the concerned PEP. Inter-cluster communication is used in case of any fault or failure in a block. The proposed architecture is scalable to handle the increasing number of requests.

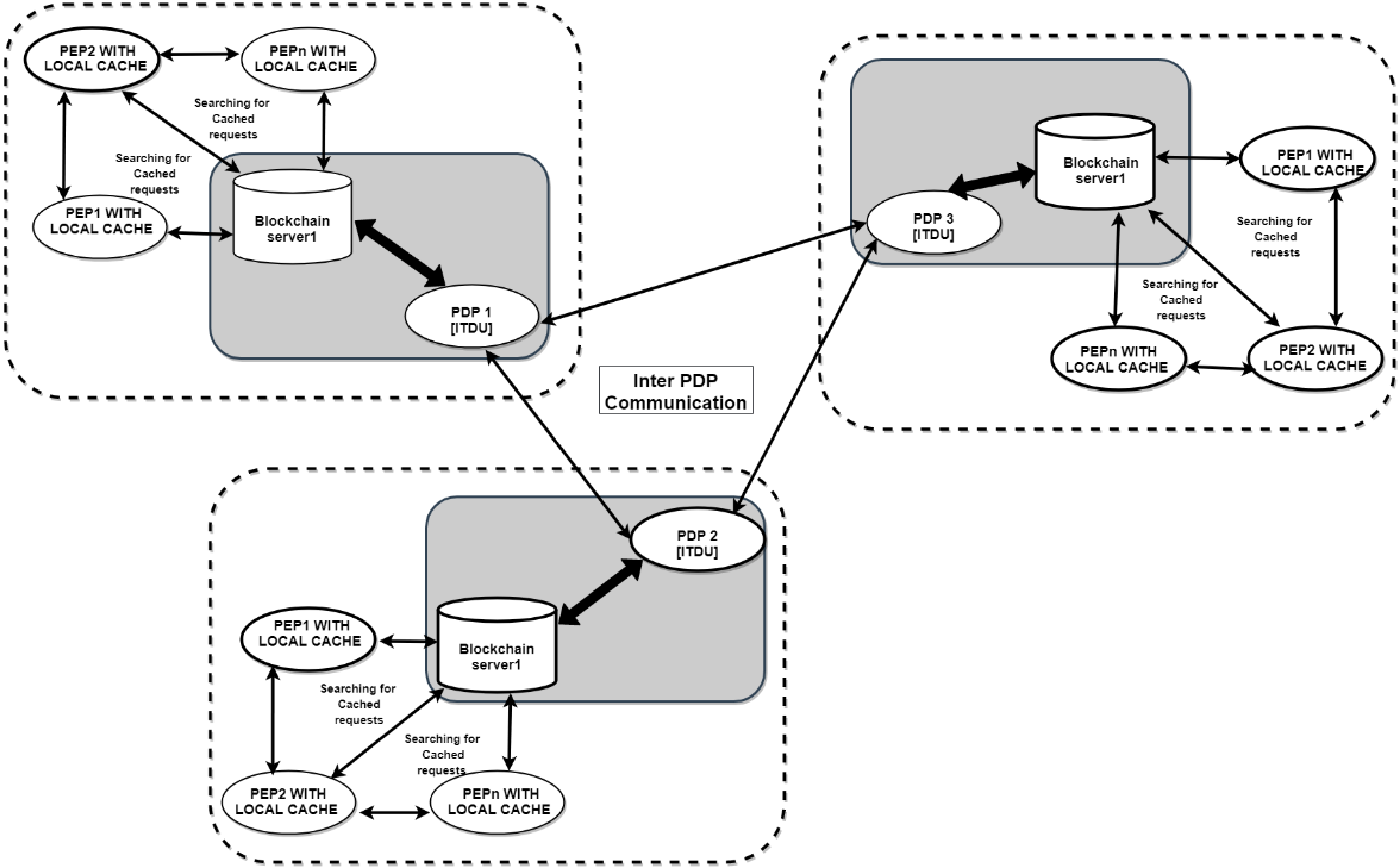


Figure 5.2: Working of proposed MSS

5.2.1 PROPOSED ALGORITHM FOR MESSAGE SECURITY SYSTEM

The proposed algorithm consists of sub-three algorithms working in coordination.

Algorithm 5.1: ITH prevention in distributed PEP-PDP architecture with PEP side caching running in each block.

Input: An insider requests Q for accessing a data item D received by PEP in one of the blocks.

Output: Access decision (Grant or Reject)

```
STEP 1: If Q (alice, D) does exist in PEP caches, then
                  Send a request to DCP to check dependencies
                       If D can be combined with K to infer information, then
                                 If alice has a cached value of K, then
                                 Forward alice request to the associated PDP to check possible threats by calling
                                 algorithm 5.2
                                Else
                                No threat found, re-issue the cache response for alice request to D
                                End IF
                       End IF
          Else
                If Q (alice, D) does not exist in PEP caches, then
                       Send Q (Alice, D) to all neighbouring PEP respective cache, heuristic
                         decision copy when found send it to PEP DCP for threat evaluation
                Else
                        No decision copy, Send Q (alice D) to the associated PDP by calling algorithm 5.2
                 End If
         End if
```

In first algorithm 5.1, an insider DA request is received and checked for availability of APL in PEP and neighbouring PEPs; in case it is not available, a request for providing APL is forwarded to PDP via a BS by accessing algorithm 5.2. BS stores messages received from PEP in BN before sending them to PDP.

Algorithm 5.2: PEP / PDP MSS using blockchain mechanism.

Input: Request Q received at BS.

Output: Access granted or rejected.

```
STEP 1: If Message == PEP/PDP else go to step 3
STEP 2: If current index value > last stored index value & hash value & timestamp value &
        nonce value== valid then
        Create a new BN with requested message details, forward the message to PDP/PEP Server by calling
        algorithm 5.3/5.1,
        Else, go to step 3.
        End If
STEP 3: Give error message and exit
```

Request for providing APL is received at the PDP and checked in ITDU for possible threats in algorithm 5.3. This ITDU consist of various components working under it, such as DM, KBs, lifetime checker, DC, DB, and Authorisations. DM extracts a knowledge base from the KBs regarding the DA request received via algorithm 5.2. The lifetime of the data is checked in DB, and it is updated in KBs. DM contacts the DC to check if there is any dependency between already provided data and requested data. DM takes the final decision after the result is received from DC in ITDU; in case of result poses a threat, then the request is rejected; otherwise, it is accepted.The final decision of the PDP is communicated to the intended PEP via algorithm 5.2.

Algorithm 5.3: Decision on message received by designated PDP server.

Input: Request Q received at designated PDP server for each PEP; it checks for possible threats at ITDU

Output: Request approved or rejected

```
STEP 1: If the designated PDP, using the ITDU, decides that no threat exists, then alice is allowed to get their
        requested D and send the PDP decision to all CPEPs and corresponding CPEP by calling algorithm
        5.2
        Else go to Step 2.
        End If
STEP 2: If the designated PDP, Using ITDU, decides that a threat exists, then alice request is rejected, send
        PDP decision to all CPEPs and corresponding CPEP by calling algorithm 5.2.
```

The flow of information is shown in the flow chart in Figure 5.3. More detailed Working of the Proposed system can be well understood from Figure 5.4, which also describes the role of each algorithm.

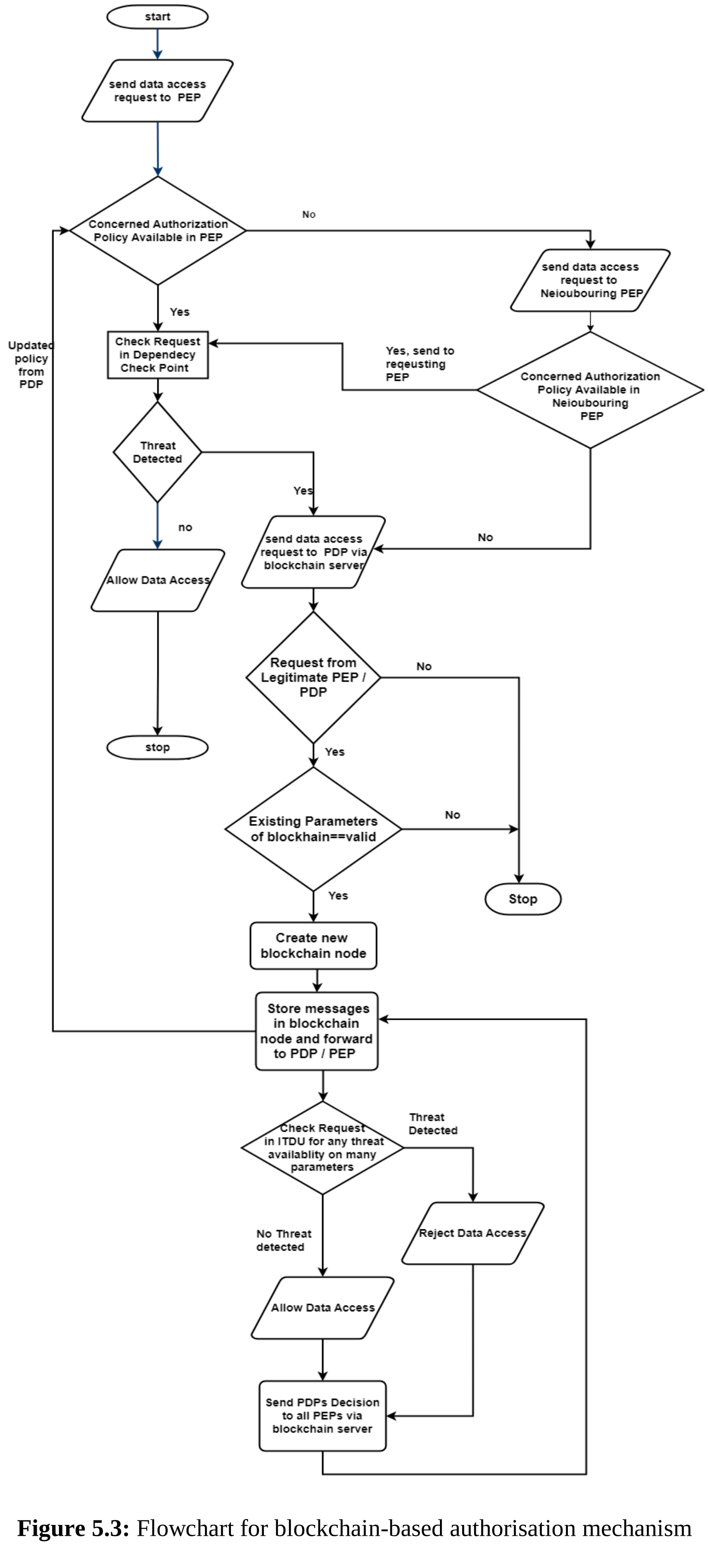

start
send data access request to PEP
Concerned Authorization Policy Available in PEP
No
send data access request to Neioubouring PEP
Yes
Updated policy from PDP
Check Request in Dependecy Check Point
Yes, send to reqeusting PEP
Concerned Authorization Policy Available in Neioubouring PEP
Threat Detected
Yes
no
send data access request to PDP via blockchain server
No
Allow Data Access
Request from Legitimate PEP / PDP
No
stop
Yes
Existing Parameters of blockchain==valid
No
Yes
Stop
Create new blockchain node
Store messages in blockchain node and forward to PDP / PEP
Check Request in ITDU for any threat availablity on many parameters
Threat Detected
No Threat detected
Reject Data Access
Allow Data Access
Send PDPs Decision to all PEPs via blockchain server

Figure 5.3: Flowchart for blockchain-based authorisation mechanism

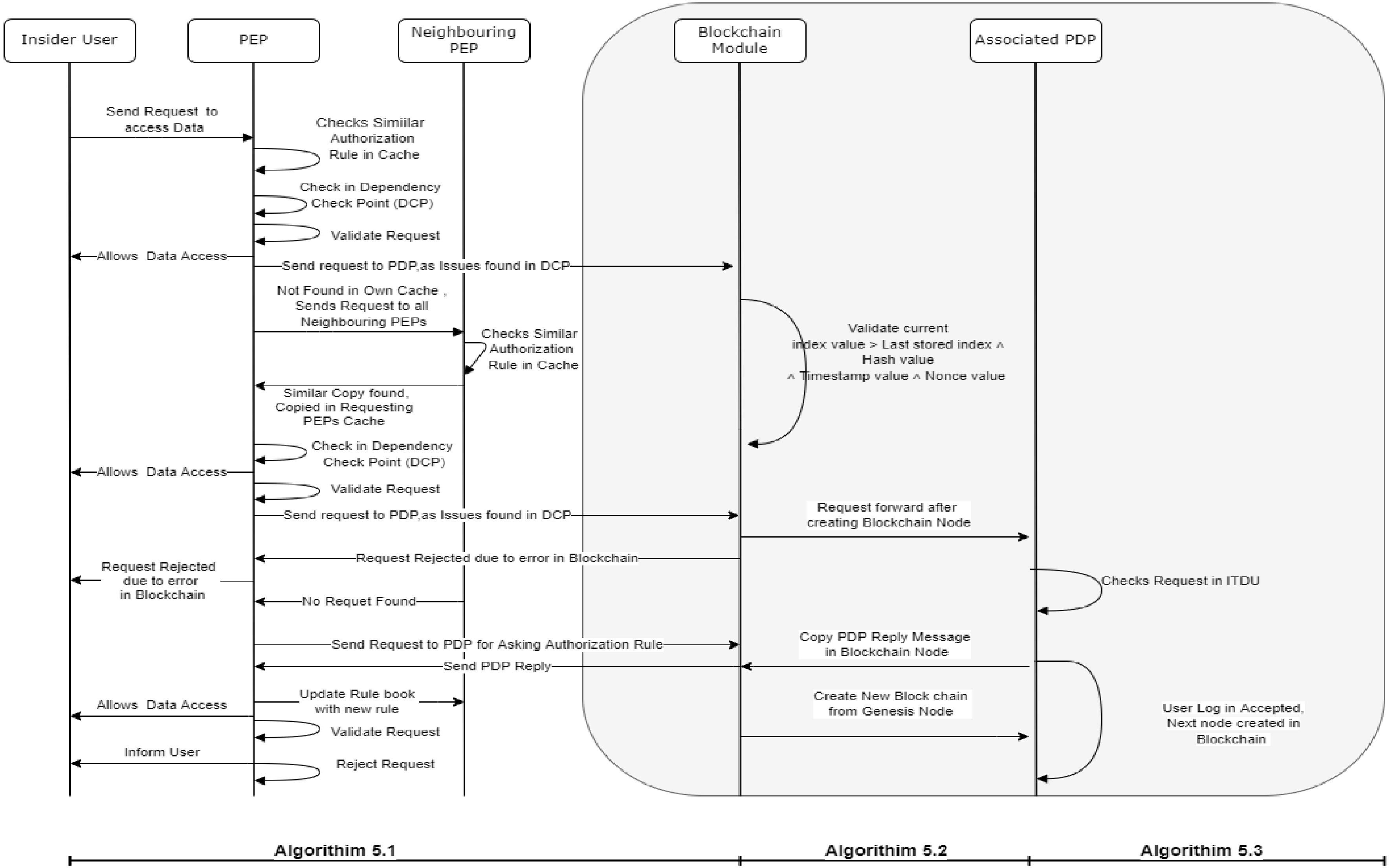

Insider User
PEP
Neighbouring PEP
Blockchain Module
Associated PDP
Send Request to access Data
Checks Simiilar Authorization Rule in Cache
Check in Dependency Check Point (DCP)
Validate Request
Allows Data Access
Send request to PDP,as Issues found in DCP
Not Found in Own Cache , Sends Request to all Neighbouring PEPs
Checks Similar Authorization Rule in Cache
Validate current index value > Last stored index ^ Hash value ^ Timestamp value ^ Nonce value
Similar Copy found, Copied in Requesting PEPs Cache
Check in Dependency Check Point (DCP)
Allows Data Access
Validate Request
Send request to PDP,as Issues found in DCP
Request forward after creating Blockchain Node
Request Rejected due to error in Blockchain
Request Rejected due to error in Blockchain
Checks Request in ITDU
No Requet Found
Send Request to PDP for Asking Authorization Rule
Copy PDP Reply Message in Blockchain Node
Send PDP Reply
Update Rule book with new rule
Allows Data Access
Create New Block chain from Genesis Node
User Log in Accepted, Next node created in Blockchain
Validate Request
Inform User
Reject Request
Algorithim 5.1
Algorithim 5.2
Algorithim 5.3

Figure 5.4: Working of blockchain-based authorisation mechanism

5.2.2.1 EXPLANATION OF PROPOSED BLOCKCHAIN-BASED AUTHORISATION MECHANISM

PEP sends its identification key, insider user info as message m, and APL info required. One way hash function H(m) is applied on this message, public-key cryptography with nonce value N_i is done to make it disguise is sent to the blockchain and PDP server as $N_i(H(m)\ K_{Ai})$. After receiving this, BS applies $N_i(c)\ K^{-1}{}_{Ai}$ to decrypt.

After successfully validating its details new node for that PEP is created in the blockchain list in the server only if $BC_{iv} > BCl_{iv} \wedge H_v \wedge T_v \wedge N_v$ == valid store PEP details; otherwise, it generates an error in the blockchain. The request is forwarded to PDP. PDP checks the request and sends a decision copy or new APL to PEP as a reply $N_i(H(m)\ K_{Ai})$ with a new nonce value and public key. Reply to PEP is also stored in BN of BS. After receiving the message from PDP, PEP applies $N_i(c)\ K^{-1}{}_{Ai}$ to decrypt to get a decision copy or new APL.

5.2.2.2 SECURITY ANALYSIS OF CRYPTOGRAPHIC WORK

Security analysis is carried out to analyse the resistance of the protocol for various attacks. The proposed protocol is secure against the following attacks.

1) Security against replay attack: Replay attack is not possible by attackers as random nonce values used in the proposed scheme are generated independently, and their values are session dependent.

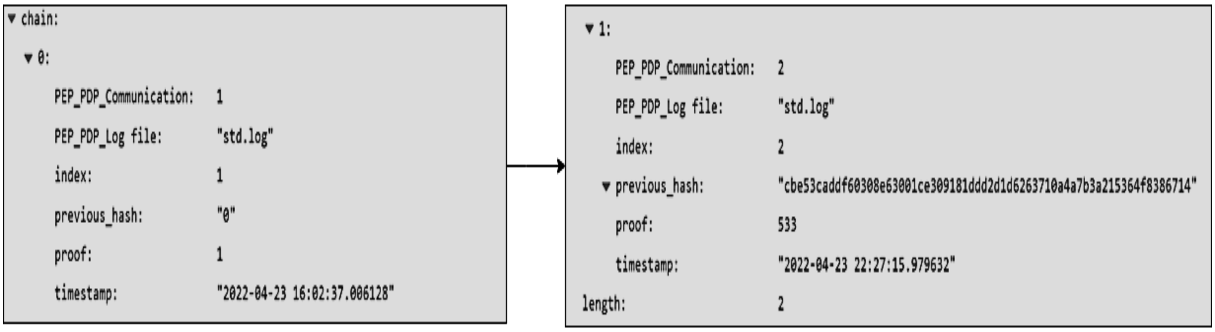
2) Security against guessing attack: The data is never transmitted in the plain-text form in the proposed scheme. Moreover, the data is modified into $N_i(H(m)\ K_{Ai})$ before transmission. Hence even if the attacker needs to verify the guessed password, he needs to solve the $N_i(\ c\ )\ ^{K-1}{}_{Ai}$, which is infeasible as the private key cryptographic value is available only at BS.

3) Security against Denial-of-Service Attack: A denial-of-service attack can be launched by an adversary by creating invalid request messages and bombarding the server with the same or by modifying the current message. The adversary cannot create valid request messages without knowing the correct identification keys of PEP and Servers.

4) Security against server impersonation attack: An impersonation attack is infeasible in this proposed scheme as the unauthorised server cannot know the server's secret key. Also, it cannot calculate the nonce value, which is never transmitted across the communication channel during any session.

5.3 EXPERIMENTAL RESULTS AND DISCUSSION

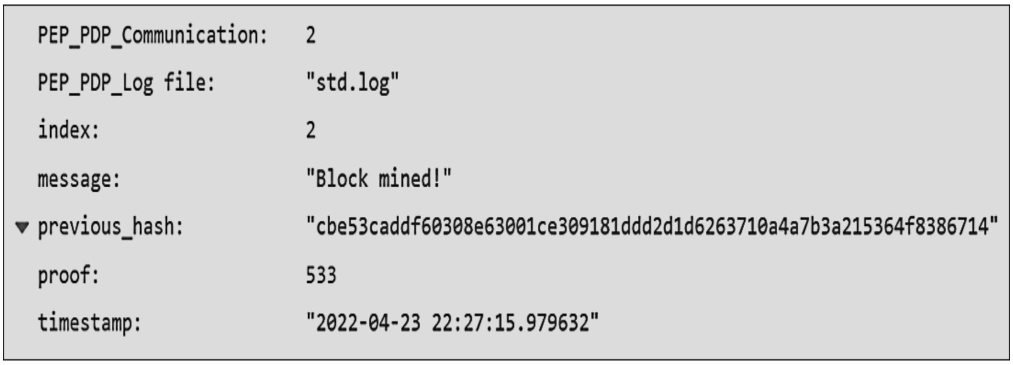
Figure 5.5 shows two BN have been created in Python as two messages have been received at BS. The Secure Hashing Algorithm 256 is used in Python to create hashing values. This SHA algorithm works as a one-way cryptographic function. This Blockchain implementation in Python uses Proof of Work as a Consensus algorithm. Figure 5.6 shows Blockchain Mining (BNM) results achieved in Python, confirming the proposed work's validity.



```
▾ chain:
  ▾ 0:
      PEP_PDP_Communication:   1
      PEP_PDP_Log file:        "std.log"
      index:                   1
      previous_hash:           "0"
      proof:                   1
      timestamp:               "2022-04-23 16:02:37.006128"
```

```
▾ 1:
      PEP_PDP_Communication:   2
      PEP_PDP_Log file:        "std.log"
      index:                   2
    ▾ previous_hash:           "cbe53caddf60308e63001ce309181ddd2d1d6263710a4a7b3a215364f8386714"
      proof:                   533
      timestamp:               "2022-04-23 22:27:15.979632"
  length:                      2
```

Figure 5.5: MSS in BN

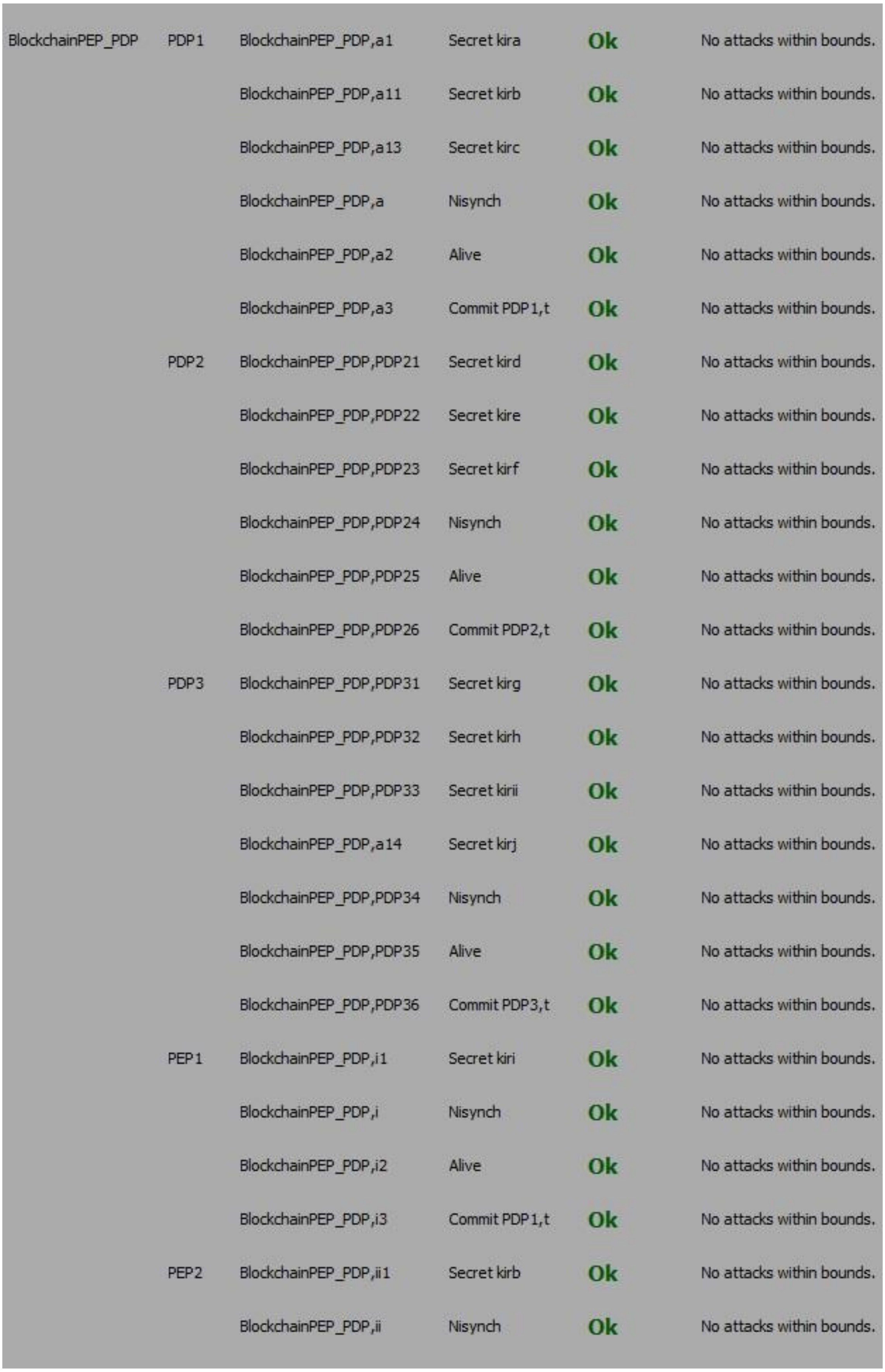


```
  PEP_PDP_Communication:   2
  PEP_PDP_Log file:        "std.log"
  index:                   2
  message:                 "Block mined!"
▾ previous_hash:           "cbe53caddf60308e63001ce309181ddd2d1d6263710a4a7b3a215364f8386714"
  proof:                   533
  timestamp:               "2022-04-23 22:27:15.979632"
```

Figure 5.6: BNM of MSS

The proposed architecture was tested and validated using scyther's formal method tool. It checks the proposed architecture against the adversary's capabilities as given in the dolev – yao model. It verifies the proposed methodology against required security protocols. For this, it verifies against claims: Alive, Nisynch, Secret, and Commitment.

It can be concluded from the results in Figure 5.7 proposed mechanism resists all the well-known primary attacks such as impersonation attacks, offline guessing attacks, denial of service attacks and replay attacks.



BlockchainPEP_PDP	PDP1	BlockchainPEP_PDP,a1	Secret kira	Ok	No attacks within bounds.
		BlockchainPEP_PDP,a11	Secret kirb	Ok	No attacks within bounds.
		BlockchainPEP_PDP,a13	Secret kirc	Ok	No attacks within bounds.
		BlockchainPEP_PDP,a	Nisynch	Ok	No attacks within bounds.
		BlockchainPEP_PDP,a2	Alive	Ok	No attacks within bounds.
		BlockchainPEP_PDP,a3	Commit PDP1,t	Ok	No attacks within bounds.
	PDP2	BlockchainPEP_PDP,PDP21	Secret kird	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP22	Secret kire	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP23	Secret kirf	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP24	Nisynch	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP25	Alive	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP26	Commit PDP2,t	Ok	No attacks within bounds.
	PDP3	BlockchainPEP_PDP,PDP31	Secret kirg	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP32	Secret kirh	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP33	Secret kirii	Ok	No attacks within bounds.
		BlockchainPEP_PDP,a14	Secret kirj	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP34	Nisynch	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP35	Alive	Ok	No attacks within bounds.
		BlockchainPEP_PDP,PDP36	Commit PDP3,t	Ok	No attacks within bounds.
	PEP1	BlockchainPEP_PDP,i1	Secret kiri	Ok	No attacks within bounds.
		BlockchainPEP_PDP,i	Nisynch	Ok	No attacks within bounds.
		BlockchainPEP_PDP,i2	Alive	Ok	No attacks within bounds.
		BlockchainPEP_PDP,i3	Commit PDP1,t	Ok	No attacks within bounds.
	PEP2	BlockchainPEP_PDP,ii1	Secret kirb	Ok	No attacks within bounds.
		BlockchainPEP_PDP,ii	Nisynch	Ok	No attacks within bounds.

Figure 5.7: Scyther claim results for MSS

5.4 SUMMARY

This chapter addresses the issue of data protection raised by various researchers in the literature. Although, some efforts have been made in the literature for improvements, yet they are not significantly effective. Insiders manage the PEP and PDP servers, so it is required to track their activity. This chapter complements the distributed architecture for better insider Authorisation control in the cloud environment with BS to store all the messages between PEP and PDP. Proposed work helps in achieving better data control and protection. The proposed protocol's working is verified and tested based on Scyther's four claims, proving that the proposed protocol is robust enough for real-time implementations. Its operational competence has also been tested in python by creating BN to store messages. The proposed solution attempts to solves the ITH issue at the architectural level by providing MSS in BS in the cloud environment.

CHAPTER-6

CONCLUSION AND FUTURE SCOPE

This thesis provides a CAC mechanism for ITH detection in a distributed cloud environment. This work aims to overcome the critical cyber security issue of ITH. This issue requires looking at two places, such as the local and architectural levels.

6.1 CONCLUSION

The issue of data protection is required to be handled at the architectural level. This thesis provides solutions in three chapters. Chapter three addresses the issue of ITH at the local level, where insiders with privileges try to remain hidden. They can delete or corrupt system logs, making identifying ITH difficult. Various techniques have been proposed in the literature to tackle this critical issue, but they remain non-effective as they store insider's activity analysis in system logs, and the insider is aware of them. This chapter proposes a blockchain-based robust technique for authorisation of log files of insiders in the cloud environment. Insider authentication and activity details are stored in the blockchain. Robustness, distributed ledger, immutability and other benefits do not allow insiders to change these system logs. The proposed solution significantly solves the ITH issue by providing access control to system logs. This technique is tested and validated using a scyther formal system tool. The result ascertains that the proposed system is highly significant and efficient and successfully mitigates various ITH. The working of the protocol is also verified based on the four claims, and scyther proved that the proposed protocol is robust enough for real-time implementations. Its operational competence has also been tested in python by creating BN for multiple users.

Chapter four addresses the issue of achieving better Authorisation control at the architecture level in the cloud environment. Work available in literature tried to handle this issue either at the PEP level or at the PDP level. Scalability remains the significant issue in these proposed architectures; performance deteriorate while achieving scalability. This chapter proposes a distributed architecture for improved insider Authorisation control in the cloud environment. It is based on multiple PEP–PDP servers and is successful in achieving significantly better results in scalability and performance. Results have also been validated statically in the

ANOVA test, which proves that the proposed system is highly efficient and successful compared to the existing multiple PEP single PDP architecture.

Chapter five addresses the issue of data protection as discussed in literature. However, some efforts are made in literature for improvements, but they are not significantly effective, Insiders manage the PEP and PDP servers, so it is a requirement to track their activity. This chapter complements the already proposed distributed architecture in chapter four for better insider Authorisation control in the cloud environment with BS to store all the messages between PEP and PDP. The proposed protocol's working is verified and tested based on the four claims, and scyther proved that the proposed protocol is robust enough for real-time implementations. Its operational competence has also been tested in python by creating BN to store messages. The proposed solution greatly solves the ITH issue and achieves better data control and protection at the architectural level by providing a MSS in BS in distributed architecture in the cloud environment.

6.2 FUTURE SCOPE

This thesis addresses the CAC mechanism for detecting ITH in the distributed cloud environment. In future, authorisation of log files of outsiders in BN and distributed PEP-PDP architecture for CAC for outsiders will be taken in the cloud environment. Furthermore, in future, improvement in inter-PDP communication, fault handling, failure of any PDP or PEP, Issue of cache coherence and Multi-threading will also be explored.

REFERENCES

[1] Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M., “A survey on security issues and solutions at different layers of Cloud computing,” The journal of supercomputing, vol.63 (2), pp. 561-592, 2013.

[2] Hsu, C. H., Ma, J., & Obaidat, M. S., “Dynamic intelligence towards merging cloud and communication services,” Information Systems Frontiers, vol.16 (1), pp.1-5, 2014.

[3] Flahive, A., Taniar, D., & Rahayu, W., “Ontology as a Service (OaaS): a case for sub-ontology merging on the cloud,” The Journal of Supercomputing, vol.65 (1), pp.185-216, 2013.

[4] A. K. Luhach, S. K. Dwivedi and C. K. Jha, "Applying SOA to an E-commerce system and designing a logical security framework for small and medium sized E-commerce based on SOA," 2014 IEEE International Conference on Computational Intelligence and Computing Research, 2014, pp. 1-6.

[5] Varghese, B., & Buyya, R., “Next generation cloud computing: New trends and research directions,” Future Generation Computer Systems, vol.79, pp.849-861, 2018.

[6] National Institute of Standards and Technology, “NIST Cloud Computing Standards Roadmap,” [Online].Available: https://www.nist.gov/system/files/documents/itl/cloud/ NIST_SP-500-291_Version-2_2013_June18_FINAL.pdf. [Accessed: Dec. 1, 2021].

[7] Sparks, A., “Databases in the cloud: how business communication is changing,” Journal of Database Marketing & Customer Strategy Management, vol.19 (2), pp.134-137, 2012.

[8] Mehak, F., Masood, R., Ghazi, Y., Shibli, M. A., & Khan, S., “Security aspects of database-as-a-service (DBaaS) in cloud computing,” Cloud Computing, Springer, Cham pp. 297-324, 2014.

[9] Rady, M., Abdelkader, T., & Ismail, R., “Integrity and confidentiality in cloud outsourced data,” Ain Shams Engineering Journal, vol. 10(2), pp. 275-285, 2019.

[10] Yaseen, Q., Althebyan, Q., Panda, B., & Jararweh, Y., “Mitigating insider threat in cloud relational databases,” Security and Communication Networks, vol. 9(10), pp. 1132-1145, 2016.

[11] Ramachandran, M., & Chang, V., “Towards performance evaluation of cloud service providers for cloud data security,” International Journal of Information Management, 36(4), 618-625, 2016.

[12] Barrowclough, J. P., & Asif, R., “Securing cloud hypervisors: a survey of the threats, vulnerabilities, and countermeasures,” Security and Communication Networks, vol. 2018, pp.1-20, 2018.

[13] Sandhu, R. S., & Samarati, P., “Access control: principle and practice,” IEEE communications magazine, vol. 32(9), pp. 40-48, 1994.

[14] Kizza, J., & Kizza, F. M., “Access control, authentication, and authorization,” IGI Global, Securing the Information Infrastructure, pp. 180-208, 2008.

[15] Yusop, Z. M., & Abawajy, J., “Analysis of insiders attack mitigation strategies,” Procedia-Social and Behavioral Sciences, vol.129, pp. 581-591, 2014.

[16] Duncan, A., Creese, S., & Goldsmith, M., “An overview of insider attacks in cloud computing,” Concurrency and Computation: Practice and Experience, vol.27 (12), pp. 2964-2981, 2015.

[17] Nucleus Cyber, “2019 Insider Threat Report,” [Online].Available: Https://Nucleuscyber.Com/Wp-Content/Uploads/2019/07/2019_Insider-Threat-Report_Nucleus_Final.Pdf. [Accessed: Dec. 1, 2021].

[18] Mlller, S., “2016 U.S. State of Cybercrime Highlights,” Jan. 19, 2017. [Online].Available: Https://Insights.Sei.Cmu.Edu/Blog/2016-Us-State-of-Cybercrime-Highlights/. [Accessed: Dec. 1, 2021].

[19] ObserveIT, “5 Examples of Insider Threat-Caused Breaches That Illustrate the Scope of the Problem,” March 22, 2018. [Online]. Available: Https://Www.Observeit.Com/Blog/5-Examples-of-Insider-Threat-Caused-Breaches/. [Accessed: Nov. 10, 2020].

[20] N.d,”5 Real-Life Examples of Breaches Caused by Insider,” [Online]. Available: https://www.ekransystem.com/en/blog/real-life-examples-insider-threat-caused-breaches. [Accessed: Sep. 11, 2021].

[21] Ko, L. L., Divakaran, D. M., Liau, Y. S., & Thing, V. L., “Insider threat detection and its future directions,” International Journal of Security and Networks, vol.12 (3), pp.168-187, 2017.

[22] Gheyas, I. A., & Abdallah, A. E., “Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis,” Big Data Analytics, vol.1 (1), pp.1-29, 2016.

[23] SOBERS, R. O. B., “2019 Data Risk Report Stats and Tips You Won’t Want to Miss,” [Online]. Available: Https://Www.Varonis.Com/Blog/Data-Risk-Report-Highlights-2019. [Accessed: Dec. 1, 2021].

[24] Van Tilborg, H. C., & Jajodia, S. (Eds.), “Encyclopedia of cryptography and security,” Springer Science & Business Media., 2014.

[25] Netsurion, “Logs for Insider Abuse Investigations,” June 23, 2021. [Online]. Available: Https://Www.Netsurion.Com/Articles/Logs-for-Insider-Abuse-Investigations. [Accessed: Dec. 10, 2021].

[26] El Sibai, R., Gemayel, N., Bou Abdo, J., & Demerjian, J., “A survey on access control mechanisms for cloud computing,” Transactions on Emerging Telecommunications Technologies, vol.31(2), e3720, 2020.

[27] Chen, H. C. J., Violetta, M. A., & Yang, C. Y., “Contract RBAC in cloud computing,” The Journal of Supercomputing, vol.66(2), pp.1111-1131, 2013.

[28] Xu, J., Yu, Y., Meng, Q., Wu, Q., & Zhou, F., “Role-Based Access Control Model for Cloud Storage Using Identity-Based Cryptosystem,” Mobile Networks and Applications, vol.26 (4), pp.1475-1492, 2020.

[29] Kerschbaum, F., & Robinson, P., “Security architecture for virtual organizations of business web services,” Journal of Systems Architecture, vol.55 (4), pp.224-232, 2009.

[30] Aziz, B., “Modelling fine-grained access control policies in grids,” Journal of Grid Computing, vol.14 (3), pp.477-493, 2016.

[31] Miniwatts Marketing Group, “Internet World Stats”, April 1st, 2021. [Online]. Available: https://www.internetworldstats.com/stats.htm. [Accessed: May 15, 2021].

[32] Roser, M., Ritchie, H., & Ortiz-Ospina, E., (2015). “Internet,” 2015. [Online]. Available: https://ourworldindata.org/internet. [Accessed: Dec. 12, 2021].

[33] Stalcup, K. A. T. Y., “AWS vs Azure vs Google Cloud Market Share 2021: What the Latest Data Shows,” May 17, 2021. [Online].Available: https: //www.parkmycloud.com/blog/aws-vs-azure-vs-google-cloud-market-share/. [Accessed: Nov. 1, 2021].

[34] Fernandes, D. A., Soares, L. F., & JV, G. MM, & Inácio, PR, “Security issues in cloud environments: a survey,” International Journal of Information Security, vol.13 (2), pp.113-170, 2014.

[35] Yaseen, Q., Althebyan, Q., Panda, B., & Jararweh, Y., “Mitigating insider threat in cloud relational databases,” Security and Communication Networks, vol.9 (10), pp.1132-1145, 2016.

[36] Iqbal, S., Kiah, M. L. M., Dhaghighi, B., Hussain, M., Khan, S., Khan, M. K., & Choo, K. K. R., “On cloud security attacks: A taxonomy and intrusion detection and prevention as a service,” Journal of Network and Computer Applications, vol.74, pp.98-120, 2016.

[37] Velásquez, I., Caro, A., & Rodríguez, A., “Authentication schemes and methods: A systematic literature review,” Information and Software Technology, vol.94, pp.30-37, 2018.

[38] Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., & Koucheryavy, Y., “Multi-factor authentication: A survey,” Cryptography, vol.2 (1), pp. 1, 2018.

[39] Ko, L. L., Divakaran, D. M., Liau, Y. S., & Thing, V. L., “Insider threat detection and its future directions,” International Journal of Security and Networks, vol.12 (3), pp.168-187, 2017.

[40] R. A. Alsowail and T. Al-Shehari, "Empirical Detection Techniques of Insider Threat Incidents," in IEEE Access, vol. 8, pp. 78385-78402, 2020.

[41] Harilal, A., Toffalini, F., Castellanos, J., Guarnizo, J., Homoliak, I., & Ochoa, M., “Twos: A dataset of malicious insider threat behavior based on a gamified competition,” In Proceedings of the 2017 International Workshop on Managing Insider Security Threats, 2017, pp. 45-56.

[42] Voris, J., Song, Y., Salem, M. B., Hershkop, S., & Stolfo, S., “Active authentication using file system decoys and user behavior modeling: results of a large scale study,” Computers & Security, vol.87, pp.101412, 2019.

[43] Maestre Vidal, J., & Sotelo Monge, M. A., “Obfuscation of malicious behaviors for thwarting masquerade detection systems based on locality features,” Sensors, vol.20 (7), pp.2084, 2020.

[44] Hu, T., Niu, W., Zhang, X., Liu, X., Lu, J., & Liu, Y., “An insider threat detection approach based on mouse dynamics and deep learning,” Security and communication networks, vol. 2019, pp.1-12, 2019.

[45] Legg, P. A., Moffat, N., Nurse, J. R., Happa, J., Agrafiotis, I., Goldsmith, M., & Creese, S., “Towards a conceptual model and reasoning structure for insider threat detection,” Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 4(4), pp.20-37, 2013.

[46] Agrafiotis, I., Legg, P. A., Goldsmith, M., & Creese, S., “Towards a User and Role-based Sequential Behavioural Analysis Tool for Insider Threat Detection,” J. Internet Serv. Inf. Secur., vol. 4(4), pp.127-137, 2014.

[47] Alsowail, R. A., & Al-Shehari, T., “A multi-tiered framework for insider threat prevention,” Electronics, vol.10 (9), pp.1005, 2021.

[48] Sticha, P. J., & Axelrad, E. T., “Using dynamic models to support inferences of insider threat risk,” Computational and Mathematical Organization Theory, vol.22 (3), pp.350-381, 2016.

[49] Tian, Z., Shi, W., Tan, Z., Qiu, J., Sun, Y., Jiang, F., & Liu, Y., “Deep learning and dempster-shafer theory based insider threat detection,” Mobile Networks and Applications, pp.1-10, 2020.

[50] Wu, Z., Xu, G., Lu, C., Chen, E., Jiang, F., & Li, G., “An effective approach for the protection of privacy text data in the CloudDB,” World Wide Web, vol.21 (4), pp.915-938, 2018.

[51] Moon, C. S., Chung, S., & Endicott-Popovsky, B., “A cloud and in-memory based two-tier architecture of a database protection system from insider attacks,” In International Workshop on Information Security Applications, Springer, Cham., 2013, pp. 260-271.

[52] Yaseen, Q., & Panda, B., “Predicting and preventing insider threat in relational database systems,” In IFIP International Workshop on Information Security Theory and Practices, Springer, Berlin, Heidelberg, 2010, pp. 368-383.

[53] Yaseen, Q., & Panda, B., “Insider threat mitigation: preventing unauthorized knowledge acquisition,” International Journal of Information Security, vol.11 (4), pp.269-280, 2012.

[54] Yaseen, Q., Jararweh, Y., Panda, B., & Althebyan, Q., “An insider threat aware access control for cloud relational databases,” Cluster Computing, vol.20 (3), pp.2669-2685, 2017.

[55] Dou, Z., Khalil, I., Khreishah, A., & Al-Fuqaha, A., “Robust insider attacks countermeasure for Hadoop: Design and implementation,” IEEE Systems Journal, vol.12 (2), pp.1874-1885, 2017.

[56] Shaghaghi, A., Kanhere, S. S., Kaafar, M. A., Bertino, E., & Jha, S., “Gargoyle: A network-based insider attack resilient framework for organizations,” In 2018 IEEE 43rd Conference on Local Computer Networks, IEEE, 2018, pp. 553-561.

[57] Chattopadhyay, P., Wang, L., & Tan, Y. P., “Scenario-based insider threat detection from cyber activities,” IEEE Transactions on Computational Social Systems, vol.5 (3), pp.660-675, 2018.

[58] Baracaldo, N., Palanisamy, B., & Joshi, J., “G-sir: an insider attack resilient geo-social access control framework,” IEEE Transactions on Dependable and Secure Computing, vol.16 (1), pp.84-98, 2017.

[59] Meng, W., Li, W., Wang, Y., & Au, M. H., “Detecting insider attacks in medical cyber–physical networks based on behavioral profiling,” Future Generation Computer Systems, vol.108, pp.1258-1266, 2020.

[60] Babu, B. M., & Bhanu, M. S., “Prevention of insider attacks by integrating behavior analysis with risk based access control model to protect cloud,” Procedia Computer Science, vol.54, pp.157-166, 2015.

[61] Eberz, S., Rasmussen, K. B., Lenders, V., & Martinovic, I., “Looks like eve: Exposing insider threats using eye movement biometrics,” ACM Transactions on Privacy and Security (TOPS), vol.19 (1), pp.1-31, 2016.

[62] Da Silva, C. E., Diniz, T., Cacho, N., & Lemos, R. D., “Self-adaptive authorisation in OpenStack cloud platform,” Journal of Internet Services and Applications, vol.9(1), pp.1-17,2018.

[63] Tsai, J. L., & Lo, N. W., “A privacy-aware authentication scheme for distributed mobile cloud computing services,” IEEE systems journal, vol.9 (3), pp.805-815, 2015.

[64] Kalra, S., & Sood, S. K., “Secure authentication scheme for IoT and cloud servers. Pervasive and Mobile Computing,” vol.24, pp.210-223, 2015.

[65] Amin, R., & Biswas, G. P., “A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis,” Journal of medical systems, vol.39 (3), pp.1-17, 2015.

[66] Yang, T. C., Lo, N. W., Liaw, H. T., & Wu, W. C., “A secure smart card authentication and authorization framework using in multimedia cloud,” Multimedia Tools and Applications, vol.76(9), pp.11715-11737, 2017.

[67] Kumari, S., Karuppiah, M., Das, A. K., Li, X., Wu, F., & Kumar, N., "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," The Journal of Supercomputing, vol.74 (12), pp.6428-6453, 2018.

[68] Shajina, A. R., & Varalakshmi, P., "A novel dual authentication protocol (DAP) for multi-owners in cloud computing," Cluster Computing, vol.20 (1), pp.507-523, 2017.

[69] Anakath, A. S., Rajakumar, S., & Ambika, S., "Privacy preserving multi factor authentication using trust management," Cluster Computing, vol.22 (5), pp.10817-10823, 2019.

[70] Chaudhry, S. A., Kim, I. L., Rho, S., Farash, M. S., & Shon, T., "An improved anonymous authentication scheme for distributed mobile cloud computing services," Cluster Computing, vol.22 (1), pp.1595-1609, 2019.

[71] Kumar, S., Singh, S. K., Singh, A. K., Tiwari, S., & Singh, R. S., "Privacy preserving security using biometrics in cloud computing," Multimedia Tools and Applications, vol.77 (9), pp.11017-11039, 2018.

[72] Chatterjee, K., "Biometric re-authentication: An approach towards achieving transparency in user authentication," Multimedia Tools and Applications, vol.78 (6), pp. 6679-6700, 2019.

[73] Deebak, B. D., & Al-Turjman, F., "Secure-user sign-in authentication for IoT-based eHealth systems,". Complex & Intelligent Systems, pp.1-21,2021.

[74] Abomhara, M., Yang, H., Køien, G. M., & Lazreg, M. B., "Work-based access control model for cooperative healthcare environments: Formal specification and verification," Journal of Healthcare Informatics Research, vol.1 (1), pp.19-51, 2017.

[75] Alam, M., Emmanuel, N., Khan, T., Xiang, Y., & Hassan, H., "Garbled role-based access control in the cloud," Journal of Ambient Intelligence and Humanized Computing, vol.9 (4), pp.1153-1166, 2018.

[76] Habiba, M., Islam, M., & Ali, A. B. M., "A new approach to access control in cloud," Arabian Journal for Science and Engineering, vol.41 (3), pp.1015-1030, 2016.

[77] Jo, S. M., "Secure access policy for efficient resource in mobile computing environment," Journal of Computer Virology and Hacking Techniques, vol.13 (4), pp.297-303, 2017.

[78] Chen, F., Luo, Y., Zhang, J., Zhu, J., Zhang, Z., Zhao, C., & Wang, T., "An infrastructure framework for privacy protection of community medical internet of things," World Wide Web, vol.21 (1), pp.33-57, 2018.

[79] Shin, S., & Kwon, T., “AAnA: Anonymous authentication and authorization based on short traceable signatures,” International journal of information security, vol.13 (5), pp.477-495, 2014.

[80] Gabillon, A., Gallier, R., & Bruno, E., “Access controls for iot networks,” SN Computer Science, vol.1 (1), pp.1-13, 2020.

[81] Rathore, N. C., & Tripathy, S., “A trust-based collaborative access control model with policy aggregation for online social networks,” Social Network Analysis and Mining, vol.7 (1), pp.1-13, 2017.

[82] Nyrkov, A., Romanova, Y., Ianiushkin, K., & Li, I.,” Data Processing Model in Hierarchical Multi-agent System Based on Decentralized Attribute-Based Encryption,” In Energy Management of Municipal Transportation Facilities and Transport, Springer, Cham, 2018, pp. 429-438.

[83] Son, H. X., Nguyen, M. H., & Vo, H. K.,” Toward an privacy protection based on access control model in hybrid cloud for healthcare systems,” In International Joint Conference: 12th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2019) and 10th International Conference on EUropean Transnational Education (ICEUTE 2019), Springer, Cham., 2019 , pp. 77-86.

[84] Cao, Y., Huang, Z., Yu, Y., Ke, C., & Wang, Z., “A topology and risk-aware access control framework for cyber-physical space,” Frontiers of Computer Science, vol.14 (4), pp.1-16, 2020.

[85] Ryan, D., De Leon, M. P., Grant, N., Butler, B., Vogel, S., Mirz, M., & Lyons, P., “Deriving policies from connection codes to ensure ongoing voltage stability,” Energy Informatics, vol.2 (1), pp.1-14, 2019.

[86] Elmisery, A. M., Rho, S., & Aborizka, M., “A new computing environment for collective privacy protection from constrained healthcare devices to IoT cloud services,” Cluster Computing, vol.22 (1), pp.1611-1638, 2019.

[87] Oglaza, A., Laborde, R., Zaraté, P., Benzekri, A., & Barrère, F., “A new approach for managing Android permissions: learning users’ preferences,” EURASIP Journal on Information Security, vol.2017 (1), pp.1-16, 2017.

[88] Krempel, E., Birnstill, P., & Beyerer, J., “A privacy-aware fall detection system for hospitals and nursing facilities,” European Journal for Security Research, vol.2 (2), pp.83-95, 2017.

[89] Baldini, G., Botterman, M., Neisse, R., & Tallacchini, M., "Ethical design in the internet of things," Science and engineering ethics, vol.24 (3), pp.905-925, 2018.

[90] Ghazi, Y., Masood, R., Rauf, A., Shibli, M. A., & Hassan, O., "DB-SECaaS: a cloud-based protection system for document-oriented NoSQL databases," EURASIP Journal on Information Security, vol.2016 (1), pp.1-17, 2016.

[91] Mehak, F., Masood, R., Shibli, M. A., & Elgedway, I., "EACF: extensible access control framework for cloud environments," Annals of Telecommunications, vol.72 (5), pp. 307-323, 2017.

[92] Kovacs, A., "Cybersecurity and Data Protection Regulation in India: An Uneven Patchwork," Springer, Cham, CyberBRICS, pp. 133-181, 2021.

[93] S, Akshaya, "An Analysis of the Data Protection Laws in India," June 26, 2020. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3616637. [Accessed: May 15, 2021].

[94] Bhadade, I. P. P., Chandak, I. P. S., & Tolani, K., "The impact of European Union's General Data Protection Regulation on Indian Data Privacy Laws," Journal for the Study of Research,vol. 12(5), pp. 7-11, 2020.

[95] Gada, D., "A STUDY OF THE LAW REFORMS NEEDED IN THE DATA PROTECTION LAW OF INDIA," International Journal of Modern Agriculture, vol.10 (2), pp.1417-1424, 2021.

[96] Srivastava, A. K., "Data Protection Law in India: The Search for Goldilocks Effect," Eur. Data Prot. L. Rev., vol. 5, pp.408, 2019.

[97] Chen, G., Xu, B., Lu, M., & Chen, N. S., "Exploring blockchain technology and its potential applications for education," Smart Learning Environments, vol.5 (1), pp.1-10, 2018.

[98] Li, H., Pei, L., Liao, D., Sun, G., & Xu, D., "Blockchain meets VANET: An architecture for identity and location privacy protection in VANET," Peer-to-Peer Networking and Applications, vol.12 (5), pp.1178-1193, 2019.

[99] Chung, K., Yoo, H., Choe, D., & Jung, H., "Blockchain network based topic mining process for cognitive manufacturing," Wireless Personal Communications, vol.105 (2), pp.583-597, 2019.

[100] Sun, J., Yan, J., & Zhang, K. Z., "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," Financial Innovation, vol.2 (1), pp.1-9, 2016.

[101] Vinod, B., "Blockchain in travel," Journal of Revenue and Pricing Management, vol.19 (1), pp. 2-6, 2020.

[102] Han, H., Huang, M., Zhang, Y., & Bhatti, U. A., "An architecture of secure health information storage system based on blockchain technology," In International conference on cloud computing and security, Springer, Cham, 2018, pp. 578-588.

[103] Ryu, J. H., Sharma, P. K., Jo, J. H., & Park, J. H., "A blockchain-based decentralized efficient investigation framework for IoT digital forensics," The Journal of Supercomputing, vol.75(8), pp.4372-4387, 2019.

[104] S. Gómez Sáez, V. Andrikopoulos, F. Leymann and S. Strauch, "Design Support for Performance Aware Dynamic Application (Re-)Distribution in the Cloud," in IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 225-239, 2015.

[105] P. Malhotra, Y. Singh, P. Anand, D. K. Bangotra, P. K. Singh, and W.-C. Hong, "Internet of Things: Evolution, Concerns and Security Challenges," *Sensors*, vol. 21, no. 5, p. 1809, Mar. 2021,

[106] L. Luo, Y. Zhang, W. Ma and Y. Zhuang, "System for Land Surface Model Applications Based on Cloud Computing," in *IEEE Access*, vol. 5, pp. 12041-12048, 2017.

[107] Sigelman, O., "Personal cloud-based apps: the new insider risk," *Computer Fraud & Security*, vol. *2017*(8), pp. 10-12, 2017.

[108] Hölbl, M., Kompara, M., Kamišalić, A., & Nemec Zlatolas, L., "A systematic review of the use of blockchain in healthcare," Symmetry, vol.10 (10), pp.470, 2018.

[109] Bhatia, R., "Interoperability solutions for blockchain," In 2020 international conference on smart technologies in computing, electrical and electronics, IEEE,2020, pp. 381-385.

[110] Novak, M., "Crypto-friendliness: Understanding blockchain public policy," Journal of Entrepreneurship and Public Policy, 2019.

[111] Raddatz, N., Coyne, J., Menard, P., & Crossler, R. E., "Becoming a blockchain user: understanding consumers' benefits realisation to use blockchain-based applications," European Journal of Information Systems, pp.1-28, 2021.

[112] Gorkhali, A., Li, L., & Shrestha, A., "Blockchain: A literature review," Journal of Management Analytics, vol.7 (3), pp.321-343, 2020.

[113] Zhang, R., Xue, R., & Liu, L., "Security and privacy on blockchain," ACM Computing Surveys (CSUR), vol.52 (3), pp.1-34, 2019.

[114] Herzog, J., "A computational interpretation of Dolev–Yao adversaries," Theoretical Computer Science, vol.340 (1), pp.57-81, 2005.

[115] Backes, M., Pfitzmann, B., & Waidner, M., “Symmetric authentication in a simulatable Dolev–Yao-style cryptographic library,” International Journal of Information Security, vol.4 (3), pp.135-154, 2005.

[116] Cremers, C. J.,”The Scyther Tool: Verification, falsification, and analysis of security protocols,” In International conference on computer aided verification, Springer, Berlin, Heidelberg, 2008, pp. 414-418.

[117] Yang, H., Oleshchuk, V. A., & Prinz, A., “Verifying Group Authentication Protocols by Scyther,” J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., vol. 7(2), pp.3-19, 2016.

[118] Cremers, C. J. F.,”Scyther: Semantics and verification of security protocols,” Eindhoven, Netherlands: Eindhoven university of Technology, 2006.

[119] Cremers, C. J. F., “Scyther. Semantics and Verification of Security Protocols, Thesis,” University Press Eindhoven, 2006.

[120] Cremers, C. J. F.,” Scyther: Unbounded verification of security protocols,” Technical Report/ETH Zurich, Department of Computer Science, 572, 2011.

[121] Amin, R., Lohani, P., Ekka, M., Chourasia, S., & Vollala, S., “An enhanced anonymity resilience security protocol for vehicular ad-hoc network with Scyther simulation,” Computers & Electrical Engineering, vol.82, pp.106554, 2020.

[122] Singh, S., & Bawa, S., “A privacy, trust and policy based authorization framework for services in distributed environments,” International Journal of Computer Science, vol.2 (2), pp.85-92, 2007.

[123] Kaur, H., Kumar, N., & Batra, S., “ClaMPP: A cloud-based multi-party privacy preserving classification scheme for distributed applications,” The Journal of Supercomputing, vol.75 (6), pp.3046-3075, 2019.

[124] Cuevas, A., Febrero, M., & Fraiman, R., “An anova test for functional data,” Computational statistics & data analysis, vol.47 (1), pp.111-122, 2004.

[125] Kim, T. K., “Understanding one-way ANOVA using conceptual figures,” Korean journal of anesthesiology, vol.70 (1), pp. 22-26, 2017.

[126] Górecki, T., & Smaga, Ł., “A comparison of tests for the one-way ANOVA problem for functional data,” Computational Statistics, vol. 30(4), pp.987-1010, 2015.

[127] Deng, F., Lu, J., Wang, S. Y., Pan, J., & Zhang, L. Y., “A distributed PDP model based on spectral clustering for improving evaluation performance,” World Wide Web, vol.22(4), pp.1555-1576, 2019.

[128] de Carvalho, R. M., Del Prete, C., Martin, Y. S., Araujo Rivero, R. M., Önen, M., Schiavo, F. P., ... & Koukovini, M. N., “Protecting citizens’ personal data and privacy: Joint effort from GDPR EU cluster research projects,” SN Computer Science, vol.1(4), pp.1-16, 2020.

[129] Q. Wang, C. Wang, K. Ren, W. Lou and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847-859, 2011.

[130] Singh,H., Robertson,D.,Hasedzic, E.,Pickering,C.,& Skrypchuk,L.,” Communication system and related method,” U. S. Patent, 15304962, 6th July 2017.

[131] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman and D. Woods, "Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds," in *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 523-536, 2017.

[132] V. Chang and M. Ramachandran, "Towards Achieving Data Security with the Cloud Computing Adoption Framework," in *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 138-151, 2016.

[133] X. Luo, M. Dong and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," in *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 58-70, Jan. 2006

[134] Hoofnagle, C. J., van der Sloot, B., & Borgesius, F. Z., “The European Union general data protection regulation: what it is and what it means,” Information & Communications Technology Law, vol.28 (1), pp. 65-98, 2019.

[135] Ujcich, B. E., Bates, A., & Sanders, W. H.,” A provenance model for the European Union general data protection regulation,” In International Provenance and Annotation Workshop, Springer, Cham., 2018, pp. 45-57.

[136] Voigt, P., & Von dem Bussche, A., “The eu general data protection regulation (gdpr). A Practical Guide,” 1st Ed., Cham: Springer International Publishing, vol.10 (3152676), pp. 10-5555, 2017.

[137] Tikkinen-Piri, C., Rohunen, A., & Markkula, J., “EU General Data Protection Regulation: Changes and implications for personal data collecting companies,” Computer Law & Security Review, vol.34 (1), pp. 134-153, 2018.

[138] Kędzior, M., “GDPR and beyond—a year of changes in the data protection landscape of the European Union,” Springer Berlin Heidelberg, In era Forum vol. 19(4), pp. 505-509, 2019.

[139] M. J. Alhanahnah, A. Jhumka and S. Alouneh, "A Multidimension Taxonomy of Insider Threats in Cloud Computing," in *The Computer Journal*, vol. 59, no. 11, pp. 1612-1622, 2016.

[140] Singh, A., & Verma, M., “Attacks and security in cloud computing,” International Journal of Advanced Engineering & Application, vol.1, pp.300-302, 2011.

[141] Yusop, Z. M., & Abawajy, J., “Analysis of insiders attack mitigation strategies,” *Procedia-Social and Behavioral Sciences*, vol.*129*, pp. 581-591, 2014.

[142] D. Singh and H. K. Verma, "A new framework for cloud storage confidentiality to ensure information security," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), 2016, pp. 1-5.

[143] A. Y. Al Hammadi *et al*., "Novel EEG Sensor-Based Risk Framework for the Detection of Insider Threats in Safety Critical Industrial Infrastructure," in *IEEE Access*, vol. 8, pp. 206222-206234, 2020.

[144] M. Ahmadian and D. C. Marinescu, "Information Leakage in Cloud Data Warehouses," in *IEEE Transactions on Sustainable Computing*, vol. 5, no. 2, pp. 192-203, 2020.

[145] L. Liu, O. De Vel, Q. -L. Han, J. Zhang and Y. Xiang, "Detecting and Preventing Cyber Insider Threats: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1397-1417, Secondquarter 2018.

PUBLICATIONS

JOURNALS

1. Deep, G., Sidhu, J., & Mohana, R., “Asymmetric authentication as a service in private cloud, formal validation by scyther”, IJAST, vol. 28, no. 19, pp. 710-717, 2019**. (Published, Scopus)**

2. Deep, G., Sidhu, J., & Mohana, R., ”Access management of user and cyber-physical device in DBaaS according to Indian IT laws using Blockchain”, Scalable Computing, vol. 21, no 3, pp. 407-424, 2020. (**Published, ESCI, Scopus)**

3. Deep, G., Sidhu, J., & Mohana, R., “Insider Threat Prevention Mechanism for DBaaS Cloud environment”, Computers & Industrial Engineering, vol. 169, pp. 108278,2022. **(Published, SCI) (Impact factor: 7.18)**

4. Deep, G., Sidhu, J., & Mohana, R., “Distributed PEP-PDP Model for Insider Threat Aware Access Control for Cloud Relational Databases”, Wireless Personal Communications , vol.128, pp. 1733–1761,2023. **(Published, SCI) (Impact factor: 2.017)**

COMMUNICATED

1. Deep, G., Sidhu, J., & Mohana, R., “Insider threat detection in distributed PEP-PDP cloud architecture”, Journal of Computer Science and Technology **(SCI, Communicated)**

CONFERENCES

1. Deep, G., Sidhu, J., & Mohana, R., “Role of Indian IT Laws in Smart Healthcare Devices in the Intensive Care Unit in India” presented at 6th International Conference on Parallel, Distributed and Grid Computing , JUIT, Solan, India, Nov. 06-08, 2020. **(Published, Scopus)**

2. Deep, G., Sidhu, J., & Mohana, R., “Investigation and Validation of Distributed PEP-PDP Authorisation Architecture in Insider Access Control Mechanism in Scyther” presented at 3rd International Conference on Innovations in communication computing and Sciences, CGC, Landran, Mohali, India, Aug. 27-28, 2021. **(Published, Scopus)**