

# A Novel Multi-Factor Authentication Approach for Multi-Cloud Computing Systems

THESIS

Submitted in partial fulfillment of the  
requirements for the degree of

**DOCTOR OF PHILOSOPHY**

By

**SOUMYA PRAKASH OTTA**

**ID. No. 2016PHXF0300H**

Under the Supervision of

**Prof. Subhrakanta Panda**

Co-Supervision of

**Prof. Chittaranjan Hota**



## **BITS Pilani**

Pilani | Dubai | Goa | Hyderabad, | Mumbai (An Institution of Eminence)

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

2023

# Certificate

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

This is to certify that the thesis entitled **A Novel Multi-Factor Authentication Approach for Multi-Cloud Computing Systems** and submitted by **SOUMYA PRAKASH OTTA** ID.NO. **2016PHXF0300H** for the award of Ph.D. of the institute embodies original work done by him under my supervision.

.....

Signature of the Supervisor

Prof. SUBHRAKANTA PANDA

Associate Professor

Date:

.....

Signature of the Co-Supervisor

Prof. CHITTARANJAN HOTA

Senior Professor

Date:

# Dedication

To Maa and Bapa...<sup>††</sup>

---

<sup>††</sup>To My Parents

# Acknowledgement

I would like to thank all the amazing people who contributed in some way to the work described in this thesis. First and foremost, I would like to express my profound gratitude & sincere thanks to my Supervisor, Prof. Subhrakanta Panda, whose expertise, understanding, and patience added values at every stage of this thesis work.

I would also like to thank my Co-Supervisor, Prof. Chittaranjan Hota, for his generous advice and encouragement. I am thankful to my DAC members, Prof. G Geethakumari and Dr. Jabez Christopher, for their encouragement, insightful comments, and hard questions. I also express my gratitude to Prof. N.L.Bhanu Murthy, HOD of CSIS Dept and Prof. Tathagata Ray, DRC Convener, Dept of CSIS, along with technical support staff of the Dept for their uninterrupted mentoring and support in all stages of the thesis work.

I express my gratitude towards Prof. G Sundar (Director and Senior Professor of BITS Pilani Hyderabad Campus) and Prof. P. Yogeeswari (Dean, Administration). I would also like to express my gratitude to Prof. Venkata Vamsi Krishna Venuganti (Associate Dean, Academic Research Division) for constant support during my Ph.D. course.

I would like to thank the co-authors of my papers Dr. Maanak Gupta, Assistant Professor, Department of Computer Science, Tennessee Technological University, and First Degree & Higher Degree students of BITS Pilani Sidhharth, Anuj, Sankalp, and Khushi. I am thankful to fellow Ph.D. scholars BSA Rajita and Deepa Kumari for their continuous support.

I express my gratitude to my parents Major P K Otta & Late Smt S Otta, my wife, LtCol Sonali, and son Mr. Sambhav Otta for their constant love, support, and unfailing guidance. I owe them everything.

# Abstract

With the advancement of technology, smart and net-enabled devices have started replacing human users, leading to the existence of the *Cloud of Things*. Cloud Service Providers have already collaborated with other providers to have a *Multi-cloud* infrastructure. Malicious users on the dark web always try to exploit the vulnerability of web-enabled cloudified applications, putting sensitive information at risk. The research community has realized that it's the user of the cloud system who needs to be proactive in safeguarding the user identity profile and system resources of the cloud. Whether the user may be human or a device, result-oriented research is needed to enhance user identity security for *Security As A Service*. For secure authentication of genuine users and seamless access to the entitled resources of the cloud with effective access control, a robust *Identity and Access Management* function is essential. The first step is to have secure user authentication for the multi-cloud. *Multi-factor Authentication* is regarded as a means to enhance difficulty for the attacker, and when coupled with Cryptographic means with advanced web access facility, it can show exponential results. This research aims to achieve secure authentication with the application of *Self-Sovereign Identity*, with no special hardware or software requirement applicable for users and devices. The *Blockchain enabled Multi-factor Authentication* yields the advantages of inherent *Public Key Infrastructure* and *Secure Hash Algorithm*. Similarly, the *Continuous Multi-factor Authentication* and *Dynamic Multi-factor Authentication* provide effective results to deal with *Spoofing Attack* and *Impersonation Attack* on users and devices of the multi-cloud environment. Effective textitTrust on the cloud and Trust of the Cloud for users has been a game changer with the application of *Zero Trust Network* approach and suitable access control policy enforcement with *Software Defined Perimeter* applied in conjunction with a Zero Trust based Cloud Network. The same is experimented to find encouraging results, opening other challenging options.

# Contents

<b>Certificate</b>	<b>i</b>
<b>Abstract</b>	<b>iv</b>
<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Algorithms</b>	<b>xiv</b>
<b>List of Abbreviations</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Cloud Computing . . . . .	2
1.1.1 Service, Deployment Models and Characteristics of Cloud Computing	2
1.1.2 Cloud Computing Security Prospective . . . . .	6
1.1.3 Criticality of User Identity and Access Management in Cloud . . . . .	8
1.2 Motivation . . . . .	9
1.3 Research Gaps . . . . .	10
1.4 Objectives . . . . .	12
1.5 Contributions of the Thesis . . . . .	14
1.6 Thesis Organization . . . . .	15
1.7 Chapter Summary . . . . .	16
<b>2 Literature Survey &amp; Related Research</b>	<b>18</b>
2.1 Concerns in Cloud Security . . . . .	18
2.1.1 IaaS Security . . . . .	18
2.1.2 PaaS Security . . . . .	19

2.1.3	SaaS Security . . . . .	19
2.1.4	SECaaS Security . . . . .	19
2.2	Cloud Vulnerability & Priotization . . . . .	20
2.2.1	Cloud Vulnerability . . . . .	20
2.2.2	Cloud Vulnerability Priotization . . . . .	22
2.3	Cloud Authentication . . . . .	22
2.3.1	Cloud User Authentication . . . . .	24
2.3.1.1	Characteristics for Cloud Authentication . . . . .	24
2.3.1.2	Threats Towards Cloud Authentication . . . . .	25
2.3.2	Advanced Authentication Approaches . . . . .	31
2.4	Chapter Summary . . . . .	33
<b>3</b>	<b>Factors of Authentication &amp; Application to Cloud Based Systems</b>	<b>34</b>
3.1	Factors of Authentication . . . . .	34
3.1.1	Additional Layer for Authentication . . . . .	35
3.1.2	Emergence of MFA . . . . .	36
3.1.2.1	Obstacles & Threats to MFA . . . . .	37
3.1.3	MFA Algorithms & Analysis . . . . .	38
3.1.4	Applicability of Factors for MFA . . . . .	40
3.1.4.1	Applicability of MFA for Users . . . . .	42
3.1.4.2	Applicability of MFA for Devices . . . . .	42
3.2	Uniqueness of Users and Devices . . . . .	42
3.2.1	Privacy of Users and Devices in Cloud . . . . .	44
3.3	Chapter Summary . . . . .	44
<b>4</b>	<b>Face Recognition for User Authentication of Cloud Based Systems</b>	<b>45</b>
4.1	Analysis of Facial Recognition . . . . .	45
4.1.1	Face Detection . . . . .	46
4.1.2	CNN Based Face Recognition . . . . .	51
4.1.3	Modular Approach for Face Recognition Process . . . . .	51
4.1.4	Liveness Check & Anti Spoofing Check . . . . .	53
4.1.4.1	Detailed Analysis . . . . .	54
4.2	Chapter Summary . . . . .	55

---

<b>5</b>	<b>Secure Authentication of Users In Multi-Cloud Environment</b>	<b>57</b>
5.1	Introduction . . . . .	57
5.2	Background Literature Study . . . . .	59
5.2.1	Traditional Approach . . . . .	60
5.2.2	Self-Sovereign Approach . . . . .	61
5.2.2.1	Sovrin Netwoking . . . . .	62
5.2.3	Incorporation of MFA to SSI . . . . .	65
5.2.4	Factors for Re-Authentication . . . . .	65
5.3	Research Gaps and Questions . . . . .	67
5.4	Proposed Approach of Multi-Factor Authentication . . . . .	68
5.4.1	Use of Biometric . . . . .	69
5.4.2	Dynamic MFA (D-MFA) . . . . .	70
5.4.3	Continous MFA (C-MFA) . . . . .	71
5.4.4	Selection & Implementation of Blockchain . . . . .	73
5.4.5	Interfacing with DL for MFA System . . . . .	73
5.4.6	Implementation & Integration of Multiple Smart-Contract . . . . .	74
5.4.7	Security Requirements & Design Goals . . . . .	75
5.5	Implementation and Evaluation . . . . .	77
5.5.1	Experimental Setup . . . . .	80
5.5.2	System Implementation . . . . .	81
5.5.3	System Model . . . . .	82
5.5.3.1	Multi-Cloud Testbed & Distributed Database Mapping . . . . .	83
5.5.3.2	User Interface . . . . .	83
5.5.4	System Threat Analysis . . . . .	83
5.5.5	System Evaluation . . . . .	86
5.5.6	Security Scanning using ZAP . . . . .	90
5.5.7	API Response Time Patterns . . . . .	91
5.6	Chapter Summary . . . . .	94
<b>6</b>	<b>Secure Authentication of Devices In Multi-cloud Environment</b>	<b>96</b>
6.1	Introduction . . . . .	96
6.1.1	IoT Authentication . . . . .	98
6.1.1.1	Authentication with One Factor . . . . .	98



6.1.1.2	Authentication with Multiple-Factors . . . . .	98
6.2	Background Literature Study . . . . .	100
6.2.1	IoT Environment & Security Requirements . . . . .	102
6.2.1.1	IoT Environment . . . . .	102
6.2.1.2	Security Essentials . . . . .	103
6.3	Research Gaps and Questions . . . . .	106
6.3.1	Cryptographic Infrastructure . . . . .	106
6.3.2	Device Biometric Authentication . . . . .	107
6.3.3	Storage of Credentials . . . . .	108
6.4	Proposed Approach for Multi-Factor Authentication . . . . .	109
6.4.1	System Model . . . . .	114
6.4.2	Security Analysis . . . . .	114
6.5	Implementation and Evaluation . . . . .	117
6.5.1	Deployment of the Environment . . . . .	118
6.5.1.1	Device Registration . . . . .	119
6.5.1.2	Device Login . . . . .	120
6.5.1.3	Device to Device Communication . . . . .	122
6.5.1.4	Cryptographic Key . . . . .	122
6.5.1.5	Blockchain Mechanism for Data Integrity Verification . . .	123
6.5.2	Threat Modeling Performance Analysis . . . . .	124
6.5.2.1	Threat Modeling Analysis . . . . .	124
6.5.2.2	Liveness Check . . . . .	126
6.5.2.3	Performance Analysis . . . . .	127
6.6	Chapter Summary . . . . .	128
<b>7</b>	<b>Augmented Multi-Factor Authentication</b>	<b>131</b>
7.1	Introduction . . . . .	131
7.2	Background Literature Study . . . . .	132
7.2.1	Augmented Multi-Factor Authentication . . . . .	132
7.2.2	Zero Trust Architecture . . . . .	132
7.2.3	Zero Trust Network . . . . .	133
7.2.4	Software Defined Perimeter . . . . .	135

7.3	Research Gaps and Questions . . . . .	138
7.3.1	Research on ZTN and SDP . . . . .	140
7.4	Proposed Approach . . . . .	143
7.4.1	MFA in ZTN . . . . .	143
7.4.1.1	Design and Development of ZTN Architecture . . . . .	143
7.4.1.2	MFA for User & Device in ZTN . . . . .	145
7.4.1.3	Device to Device Lightweight Authentication . . . . .	146
7.4.1.4	Integration for A-MFA . . . . .	147
7.4.2	Design and Development of SDP Architecture . . . . .	148
7.4.2.1	Vulnerability for SDP Controller . . . . .	148
7.4.2.2	SDP Load Balancer . . . . .	149
7.5	Implementation and Evaluation . . . . .	150
7.5.1	Implementation . . . . .	150
7.5.1.1	Security Analysis . . . . .	150
7.5.2	Performance Evaluation . . . . .	151
7.6	Chapter Summary . . . . .	152
<b>8</b>	<b>Conclusions</b>	<b>154</b>
8.1	Contributions . . . . .	154
8.1.1	Findings With Respect to Users . . . . .	155
8.1.2	Findings With Respect to Devices . . . . .	155
8.2	Lessons Learnt . . . . .	156
8.3	Future Work . . . . .	157
	<b>Bibliography</b>	<b>160</b>
	<b>List of Publications</b>	<b>178</b>
	<b>Biographies</b>	<b>180</b>

## List of Tables

2.1	Characteristics for Cloud Authentication . . . . .	26
2.2	Threats to Cloud Authentication and Suggested Remedial Measures . . . . .	30
3.1	Prominent Obstacles for MFA Implementation . . . . .	37
3.2	Prominent Threats of MFA Implementation . . . . .	38
3.3	Comparative Analysis of MFA Algorithms . . . . .	40
3.4	Factors for MFA Implementation of Users . . . . .	41
3.5	Factors for MFA Implementation of Devices . . . . .	43
4.1	Comparative Analysis of Face Detection Models . . . . .	54
4.2	Comparative Analysis of Face Recognition CNN Models . . . . .	55
5.1	Limitations of Traditional Authentication Approaches . . . . .	61
5.2	Advantages of SSI Authentication Approach . . . . .	63
5.3	Suitability of Factors for Re-Authention . . . . .	66
5.4	Comparative Analysis of Optimised Face Verification Models . . . . .	73
5.5	Design Goals of Blockchain-Based MFA Solution . . . . .	76
6.1	IoT Security Factors . . . . .	97
6.2	Comparision of Communication Protocols for Authentication & Authoroza- tion . . . . .	101
6.3	Comprative Analysis of Research Works on PKI-Based Blockchain . . . . .	104
6.4	Comparison of Lightweight Hash Functions . . . . .	106
6.5	Notations Used for Lightweight Secure Authentication of Devices . . . . .	109
7.1	Prominent Observations on ZTN & SDP Research . . . . .	142

7.2 Comparison of A-MFA With Similar Functioning Identity Authentication  
Methods . . . . . 151

# List of Figures

1.1	System Administration and Control of Various Clouds Services . . . . .	4
1.2	Organization of Thesis . . . . .	17
2.1	Contributers Towards Cloud Vulverability . . . . .	23
2.2	Threats towards Cloud Authentication . . . . .	27
3.1	Factors of Authentication . . . . .	35
5.1	Functioning of SSI Ecosystem . . . . .	63
5.2	Hyperledger on Cloud for SSI Ecosystem . . . . .	64
5.3	System Architecture of D-MFA . . . . .	71
5.4	System Architecture of C-MFA . . . . .	72
5.5	Integration of Multiple Smartcontract Deployed over Ethereum . . . . .	75
5.6	System Model of D-MFA & C-MFA Over Multi-Cloud Testbed with Dis- tributed Database Mapping . . . . .	84
5.7	Login interface for Face Recognition using D-MFA as well as C-MFA . . . . .	85
5.8	On-premice Authentication time Performance Comparision . . . . .	87
5.9	On-premice Authentication Processor Utilization Comparision . . . . .	87
5.10	Processor Utilization with Azure & AWS Based Cloud Server for MFA . . . . .	88
5.11	Authentication Time Over Azure & AWS Based Cloud Server for SFA & MFA . . . . .	89
5.12	SPAN Output Screenshot of AVIPSA Security Analysis Tool . . . . .	90
5.13	ZAP Test Report of D-MFA & C-MFA with Multi-Cloud Testbed . . . . .	91
5.14	User Registration Response Time & Face Registration Response Time for D-MFA & C-MFA with Multi-Cloud Testbed . . . . .	92
5.15	User Sign-in Response Time & Face Verification Response Time for D-MFA & C-MFA with Multi-Cloud Testbed . . . . .	92

5.16 Processor Utilization and Latency for D-MFA & C-MFA with Multi-Cloud Testbed . . . . .	93
6.1 Lightweight Secure Authentication for Devices of Cloud . . . . .	110
6.2 System Architecture of Blockchain-Enabled Device Authentication . . . . .	115
6.3 Structure of the Blockchain & Block Configuration . . . . .	118
6.4 Device Registration Process . . . . .	120
6.5 Test Cases for Liveness Check of IoT Devices . . . . .	127
6.6 Key Generation and Key Validation Performance Analysis . . . . .	128
6.7 Authentication-Time Performance Analysis . . . . .	129
6.8 Time Interval for Test Cases of Liveness Check . . . . .	129
7.1 Zero Trust Network Components . . . . .	135
7.2 Software Defined Perimeter . . . . .	137
7.3 Design of MFA over ZTN . . . . .	145
7.4 Design of Dynamic MFA over ZTN . . . . .	146
7.5 Intigration for A-MFA . . . . .	148
7.6 Load Balancer for SDP Controller . . . . .	149
7.7 Testbed for A-MFA . . . . .	150
7.8 Processor Utilization for A-MFA & Comparision to C-MFA . . . . .	152
7.9 Performance of Load Balancer with Secured File Transfer . . . . .	152

# List of Algorithms

1	Dynamic Multi-factor Authentication (D-MFA) . . . . .	78
2	Continous Multi-factor Authentication (C-MFA) . . . . .	79
3	User Registration . . . . .	82
4	Device Login Process . . . . .	121
5	Communication for Authentication . . . . .	121
6	Zero Trust Network ( ZTN) . . . . .	144
7	Software Defined Perimeter (SDP) . . . . .	147

## List of Abbreviations

<b>Term</b>	<b>Definition</b>
AdaBoost	Adaptive Boosting
AAL	Authenticator Assurance Levels
AES	Advanced Encryption Standard
API	Application programming interface
A-MFA	Augmented MFA
AWS	Amazon Web Services
BAN	Burrows-Abadi-Needham
CSP	Cloud Service Provider
AI	Artificial Intelligence
B-MFA	Blockchain Enabled MFA
CA	Certificate Authority
CIA	Confidentiality (C), Integrity (I), and Availability (A)
CRM	Customer Relationship Management
C-MFA	Continuous MFA
CNN	Convolutional Neural Network
CSA	Cloud Security Alliance
CVSS	Common Vulnerability Scoring System
CoT	Cloud of Things
DID	Decentralized Identifier
DL	Distributed Ledger
DLT	Distributed Ledger Technology
DLP	Data Loss Prevention
D-MFA	Dynamic MFA
DNN	Deep Neural Network
EEP-ROM	Electrically Erasable programmable read-only memory
ETSI	European Telecommunications Standards Institute
FN	False Negatives
FP	False Positives
GUDID	Global Unique Device Identification Database
HOG	Histogram of Oriented Gradients
IBC	Identity-Based Cryptography
IBE	Identity-based Encryption
IETF	Internet Engineering Task Force
IoT	Internet of Things
IT	Information Technology
IaaS	Infrastructure as a Service



<b>Term</b>	<b>Definition</b>
IAM	Identity and Access Management
ITRC	Identity Theft Resources Centre
IDS	Intrusion Detection System
IPS	intrusion prevention system
JWT	JSON Web Tokens
KGC	key generation center
M-A-C	Message Authentication Code
MFA	Multi-factor Authentication
MQTT	Message Queuing Telemetry Transport
MTCNN	Multi-task Cascaded Convolutional Neural Networks
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
OEM	Original Equipment Manufacturer
OTP	One Time Password
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PII	Personally Identifiable Identifiers
PKI	Public Key Infrastructure
QoS	Quality of service
RFID	Radio Frequency Identification
SIEM	Security incident and event management
SECaaS	Security as a service
SaaS	Software as a Service
SLA	Service Level Agreement
SDP	Software Defined Perimeter
SDN	Software Defined Networking
SHA	Secured Hashing Algorithm
SPF	Single Point of Failure
SSD	Single Shot Detectors
SSL	Secure Sockets Layer
SSO	Single Sign On
SSI	Self-Sovereign Identity
TLS	Transport Layer Security
TTP	Trusted Third Parties
VC	Verifiable Credentials
WSN	wireless sensor networks
W3C	World Wide Web Consortium
ZTN	Zero Trust Network

# 1 Introduction

Advances in communication technology allow users and devices to be integrated and interconnected in networks. It provides different ways to communicate between systems and users. It is most appropriately reflected in today's data-driven society. Simultaneously, it opens up interesting challenges related to user privacy, resource authorization, and user authentication. Information technology (IT) assets require the enforcement of strict cyber security policies and their immaculate implementation to prevent privacy violations and fraudulent use. Any activity in the modern communication networks and digital world can be generalized as a digital *transaction*. Such a digital transaction involves three components: identity, data, and algorithm based on a trust model.

Cybersecurity is a means designed to protect networks and devices from exterior threats. Companies often employ cybersecurity professionals to protect confidential information, maintain employee productivity, and build customer trust in products and services. The world of cybersecurity revolves around the industry standard for confidentiality (C), integrity (I), and availability (A), which is commonly known as the CIA Triad. Security means that only authorized users can access the data. A key element of network security is the use of authentication mechanisms. For example, a username identifies the account a user wants to access, while a password is a mechanism to prove that the user is who they say they are. According to Cybercrime Magazine \*, cybercrime will cost the world USD 10.5 trillion annually by 2025. Additionally, the global cost of cybercrime is expected to increase by almost 15% annually over the next four years.

Since its inception, cloud computing has evolved and matured, encompassing complex forms of networking and inter-communication to perform digital transactions. As per *National Institute of Standards and Technology (NIST)*, Cloud Computing Reference Ar-

---

\*<https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/>

chitecture [1], five major actors influence and are impacted by cloud computing, along with its security implications. This Reference Architecture also focuses on threat and risk perceptions for cloud consumers and providers. Eventually, cyber security aspects with cloud computing need a high level of emphasis to ensure desired authentication and authorization for the associated users. The concept of cloud users is also changing with time, with devices or machines also performing active roles as cloud users, compared to human users. Hence, human users and devices as cloud users need due deliberation to have a desired framework for cloud security.

### 1.1 Cloud Computing

Cloud computing is a technology that allows users to access various computing resources, including servers, storage, applications, and services, over the Internet on a pay-per-use basis. Cloud resources and services can be used remotely by individuals or organizations. Cloud computing offers scalability, flexibility, cost-effectiveness, and high availability. It allows users to quickly scale their computing resources up or down as needed, pay only for what they use, and access their data and applications from anywhere with an internet connection. Cloud computing is characterized by elasticity, high availability, data backup and recovery, and security measures. It enables organizations to reduce IT infrastructure costs and increase efficiency, agility, and scalability by shifting from on-premises data centers to cloud computing.

#### 1.1.1 Service, Deployment Models and Characteristics of Cloud Computing

Due to the uniqueness and technical complexity associated with the cloud, it can be better explained under the following details.

1. Cloud Services.

- a) *Infrastructure as a Service (IaaS)*: IaaS provides basic computing resources like processing power, storage, and networking over the internet. It offers a virtualized environment allowing businesses to create and run their software

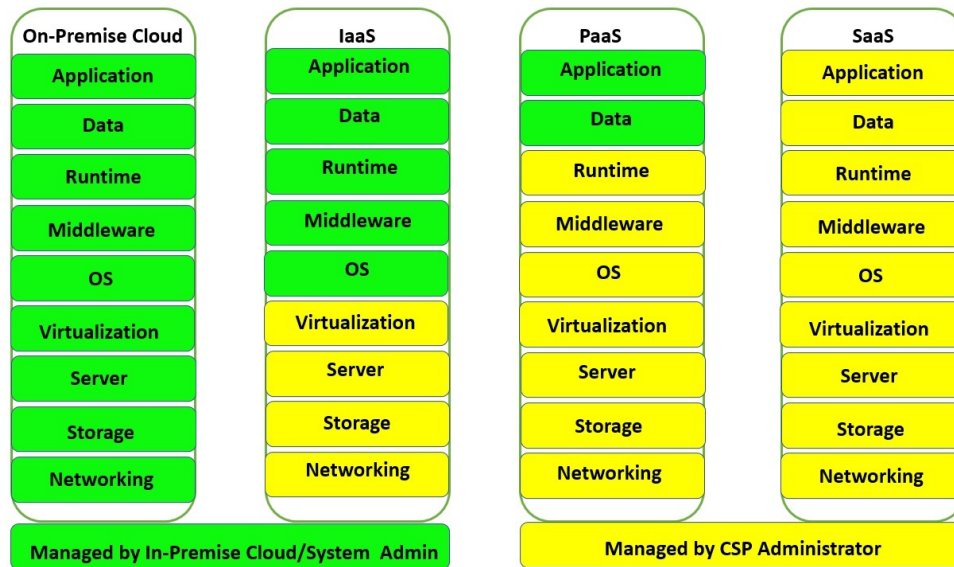
applications and manage their operating systems and middleware. Examples of IaaS providers include Amazon Web Services (AWS), Microsoft Azure, and Google Compute Engine.

- b) *Platform as a Service (PaaS)*: PaaS provides a platform for developers to build, deploy, and manage applications over the internet. It includes tools, libraries, and other resources that developers can use to create their applications, and it typically contains features like automatic scaling and load balancing. Examples of PaaS providers include Heroku, Google App Engine, and Microsoft Azure.
- c) *Software as a Service (SaaS)*: SaaS provides software applications accessed over the internet rather than installed locally on a user's device. The Software is typically hosted on a remote server and is accessed through a web browser or mobile app. SaaS applications can range from productivity tools like email and office suites to more specialized business software like customer relationship management (CRM) systems. Examples of SaaS providers include Salesforce, Microsoft Office 365, and Dropbox.

The cloud administrator is responsible for managing desired system administration and control activities. In the case of a private and in-premise cloud, the cloud administrator is fully accountable for managing the cloud system control activities. However, in the case of public and service provider clouds, commonly known as *Cloud Service Provider (CSP)*, the in-premise cloud administrator and the system administrator of CSP come into play. Detailed control activity of cloud system administration and the responsibility of each stakeholder is depicted in Figure 1.1<sup>†</sup>. For ease of system administration & control and easy deployments, this research uses a technical mix of in-premise private cloud and services from IaaS operators. It is to provision desired Infrastructure for implementing a testbed comprising components from multiple service providers. Adopting such means facilitates easy provisioning of application deployment, storage for data, and runtime environment over the intended Operating System (OS) for this research. It also enables the expected output of this research to be versatile and flexible for deployment on various OS platforms.

---

<sup>†</sup><https://dev.to/artemkobilinskiy/cloud-service-models-saas-paas-iaas-which-is-better-for-business-574k>



**Figure 1.1:** System Administration and Control of Various Clouds Services

Iqbal et al. [2] have presented a comprehensive review of security concerns of the cloud-service delivery model and covered the means of solutions and approaches to handle them. Khoda et al. [3] surveyed service-based cloud computing security issues to analyze the state of cloud security and provide a unified taxonomy of security issues over the three-layer model, namely, IaaS, PaaS, and SaaS.

2. Cloud Deployment Models: Cloud deployment models refer to the different ways of deploying cloud computing services. There are four main types of cloud deployment models.
  - a) *Public Cloud*: Public cloud services are owned and operated by third-party cloud providers and are made available to the general public over the Internet. Public cloud services are typically delivered on a pay-per-use basis and are accessible from anywhere with an internet connection. Examples of public cloud providers include AWS, Microsoft Azure, and Google Cloud Platform.
  - b) *Private Cloud*: Private cloud services are operated solely for a single organization and do not share with other organizations. Private clouds hosted on-premises or in a third-party data center offer more control over security, compliance, and customization. Private cloud deployments are typically more

expensive than public cloud deployments and require a significant upfront investment in hardware and Infrastructure.

- c) *Hybrid Cloud*: Hybrid cloud deployments combine public and private cloud environments to provide greater flexibility and scalability. Hybrid clouds allow organizations to take advantage of the cost-effectiveness and scalability of public clouds for non-sensitive workloads while keeping sensitive data and applications on private clouds. Hybrid cloud deployments can be complex and require specialized expertise to manage.
- d) *Community Cloud*: Community cloud services are shared by organizations with similar interests, such as government agencies, universities, or healthcare providers. Community cloud deployments allow organizations to share Infrastructure and resources while maintaining greater control over security and compliance. Community clouds can be hosted on-premises or in third-party data centers and are typically more cost-effective than private cloud deployments.

Jangjou et al. [4] have extensively analyzed security issues faced by Private, Public, Hybrid, and Community clouds. They have also suggested possible countermeasures for the problems.

3. Cloud Characteristics: Cloud Characteristics refer to the distinct nature of the cloud. NIST has defined five essential characteristics of cloud computing.

- a) *On-demand self-service*: Cloud computing allows users to provision computing resources, such as servers, storage, and applications, without a manual approval process. Users can quickly request and configure the resources they need, which the cloud provider automatically allocates.
- b) *Broad network access*: Cloud computing resources are accessible from anywhere with an internet connection. Users can access cloud services using various devices, such as laptops, tablets, and smartphones, and connect from different locations.
- c) *Resource pooling*: Cloud computing providers use a multi-tenant model, sharing

computing resources among multiple users. It allows cloud providers to achieve economies of scale and offer more cost-effective services to their customers. Resource pooling also enables cloud providers to allocate resources dynamically to different users as demand changes.

- d) *Rapid elasticity*: Cloud computing resources can be quickly and easily scaled up or down in response to changing demand. It allows users to quickly add or remove computing resources as needed without investing in new hardware or software.
- e) *Measured service*: Cloud computing providers offer a usage-based billing model where users only pay for the resources they consume. It allows users to easily track their resource usage and adjust their consumption to control costs. Measured service also provides transparency and accountability, as users can see the resources they are using and how much they are paying for them.

PanJun Sun [5] conducted a detailed survey and analysis of cloud characteristics and their security challenges and implications.

A multi-cloud refers to using multiple cloud computing services from different CSPs to address different organizational computing needs. Instead of relying on a single cloud provider for all computing services, an organization can use multiple cloud providers to meet different requirements, such as data storage, application development and deployment, disaster recovery, and more. In a multi-cloud scenario, the capability of CSPs to orchestrate resources to maximize profits without degrading customer expectations is paramount. From the customer's viewpoint, efficient resource selection with adequate security measures in place is equally a matter of concern. Regarding multi-cloud [6] highlighted the open issues both from the points of view of CSP and customers or tenants.

### 1.1.2 Cloud Computing Security Prospective

Cloud computing security refers to the practices and technologies used to protect data, applications, and Infrastructure in cloud computing environments. Some specific aspects of cloud computing security worth discussing are as follows.

1. *Data security*: One of the biggest concerns in cloud computing is the security of the stored data in the cloud. It includes protecting data from unauthorized access, ensuring data confidentiality, and securing data in transit. Cloud service providers use various security measures such as encryption, access control, and data backup to protect data.
2. *Application security*: Cloud applications are also vulnerable to security threats such as denial-of-service attacks, SQL injection attacks, and cross-site scripting attacks. Cloud service providers use security measures such as firewalls, intrusion detection systems, and penetration testing to protect cloud applications.
3. *Infrastructure security*: Cloud infrastructure refers to the physical and virtual components of the cloud, such as servers, storage, and networking. Cloud service providers use security measures such as network segmentation, intrusion detection systems, and security monitoring to protect cloud infrastructure.
4. *compliance and governance*: Organizations that use cloud services must comply with various regulations and standards.
5. *Identity and Access Management*: Cloud service providers use various identity and access management (IAM) tools to ensure that only authorized users can access cloud resources.

The security architecture of cloud computing services refers to designing and implementing security controls and mechanisms to protect cloud computing systems and data from cyber threats and unauthorized access. The security architecture of cloud computing services encompasses various components and factors that work together to provide a secure environment for cloud computing. These components include authentication and access control, data encryption, network security, security monitoring and incident response, compliance, and risk management.

Overall, cloud computing security is a complex and evolving field that requires a multi-layered approach to protect data, applications, and Infrastructure in the cloud. Cloud service providers play a critical role in ensuring the security of cloud environments. At the same time, it maintains the Quality of service (QoS) offered by CSPs. QoS determines



the level of service that cloud providers offer to their customers. It encompasses various factors that affect cloud services' performance, availability, scalability, reliability, support, and security. The CSPs are under Service Level Agreements (SLAs) to offer desired services. Cloud SLA is a contract between a CSP and its customers that specifies the level of service that the provider will deliver. It also outlines the terms and conditions of the cloud service, including its availability, performance, security, support, and disaster recovery capabilities. By defining these factors, cloud providers can ensure that they offer high-quality cloud services that meet the needs of their customers.

### 1.1.3 Criticality of User Identity and Access Management in Cloud

Historically, the accidental disclosure of secret phrases used by thieves for secured gate opening and closing of hidden storage for stolen valuables and its usage by Ali Baba is the first known example of unauthorized access. Similarly, in the technical community, the password was used for the first time at MIT to control access to time-shared computers among the students and faculty. Since then, many technological advancements have taken place in this direction to consider user credentials and authentication methods, considering insecure authentication as the weakest link for the most robust secured chain for cloud computing. With ever-increasing requirements for mobility in computation and human beings replaced with smart and intelligent devices, this has provided a considerable base for establishing *Cloud of Things*.

*Identity Theft Resources Centre (ITRC)*, in the annual report for the year 2020 [7], has listed out the root causes of identity theft and its implications, showing 38% of data breach occurrences are due to improper cloud security configuration. It also revealed that 36% of data breaches could occur due to phishing attacks and related disclosure of user credentials. Vincent et al. [8] did a detailed analysis of *Access Control* mechanisms on cloud covering IAM. Fagan et al. [9] have suggested a clear set of guidelines on cybersecurity aspects of IoT devices on the cloud.

### 1.2 Motivation

No matter how robust security encryption or any cryptographic measure is in place, the critical point of focus for making a cloud ecosystem secure revolves around the end user. It is not the machine but rather the users in front of the system who can access the system depending on the entitled resources for their meaningful usage. However, with the advent of modern Artificial Intelligence (AI) Technologies, it is observed that bots sometimes replace human users [10]. The same can facilitate hacking or injecting malicious code into a secured system. Even a legitimate user may share its credentials with other users for convenience of working or negligently, which could be exploited by malicious users, causing harm to the cloud ecosystem employing unsecured authentication [11]. Since the cloud domain has expanded from human users to machines or devices as users in the cloud ecosystem [12, 13], the focus on safeguarding user identity and a secured authentication system merits deliberation and further research.

User identity systems can either be centralized, decentralized, or federated. However, these could use the forms of user authentication either in traditional, Self-Sovereign, or by Multi-Factor Authentication (MFA) [14] means. MFA is a security mechanism that requires users to provide two or more factors of authentication to access a system or service. MFA has become increasingly crucial in cloud computing environments because of the significant security risks associated with cloud-based services. Many challenges in the applicability of MFA are addressed in several pieces of research [15, 16]. However, many of them are yet to be explored due to the ever-changing threat perspective in a cloud computing environment.

Many untoward incidents involving cyber activities and data breaches have come to light in recent years. These incidents have tarnished the reputation of many organizations with breaches of confidential data. Analysis of Cloud Vulnerability is the primary motivation developed for research work. Since the increasing number of cyber threats that target cloud-based services, MFA has emerged as a potent way of handling it by ensuring desired user authentication [17]. These threats include unauthorized access, data breaches, and denial-of-service attacks. Since MFA provides an additional layer of security [18], it makes

it more difficult for hackers to access cloud resources, even if they manage to obtain a user's credentials.

Another motivation for the research work on MFA in the cloud is the need to comply with industry regulations and data protection laws. Many industries, such as healthcare and finance, have specific regulations that require using MFA to protect sensitive data [19]. Failure to comply with these regulations can result in severe penalties, fines, and legal action.

In addition, MFA can help organizations protect against internal threats, such as employees who may attempt to access data or systems without proper authorization. By requiring multiple forms of authentication [20], MFA can help ensure that only authorized users have access to sensitive resources.

Finally, as more organizations adopt cloud computing, the need for robust security measures, such as MFA, becomes even more critical. With the increasing amount of data and sensitive information stored in the cloud, the risk of security breaches becomes greater [21]. MFA can provide an additional layer of protection that can help mitigate these risks.

### 1.3 Research Gaps

Based on the research literature, the following gaps are identified.

1. Timely detection of cloud vulnerability is key to its timely mitigation to prevent greater damage. Kritikos et al. [22] brought out the vulnerabilities of the cloud and highlighted the important relevance of user authentication of the cloud using web applications. Due to the increased complexity of cyber-physical systems (CPS), cyber-attacks nowadays are more sophisticated and less predictable, which makes risk management tasks more challenging. Kure et al. [23] highlighted the matter in light of its effect on cloud computing authentication. **However, the criticality of the corresponding vulnerability of the cloud remains unexplored.**
2. Surbiryala et al. [24] have conducted a detailed analysis of cloud computing and its characteristics. Hamid et al. [25] explored desirable attributes of cloud computing.

**However, considering cloud security as an essential cloud characteristic, its relevance concerning cloud authentication needs to be explored.**

3. A detailed survey conducted by Shah et al. [26] on authentication factors has brought many aspects of user-friendliness and complexity of implementation. Khan et al. [27] have conducted a detailed review of the authentication framework for the cloud. Voegelé et al. [28] proposed an innovative approach for MFA. **However, utilities of particular authentication factors require in-depth research for an effective and secured MFA framework that does not require specialized hardware and software for its implementation.**
4. Authentication in a multi-cloud environment is a complex issue. Megouache et al. [29] have described the data security and integrity aspects of multi-cloud with its authentication measures. Prithi et al. [30] have explored a trust framework for user authentication in a multi-cloud. **Still, a secured authentication framework for a multi-cloud environment remains unexplored, especially regarding the applicability of MFA.**

Considering the above research gaps, some questions were found pertinent to Cloud Security, specifically on User Authentication for this research. While trying to answer the same, this research mitigates the following identified gaps.

RQ1.1 What are vulnerabilities for Cloud Authentication and their effective criticality?

RQ1.2 What are the desired characteristics of Cloud authentication solutions and their associated threats?

RQ1.3 What are the factors of authentications and their utility regarding user credentials and a device for identification and authentication in light of MFA?

RQ1.4 How an efficient, robust, and secured authentication system with an MFA approach can be engineered for users and devices in a Multi-Cloud environment, taking advantage of modern technologies?

## 1.4 Objectives

Identification and analysis of vulnerability contributors to the cloud and multi-cloud environment is the primary focus of this research. Various characteristics desired for cloud user or device authentication need due deliberation before finalizing the authentication factors for the MFA approach. This research uses modern cryptographic means with MFA at the core of a solution toward secured user and device authentication in a multi-cloud environment. Hence, this research intends to *design and implement secured authentication towards Identity and Accessment Management of multi-cloud environment for maintaining Confidentiality, Integrity, and Availability*. To realize this broad objective, we identify the following objectives based on the motivations and research gaps outlined in Section 1.2 and Section 1.3, respectively.

1. The first objective is to study and conduct vulnerability analysis of Cloud and Multi-Cloud Systems and analyze the desired characteristics of Cloud authentication solutions in light of choosing a suitable authentication factor for user and device identification as part of the MFA process.

The following sub-objectives address this major objective:

- a) To find out the various contributors of cloud vulnerability and rank them to access the most prominent cause of threat critical for cloud security.
  - b) To identify and analyze desired characteristics of Cloud authentication solutions using various aspects for user and device authentication along with hindrances to their applicability to the cloud environment.
  - c) To establish MFA as an additional layer for secured authentication considering the hindrances in its implementation and the applicability of various MFA algorithms to users and devices.
  - d) To analyze various MFA Algorithms regarding the applicability of various factors towards MFA for users and devices.
2. To design and develop an efficient, robust, and secured user authentication system with the MFA technique at its core, using modern cryptographic means. Further,

this research will deploy a Multi-Cloud testbed to evaluate the proposed solution. The following sub-objectives address this major objective:

- a) To implement a safe storage and retrieval mechanism with verifiable credentials for cloud access management on a private cloud using user-friendly means with technological advancements of Distributed Ledger Technology (DLT), which is resistant to a single point of failure.
  - b) To propose and implement an MFA system that does not require specialized hardware or software.
  - c) To explore modern biometric means for effective authentication for user MFA.
  - d) To effectively interface Distributed Ledger (DL) to support the proposed MFA approach.
  - e) To use state-of-the-art cryptographic means to achieve robust and effectively secured user authentication.
3. To create a reliable and secure system for authenticating devices, incorporating the MFA technique and modern cryptographic methods coupled with Distributed Ledger (DL) in the background. The proposed solution will be tested on a Multi-Cloud platform to assess its effectiveness. The following sub-goals achieve this objective:
- a) To propose and implement an MFA system that works on Transport Layer Security (TLS).
  - b) To implement a safe storage and retrieval mechanism for device MAC address for effectively verifying credentials for cloud access management on a private cloud using user-friendly means.
  - c) To use the technological advancements of Distributed Ledger Technology (DLT) in support of the proposed MFA approach to resist a single point of failure.

- d) To use modern cryptographic means like PKI and SHA to achieve robust and effectively secured device authentication in the multi-cloud scenario.
4. Use state-of-the-art modern techniques in a user-friendly way for device and user authentication over a multi-cloud environment.

The following sub-objectives address this major objective:

- a) To use the Zero Trust Network(ZTN) approach for implementing a secured user and device authentication.
- b) To implement Software Defined Perimeter (SDP) for mutual authentication of users and devices for access control.
- c) To test and validate the proposed technique for Confidentiality, Integrity, and availability for private as well as Multi-cloud ecosystems.

### 1.5 Contributions of the Thesis

To mitigate the vulnerability associated with user authentication, MFA is an effective means since this mechanism tries to enhance the degree of difficulty for the attacker to get a legitimate entry to the secured system as a genuine user of the cloud system.

The researcher has put his best efforts into incorporating various modern technological means implemented and tested over Private Cloud and a multi-cloud scenario comprising infrastructure from multiple commercial CSPs over the internet. Based on the objectives mentioned above, our research work makes the following contributions:

1. The research identifies the most prominent reasons contributing to cloud security using NVD and NIST-backed repositories and methodology, respectively. It also explores the possible mitigation measures for tackling the causes that make the cloud vulnerable. After analyzing the characteristics of cloud authentication, different authentication factors are explored to apply them in the cloud for effective use in MFA. Emphasis is given to planning a solution without requiring specialized hardware or software to implement MFA.

2. User-friendly web-enabled means have been incorporated for system security and ease of access. Cryptographic standards like Public Key Infrastructure (PKI) and Secured Hashing Algorithm (SHA) have been implemented along with web security. Utilization of DLT has been incorporated to avoid a single point of failure. Blockchain and Smartcontract have been deployed over the cloud with an IaaS facility from commercially available CSPs for multi-cloud and multi-tenant testbeds.
3. Considering device authentication in the Cloud of Things (CoT) and associated vulnerabilities, a Blockchain-based, smart-contract-controlled solution is designed and implemented for lightweight device authentication in a CoT. Cryptographic means like PKI and SHA256 have been incorporated. Blockchain and Smartcontract have been deployed over the cloud with an IaaS facility from commercially available CSPs for multi-cloud and multi-tenant testbeds. The performance of device authentication is evaluated over a multi-cloud testbed.
4. ZTN technology is implemented over Software Defined Perimeter (SDP) architecture for secured user and device authentication using multi-cloud testing. An insight into human behavioral patterns and their applicability to user authentication is also covered in this thesis.

### 1.6 Thesis Organization

As depicted in Figure 1.2, after introducing the fundamental research preliminaries in Chapter 1, the rest of the thesis is organized into chapters as follows:

**Chapter No:2** deliberated an aspect of cloud vulnerability, particularly its relevance to user or device authentication in the cloud. It also gave an overview of the existing advanced authentication approaches. It also briefly reviewed the closely related work relevant to our contributions.

**Chapter No:3** explored various authentication factors as applicable to the emerging mechanism of MUF. The uniqueness of users and devices in the cloud environment, along with their privacy preservation, has also been highlighted while negating the need for specialized hardware and Software for MFA implementation.

**Chapter No:4** explored details of biometric characteristics in the form of face recogni-



tion for inclusion in the MFA process. In this direction, details of the least complex and high-performance facial recognition methods have been explored to pick the suitable one for usage in MFA. While doing so, the use of only OEM-fitted hardware is considered.

**Chapter No:5** highlighted Self-Sovereign Identity (SSI) with its smooth amalgamation with Distributed Ledger Technology (DLT) for efficient and user-friendly application. It proposed, implemented, and tested a uniquely identifying and authenticating means for users with MFA in a cloud environment. As an efficient and continuous measure of user authentication, this modern approach uses facial biometrics with Convolved Neural Network (CNN) technology for implementation and testing.

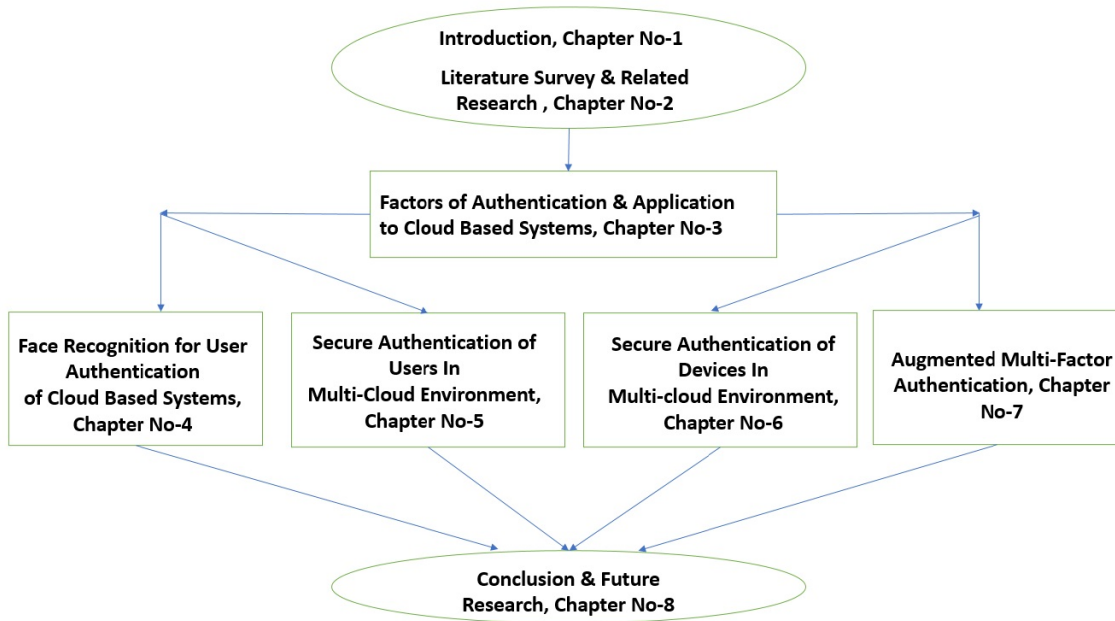
**Chapter No:6** proposed an efficient and continuous measure of device authentication in the Cloud of Things (CoT), with a modern approach using SSI with Decentralized Identifiers(DIDs). It used a lightweight authentication method and implemented various test cases of liveness test for devices in CoT. A Distributed Ledger (DL) based secured authentication approach has been proposed and tested to eliminate single points of failure (SPF) and dependency on centralized Trusted Third Parties (TTP) for authentication. It can uniquely identify and authenticate devices over a multi-cloud of things scenario.

**Chapter No:7** proposed a ZTN-based approach using the SDP mechanism to authenticate users and devices in the multi-cloud environment. Performance evaluation of this state-of-the-art approach has been done for its applicability.

**Chapter No:8** concludes the thesis with a summary of our contributions and lessons learned. We also briefly provide some insights on the possible extensions to our work.

### 1.7 Chapter Summary

Security in cloud computing is constantly evolving to keep up with newly discovered risks and is often identified after incidents. Cloud computing presents a significant threat to all involved parties due to its complex architecture, shared resources, and disruptive nature. Therefore, everyone must comprehend the risks and take appropriate measures to address them. To effectively mitigate these risks, security must be incorporated into every layer of a cloud computing platform by utilizing best practices and emerging technologies. Cloud computing security protects data, applications, and Infrastructure hosted on cloud computing platforms. Critical considerations for cloud computing security include data



**Figure 1.2:** Organization of Thesis

protection, IAM, network security, compliance, and incident response. Hence, these aspects need detailed review before a secure authentication solution for the cloud is proposed. Towards the IAM of the cloud, unique user identity and its verification for authentication are vital steps. The genuineness of users or devices of the cloud can be verified using various authentication factors that need detailed deliberation.

## 2 Literature Survey & Related Research

Cloud computing provides enormous opportunities as the Cloud Service Providers (CSPs) have extended *Everything As A Service* model. However, that brings a plethora of threats and vulnerabilities. These threats and vulnerabilities, if not detected and acted upon, are likely to disrupt the desired service and, in turn, have the potential to fail the whole purpose of adopting cloud services. Hence, a detailed deliberation is done on the security aspects of the cloud and possible mitigation measures for the factors causing concern for the cloud.

### 2.1 Concerns in Cloud Security

Cloud security concerns refer to the potential risks and threats associated with storing, processing, and accessing data and applications in cloud environments. Depending on the service the cloud provides, it can be further categorized.

#### 2.1.1 IaaS Security

Servers or storage hardware can be compromised through physical access, which might result in denial-of-service assaults and data loss. Confidentiality may also be at stake if an attacker accesses a data center. However, this risk can be reduced by encryption techniques and access control enforcement. But attackers might still be able to get through these safeguards measures. Furthermore, cloud software may have bugs and vulnerabilities that can be used against it. Virtual machines can be copied and attacked without being noticed since they can be moved and saved as files [31]. As a result, attackers can try to compromise the system without being detected.

### 2.1.2 PaaS Security

PaaS Cloud systems are exposed to various security risks, such as the potential for users to escape their virtual machines, which could result in the hypervisor crashing and causing a denial of service. As with other cloud services, the user is not responsible for maintaining the infrastructure, and the hypervisor may be managed by unscrupulous administrators who can access private data and potentially steal intellectual property [3]. It is essential to carefully consider what data is stored in such environments and how it is secured. There is also a risk associated with transmitting sensitive data over the public internet, especially when it passes through a hypervisor that is shared with other tenants. Another virtual machine could intercept the network traffic and compromise the confidentiality of the information.

### 2.1.3 SaaS Security

The primary security issue with SaaS is inadequate identity management. However, users of multiple SaaS applications may encounter varying security and identity systems. Additionally, clients often desire to store sensitive data within specific geographic locations, but with SaaS, users cannot guarantee the location of their data storage. Moreover, there currently needs to be robust standards to assess the security of SaaS or cloud security, in general, [32]. Policymakers should develop specific standards that allow users to evaluate the security of SaaS applications before using them.

### 2.1.4 SECaaS Security

Security as a service (SECaaS) is essential for enterprise users to fully understand what information can go through the SECaaS provider's hardware before allowing it to be placed on their premises. While the SECaaS provider is responsible for delivering security, the ultimate responsibility for security still rests with the enterprise user. To ensure the security of sensitive data, it is recommended that an intrusion detection system (IDS) or an intrusion prevention system (IPS) be used to filter the data before it is transmitted to

the Cloud. Additionally, a stateful packet inspector can monitor data passing through the appliance.

In IaaS, PaaS, and SaaS, security is primarily focused on limiting network connectivity. In SECaaS, permissions can be used to control access to specific resources. SECaaS can bridge the gap between the security provided by CSPs and the security that should be implemented on company premises. However, it is the CSP's responsibility to provide security solutions for IaaS, PaaS, and SaaS, and these typically only include basic security features like usernames and passwords, access control lists, or session token IDs.

SECaaS can be used to manage enterprise network configuration to achieve a secure state and to secure other cloud services. For high-risk enterprises, having a secure connection to the Cloud is essential. By installing and deploying IPS and Data Loss Prevention (DLP) solutions on the Cloud, data transmitted from enterprise premises can be inspected and prevented from being sent to another cloud without detection. It adds an extra layer of security between enterprise assets and equipment connected to the cloud [33].

## 2.2 Cloud Vulnerability & Priotization

Cloud vulnerability refers to the potential weaknesses and security gaps in cloud computing systems that cybercriminals and other malicious actors can exploit. These vulnerabilities can arise from various sources, such as misconfigured systems, insecure data storage practices, poor access controls, and inadequate encryption. These may also pose a greater vulnerability on the cloud due to their combined effect.

### 2.2.1 Cloud Vulnerability

Several factors can contribute to making a cloud vulnerable to attack. Some of the key factors include:

1. *Human error*: Human error, such as misconfiguration or neglect in applying security patches, can also make the cloud vulnerable to attack.

2. *Inadequate authentication and access control*: If weak authentication mechanisms and access controls are in place, it becomes easier for attackers to gain unauthorized access to the cloud.
3. *Insecure APIs*: Application programming interfaces (APIs) connect different cloud-based system components. If APIs are not adequately secured, attackers can exploit them to access sensitive data.
4. *Insufficient monitoring and logging*: Without adequate monitoring and logging, it can be difficult to detect attacks and identify vulnerabilities in the cloud.
5. *Lack of encryption*: If data in the cloud is not encrypted, attackers can easily access and steal it.
6. *Outdated software and hardware*: If the cloud is running on outdated software and hardware, it can have known vulnerabilities that attackers can exploit.
7. *Poorly configured security policies*: If policies are not correctly configured or enforced, the cloud can be vulnerable to attacks.
8. *Shared resources*: Shared resources in a cloud environment can create vulnerabilities, as attackers can potentially gain access to sensitive data or services by exploiting weaknesses in the shared resources.
9. *Third-party integrations*: Integrations with third-party services can introduce vulnerabilities into a cloud-based system, as attackers can exploit these integrations to access sensitive data or services.

Organizations must implement robust security measures such as strong authentication and access controls, data encryption, regular vulnerability assessments, and continuous monitoring of cloud systems to mitigate cloud vulnerabilities.

### 2.2.2 Cloud Vulnerability Prioritization

*Cloud Security Alliance (CSA)* is an organization that promotes the use of best practices for providing security assurance within cloud computing. Various literature published by CSA analyzes the types of causes and their criticality for contributing to cloud vulnerability.

A thorough contextual analysis has been carried out in this research. On contextual analysis of the National Vulnerability Database (NVD)<sup>†</sup> (updated up to Dec 2022) with Common Vulnerability Scoring System (CVSS) in accordance to CSA, *The Treacherous 12* [34] as well as *Security Guidance for Critical Areas of Focus in Cloud Computing v4.0* [35], it is observed that insufficient Identity and Access Management (IAM), as well as IAM, comprises cloud vulnerability amounting to 23.6% and 45.3% respectively as depicted in Figure 2.1. **This answers the Question No 1.3.**

## 2.3 Cloud Authentication

Authentication is verifying the identity of a user, device, or system attempting to access a resource or service. Authentication aims to ensure that only authorized users or devices are granted access to protected resources, such as computer systems, applications, networks, or online services.

The authentication process typically involves the following steps [36].

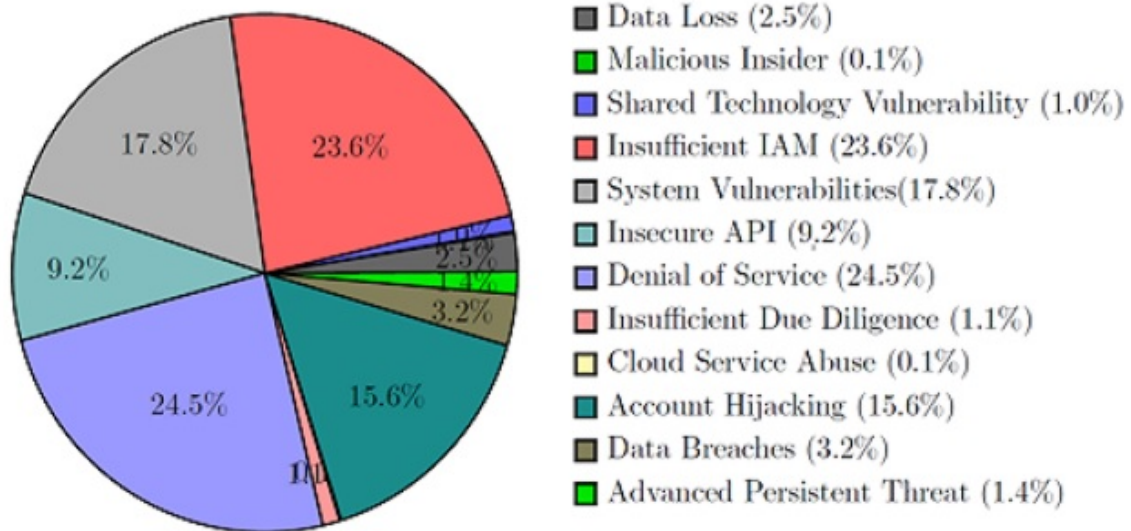
1. The user or device provides some form of identification, such as a username or password, a digital certificate, a biometric identifier (such as a fingerprint or face recognition), or a physical token (such as a smart card or key).
2. The system verifies the provided credentials against a trusted source, such as a database of authorized users, a certificate authority, or a secure authentication server.
3. If the credentials are valid, the system grants access to the requested resource or service, and the user or device is authenticated.

---

<sup>†</sup><https://nvd.nist.gov/>

**Prioritization Ranking : CSA Treacherous12**

1. **Denial of Service (24.5 %)**
2. **Insufficient IAM (23.6%)**
3. **Account Hijacking (15.6%)**



**Prioritization Ranking : CSA Guidelines V4**

1. **IAM (45.3 %)**
2. **Management Plane (22.0%)**
3. **Data Security & Encryption(5.6%)**

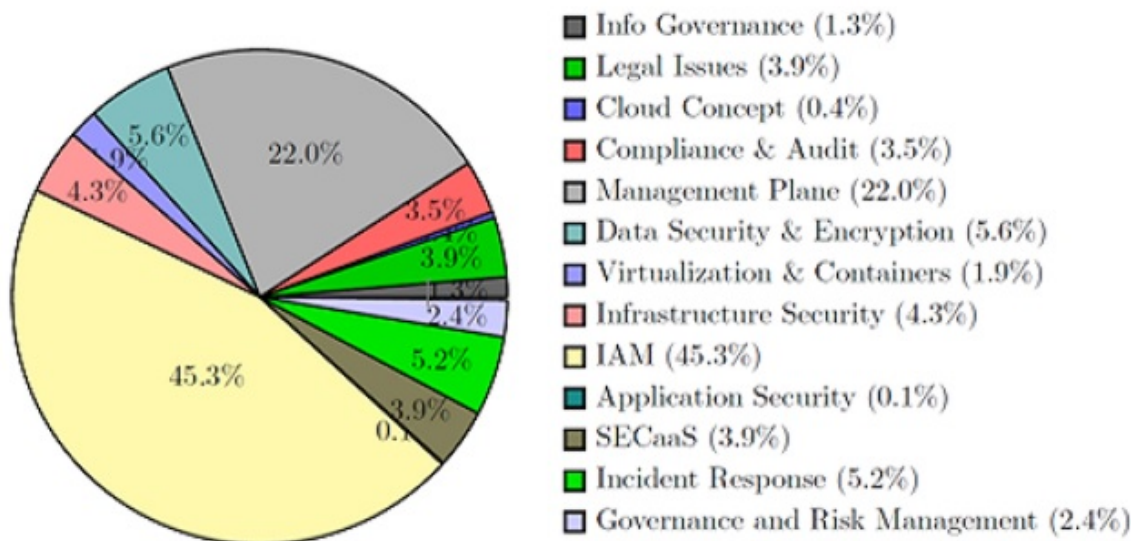


Figure 2.1: Contributors Towards Cloud Vulnerability



### 2.3.1 Cloud User Authentication

Cloud authentication is the process of verifying the identity of a user or device that is trying to access cloud services or resources. It is a critical component of cloud security as it ensures that only authorized individuals or devices can access sensitive data and applications stored in the Cloud. This authentication process is crucial for ensuring the security and privacy of cloud computing systems.

The cloud authentication process typically involves the following steps [37].

1. *User initiates access:* The user attempts to access a cloud service or resource by providing their credentials, such as a username and password.
2. *Identity verification:* The cloud service provider verifies the user's identity by comparing the credentials provided with the user's account information stored in their authentication database. It can include checking the username and password combination and possibly additional factors such as a security token or biometric verification.
3. *Session management:* The cloud service provider establishes a session for the authenticated user, allowing the user to access resources for a limited time. The session can be terminated automatically after a set period of inactivity or manually by the user.

#### 2.3.1.1 Characteristics for Cloud Authentication

Establishing digital identity is technically challenging as it typically requires verifying individuals and always involves authenticating individuals via an open network. This situation creates numerous opportunities for impersonation and other attacks, which can result in false claims of someone's digital identity. NIST has published consolidated guidelines [38] which provide recommendations on types of authentication processes, including choices of authenticators at various Authenticator Assurance Levels (AALs). It describes details of authenticator and verifier requirements along with usability considerations. Some com-

mon characteristics of cloud authentication are listed for our research consideration from available literature and other open-source data, as enlisted in Table 2.1.

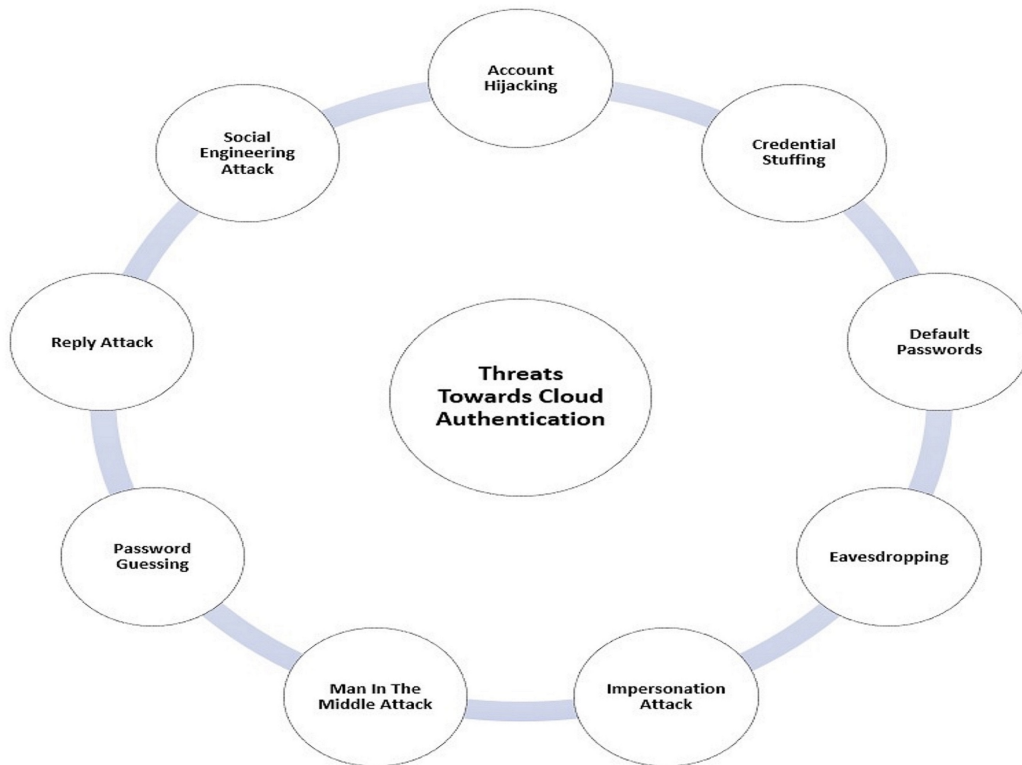
### 2.3.1.2 Threats Towards Cloud Authentication

The system verifies the user's credentials and confirms the user's identity to authenticate the user. The system designers use several established methods to authenticate users of the system. Further access control measures for accessing system resources are enforced based on successful authentication. The gravity of the situation has increased exponentially as nearly every system has been networked and connected to the internet. Threats to authenticators can be categorized based on attacks on the types of authentication factors that comprise the authenticator. Passwords have traditionally been the most commonly used means of establishing identity and authorizing additional resources to the identified user. While the well-known and conventional methods give some convenience to users, they are also determined to be significantly susceptible variables that pose a significant threat to the system, user credentials, and system resources [39]. The following are some of the most well-known susceptibilities, triggering causes, and processes considering cloud authentication, as depicted in Figure 2.2.

1. *Account hijacking*: An attacker stealing or hijacking a cloud account is known as cloud account hijacking. In identity theft schemes, cloud account hijacking is a typical strategy in which the attacker utilizes stolen account information to carry out illegal or unauthorized behavior. In reality, the attacker usually impersonates the account owner using stolen credentials to hijack a cloud account [40]. Attackers might use stolen credentials to access sensitive sections of cloud computing systems, jeopardizing their security, integrity, and availability.
2. *Credential Stuffing*: In several cases, hackers have posted hacked and compromised credentials on the dark web. For credential stuffing, the attacker searches the dark web for an already hacked password. Then, an attempt is made to penetrate the system using the already compromised password as a credential. Similar efforts are made with other accounts of the same user with passwords that have been hacked to access the system. When a person has numerous accounts in the system, it is usual

**Table 2.1:** Characteristics for Cloud Authentication

<b>Characteristics</b>	<b>Explanation</b>
<i>Auditability</i>	Cloud authentication should facilitate auditable function to keep track of user activity and provide a record of access to cloud-based resources for compliance and security purposes.
<i>Controllability</i>	Cloud authentication should be either centralized or decentralized to provide a platform of control and manage user access to cloud-based resources.
<i>Cost-effectiveness</i>	Cloud authentication should be cost-effective to implement and manage, as organizations may need to authenticate a large number of users and devices.
<i>Ease of use</i>	Cloud authentication systems should be easy to use for both administrators and end-users. This includes providing intuitive interfaces and clear documentation, as well as offering self-service options for password resets and other routine tasks.
<i>Flexibility</i>	Cloud authentication should be flexible enough to support various authentication mechanisms, such as passwords, biometrics, and hardware lock etc.
<i>Integration</i>	Cloud authentication systems should be able to various cloud-based services and applications to provide a seamless user experience, as well as on-premises systems. This ensures that users can access resources across different platforms and environments.
<i>Multi-factor authentication (MFA)</i>	Cloud authentication often involves MFA, which requires users to provide at least two forms of identification, such as a password and a security token or biometric authentication, to gain access to cloud resources.
<i>Reliability</i>	Cloud authentication should be highly reliable, as downtime can impact the availability of cloud-based services.
<i>Scalability</i>	Cloud authentication systems should be designed to scale as the number of users and resources grow. This means that authentication systems must be able to handle large volumes of requests and be flexible enough to accommodate different types of cloud services.
<i>Security</i>	Cloud authentication systems should be highly secure to protect against unauthorized access and data breaches. This includes using strong encryption, secure protocols, and regularly auditing user access and activity.
<i>Visibility &amp; Analytics</i>	Cloud authentication systems should have a familiar user interface to enable effective interaction with the system. It should have in-built tools for desired analytics operation on the system events.



**Figure 2.2:** Threats towards Cloud Authentication

practice to share a single password for convenience. Users' habits of not choosing separate passwords for multiple accounts and reusing a common password are exploited in this form of attack [41]. Organizations like the *Open Web Application Security Project (OWASP)* have proposed many techniques to combat credential-stuffing attacks. The most generally recommended methods include separate passwords for various user accounts and using the CAPTCHA system for authentication.

3. *Default Passwords:* A pre-installed and factory-configured password is a default password. The system administrator and users do not update the system's default password for convenience and occasionally due to ignorance. The failure to consider this essential factor is a concern for rendering the system vulnerable to cyber-attacks [42]. As a remedy, the system urges the system user to change the default password at initial use and with similar redirections for routine password changes with a pre-defined level of difficulty. Password policies such as a minimum length and a mix of upper- and lower-case alphabets, digits, and special characters are imposed on the user.

4. *Eavesdropping*: The attacker uses this approach to secretly listen to and sniff private conversations between two people without their consent or knowledge. It is thought more straightforward for the attacker to control the system's networking equipment and network traffic [43]. Sniffing of non-secured HTTP and FTP-like service traffic using the default networking port and data traffic is studied. In that case, an attacker can quickly uncover the password and credentials from the analyzed network's plain-text data traffic using tools or software like Wireshark. However, employing encrypted services with standard encryption techniques may alleviate this.
5. *Impersonation Attack*: In such an attack, an unauthorized user or wrong user attempts to act as a genuine user by fraudulently acquiring the credentials of the actual user [11]. Such attacks could lead to serious data breaches in highly secured working environments like bank and defense sectors, which handle highly sensitive, crucial information and applications. Biological uniqueness associated with the user can address it.
6. *Man-in-the-Middle Attacks*: A man-in-the-middle attack can steal user credentials if the attacker can get inside the sender and the receiver. The attacker may now transmit and receive all data exchanges between the two computers. As a result, the attacker can pose as a sender to the recipient, and vice versa [44]. The attacker has the power to modify and delete sections of the communications in transit in addition to sending and receiving messages. As a result, the attacker can collect sensitive information, such as the username and password, and use it for malicious purposes.
7. *Password Guessing*: Password guessing is a technique in which an adversary attempts to guess the username and password of a legitimate user and then authenticate as being such. The attacker merely guesses probable passwords that the user will likely use in a password-guessing attack. A brute-force attack is generally an exhaustive search that an adversary can use to guess a password. It is an attack in which the attacker attempts to generate all potential password combinations and then authenticates to the system using the username and various password combinations [45]. The password's length determines the time it takes to carry out this assault. A dictionary attack is when an attacker tries each word in a dictionary as a password to breach a password-protected authentication system. A password dictio-

nary attack is still classified as both brute-force and a dictionary attack. Similarly, a password-spraying assault is a sort of attack that depends on a small number of frequently utilized passwords.

8. *Replay Attacks*: Another prominent method of attacking authentication methods is a replay attack. The replay attack involves a hacker copying a password or Credential from one organization and utilizing it to authenticate with another. The goal is to mimic the user whose credentials or passwords have been copied. The attacker replicates the message or credentials and transmits them to an authenticator, hoping they will be validated successfully [46].
9. *Social Engineering Attack*: Using personal and interpersonal skills is common in social engineering approaches, although applying information technology is not always essential. When a user is subjected to a social engineering attack, the adversary tries to persuade them so that they are obliged to disclose certain information or even do a specific action [47]. In today's world, social engineering may take three primary forms. In-person, phone, and digital social engineering are the three types of social engineering attacks.

A detailed analysis of the types of attack mentioned above and corresponding suggested countermeasures as depicted in Table 2.2.

The preceding explanation and analysis describe potential threats to Cloud Authentication, their effective criticality, and suggested remedial measures. **This answers the Research Question No 1.3.**

Impersonation attack is a typical case study applicable to both human users and devices as users of the cloud ecosystem. In all practical scenarios, the original human user or device could be swapped with a duplicate or a malicious one. Cloud authentication systems often use multiple layers of security, like multi-factor authentication, encryption, and continuous monitoring for suspicious activities to address these threats. It is also essential for users to follow best practices for password management, such as using strong and unique passwords and avoiding password reuse. Overall, the effectiveness of security measures depends on the specific threats faced and the level of protection needed for the resources being accessed. From the above discussion, we have concluded that *MFA and*

**Table 2.2:** Threats to Cloud Authentication and Suggested Remedial Measures

<b>POTENTIAL THREAT</b>	<b>SUGGESTED REMEDIAL MEASURES</b>	<b>REF</b>
Account Hijacking	Use of MFA Use of One Time Password (OTP) Use of End-to-End Encryption	[48], [40]
Credential Stuffing	Use of different passwords for different accounts Apply MFA	[49], [41]
Default Passwords	Use of random and unique default passwords Prompting and forcing users for changing default passwords	[49], [42]
Eavesdropping	Adopting strong and robust encryption mechanism	[50], [43]
Impersonation Attack	Use of biometric means to uniquely identify the user Use of MFA	[51]
Man-in-the-Middle Attacks	Use of Virtual Private Network (VPN) Adopting strong and robust encryption mechanism	[52], [44]
Password Guessing	Using long and strong passwords that are not obvious No reuse of same and already used password	[53], [45]
Replay Attacks	Use of a strong and robust Challenge-Response means	[52], [46]
Social Engineering Attack	Educating users on how to avoid being a victim of in-person, over-the-phone, and digital attacks such as phishing or e-mail attacks.	[52], [47]

*use of Biometric means* are effective for mitigating the cloud vulnerability about user identification and related authentication, to make it a *Secured Authentication*.

### 2.3.2 Advanced Authentication Approaches

Several researchers have relatively recently introduced technologies and approaches for secured authentication. Some research work introducing potential techniques and their technological advantages are mentioned below.

1. *Blockchain and Distributed Ledger Technology*. To eliminate the vulnerability of a single point of failure, researchers have switched to the application of Distributed Ledger (DL) Technology. Further, Blockchain application uses inherent cryptographic means for secured user authentication. This approach applies to the user and device unique identification, using cryptographic features for secured authentication. Kumar et al. [54] explored the possible security and privacy issues considering the component interaction in IoT devices and distributed ledger-based blockchain (DL-BC) technology and its applications. Sheth et al. [55] proposed security models in terms of authentication and security using Blockchain, Deep Learning, or the integration of both based on certain characteristics for unique user identification.
2. *Machine Learning or Deep Learning Approaches*. Users and devices are potentially identified uniquely through *Human Biometrics* as well as *Device Hardware Addresses*. Many researchers have explored the same to make the best utilization for user or device identification towards a secured user or device authentication in a cloud. Further safeguarding of user biometric data or device hardware address is usually done using cryptographic methods like hashing or digital certificates for user or device credentials. Also, the application of textit Convoluted Neural Network (CNN) in this regard is effective in several types of research. Oza et al. [56] proposed a CNN-based approach in which the overall network is trained using cross-entropy and the reconstruction error losses. Zulfiqar et al. [57] proposed a face recognition-based system that detects faces in an input image using a Viola-Jones face detector and automatically extracts facial features from detected faces using a pre-trained CNN for recognition towards user authentication.



3. *Self-sovereign Identity.* In a Multi-Cloud environment or a scenario where a user must identify himself whenever accessing a new service or resources spread across many clouds or domains of operation. There is a potential threat of credential reuse or compromise of *Personally Identifiable Identifiers (PII)*. Self-sovereign identity facilitates the protection of PII, and the user controls credentials sharing. It enables the *Single Sign On (SSO)* feature on a Multi-Cloud Environment with secured user authentication. Lux et al. [58] proposed a proof of concept decentralized OpenID Connect Provider by marrying it with Self-Sovereign Identity, which gives users the freedom to choose from a large pool of identity providers. It enabled democratizing the highly centralized digital identity landscape with decentralized means.
  
4. *Zero Trust Network and Software Defined Perimeter.* Zero Trust Network (ZTN) is a security model that requires strict identity verification for access to resources on a network. The ZTN model assumes that all network traffic is potentially malicious; thus, authentication and authorization must occur at every process step. In a ZTN environment, user identities are verified and authenticated before accessing any resources on the network. It is achieved through various methods, including MFA and device profiling. ZTN identity security also involves continuous monitoring of network traffic for any suspicious activity or anomalous behavior. Software-defined perimeter (SDP) is a security framework that uses a "zero trust" approach to secure network connections, which works through a series of steps, including identity verification, device profiling, and network discovery. SDP is a robust security framework that can help organizations reduce the risk of data breaches and cyber attacks by providing secure access to network resources. In this approach, network discovery involves determining the available network resources the user or device can access. It typically uses software-defined networking (SDN) techniques, such as virtual private networks (VPNs) or network segmentation. Omar et al. [59] performed an extensive Comparative Study of Network Access Control and Software-Defined Perimeter. Hatakeyama et al. [60] proposed a security model that authenticates users who request access and then authorizes such requests using various information about users and devices called contexts. Since context is sensitive to user privacy, they also explored a mechanism for sharing contexts under user control, called *Self-sovereign Identity*.

### 2.4 Chapter Summary

Detailed analysis of the probable causes that are potentially against cloud security has been carried out. Identity management is assessed as a prime reason to make the cloud secure. Towards that end, various means like MFA have been studied to enhance cloud authentication security. Modern and comparatively new technological approaches to secured authentication are further to be explored in this research. However, the applicability of MFA for factors not requiring specialized hardware and software for user authentication needs to be explored as a scope of this research. Simultaneously, the uniqueness of each user and device in a cloud ecosystem must be established for effective authentication. Such uniqueness by means of distinct characteristics of the user or device needs to be verified through utilized authentication factors. Similarly, the hindrances in MFA implementation and associated MFA characteristics need due deliberation before its successful implementation. For secure and optimal user or device authentication using modern schemes, these aspects play a crucial role as a sound foundation for dynamically or continuously checking and verifying the authenticity of the user or device as a functional interacting entity of the cloud ecosystem. The following chapters focuses on the selection of biometric identifiers and its implementation for multi-factor authentication of both human users and devices.

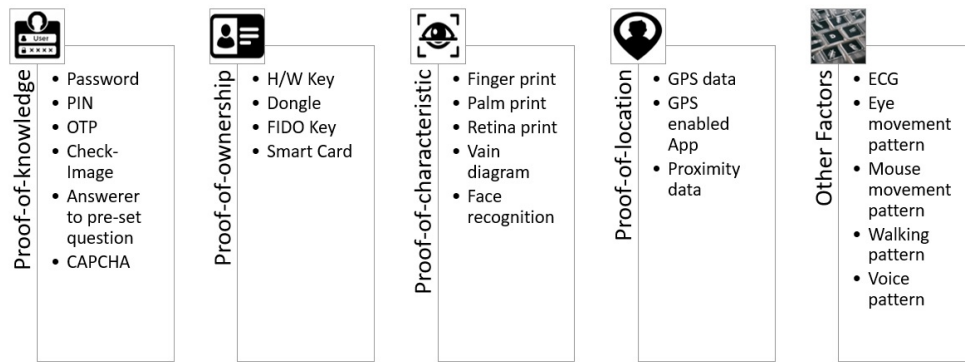
## 3 Factors of Authentication & Application to Cloud Based Systems

*Multi-factor authentication (MFA)* is a security mechanism that requires users to provide two or more forms of identification to access a system or application. MFA is becoming increasingly important in today's digital age as cyber-attacks are becoming more sophisticated. MFA is an essential security measure that can help protect sensitive data and prevent unauthorized access. Hence, organizations must implement MFA wherever possible to increase their systems' security with enhanced authentication. For effective implementation, it is essential to deliberate upon its genesis as an additional layer of authentication to various MFA algorithms. While doing so, this research explores further and eliminates the need for any other specialized hardware requirement for user and device MFA.

### 3.1 Factors of Authentication

Several factors of authentication can be used to establish the identity of a user or device as a legitimate cloud user, as depicted in Figure 3.1. These factors are as follows.

1. *Something the user knows* This factor is based on something the user knows, such as a password, PIN, or a secret question. This factor is commonly used in conjunction with other factors of authentication.
2. *Something the user has* This factor is based on something the user has, such as a smart card, a security token, or a smartphone app that generates one-time passwords (OTP).



**Figure 3.1:** Factors of Authentication

3. *Something the user is* This factor is based on something the user is, such as biometric information like a fingerprint, facial recognition, or voice recognition.
4. *Somewhere the user is* This factor is based on the user's location, such as a specific IP address or geolocation data. This factor can detect fraudulent access attempts from remote locations.
5. *Something the user does* This factor is based on something the user does, such as typing patterns, mouse movements, or other behavioral biometrics.

#### 3.1.1 Additional Layer for Authentication

Adding an extra layer of authentication is a common approach to enhance the security of authentication systems. In this method, users are required to provide two or more types of credentials to gain access to a system or resource. These credentials can be something the user knows (like a password or Personal Identification Number (PIN)), something they have (like a smart card or mobile phone), or something they are (like a biometric identifier, fingerprint, or face recognition) or somewhere the user is (like the Global Positioning System (GPS) signature of the user device) or even something the user does (like specific patterns). To generalize, this combined effect of more than one authentication factor and their unified and simultaneous utilization for user verification is represented in Equation 3.1, which is capable of producing Authentication function results in terms of Boolean

Values [61].

$$F(x) = f1(x1) \cap f2(x2) \begin{cases} 1, & \text{if } f1(x1) \cap f2(x2), \text{TRUE} \\ 0, & \text{otherwise.} \end{cases} \quad (3.1)$$

$F(x)$  is Multi-factor System Authentication Function over user  $x$  with a resultant Boolean Value.  $f1(x1)$  is the First Authentication Factor Function with input value  $x1$ .  $f2(x2)$  is the Second Authentication Factor Function with the input value  $x2$ . And so on, for its applicability to third and subsequent factors for authentication

#### 3.1.2 Emergence of MFA

In the early 2000s, Bill Gates predicted the death of the password as a single measure of account protection. The genesis of MFA goes back to the early days of computing when passwords were introduced as a security measure. However, passwords alone have proven insufficient to protect against cyber threats. It paved the way for the development of more potent authentication methods. MFA emerged as a response to the increasing sophistication of cyber-attacks and the need for more robust authentication methods to protect sensitive data and secure the systems. MFA adds an extra layer of security to the authentication process by requiring users to provide multiple forms of identification to access a system or data.

The initial MFA methods came into existence in the 1980s and 1990s. They typically involved using a physical token that users carried with them, such as a smart card and user passwords. As technology advanced, the authentication process was augmented with additional factors, such as biometric authentication (fingerprint, facial recognition, etc.) and location-based authentication (using the user's physical location as an additional factor). Nowadays, MFA is used in various settings, including online banking, e-commerce, health-care, and government systems, to provide more robust protection against unauthorized access and fraud. The use of MFA has become more widespread in recent years, driven in part by regulatory requirements and industry best practices that have also played a role in driving the adoption of MFA by various CSPs. Presently, researchers have been

**Table 3.1:** Prominent Obstacles for MFA Implementation

<b>Reason</b>	<b>Explanation</b>
<i>Compatibility</i>	MFA may not be compatible with certain legacy systems or software, which can limit its adoption or effectiveness.
<i>Cost</i>	Implementing MFA can involve additional expenses for purchasing hardware tokens, biometric scanners, or other authentication mechanisms.
<i>Integration</i>	Integrating MFA with existing systems and applications can also pose challenges, particularly if they are outdated or not designed to support MFA.
<i>Maintenance &amp; Support</i>	MFA requires ongoing maintenance and support, which can be costly and time-consuming for organizations.
<i>Privacy</i>	The use of biometric authentication can raise privacy concerns as it involves the collection and storage of sensitive personal data.
<i>Regulatory Compliance</i>	MFA may be required for compliance with certain regulatory standards, which can pose challenges for organizations that have not previously implemented such measures.
<i>Resistance from end-users</i>	End-users may resist MFA due to the extra steps involved in the authentication process, which can be time-consuming or perceived as inconvenient.

exploring combining two or more factors of authentication and Public Key Encryption to Ensure authentication in the Cloud Computing ecosystem.

#### 3.1.2.1 Obstacles & Threats to MFA

Implementing MFA can pose challenges because it may involve additional expenses and inconvenience, such as introducing new hardware like portable tokens or biometric scanners. Moreover, users may experience added difficulty during the login process as they must memorize information, access tokens, or undergo biometric scanning, causing extra effort and slowing down the process.

Prominent obstacles for MFA implementation, which directly impact this research, are listed in Table 3.1. At the same time, multiple threats are envisaged in the implementation and adoption of MFA. The most prominent threats, which directly affect the scope of present research for threat modeling of MFA, are listed in Table 3.2. The preceding

**Table 3.2:** Prominent Threats of MFA Implementation

Threat	Envisaged Effect
<i>Biometric Spoofing</i>	Biometric authentication mechanisms may be vulnerable to spoofing attacks, where attackers create fake biometric data to fool the authentication system.
<i>Credential stuffing</i>	Attackers may use stolen MFA credentials to gain access to other accounts belonging to the same user.
<i>Denial of Service (DoS) attacks</i>	Attackers may launch DoS attacks against the authentication system, preventing legitimate users from accessing their accounts.
<i>Insider Threats</i>	Employees or contractors with access to sensitive systems may abuse their privileges to bypass MFA or steal MFA credentials.
<i>Malware</i>	Malicious software such as keyloggers or screen capture tools may be used to steal MFA credentials from infected devices.
<i>Man-in-the-middle attacks</i>	Attackers may intercept communication between users and the authentication system, allowing them to steal MFA credentials.
<i>Social Engineering</i>	Attackers may attempt to trick users into divulging their MFA credentials through phishing attacks or other forms of social engineering.

explanation and analysis describe various authentication factors and their practical applicability as an additional layer for secure authentication. It also describes the potential obstacles in the path of MFA implementation.

#### 3.1.3 MFA Algorithms & Analysis

Though MFA increases the degree of difficulty for the attacker to acquire the original credentials of the user, it can have a variety of implementation methods. Depending on the MFA implementation method, the MFA algorithm can be classified and represented as the following MFA algorithms.

1. *Static MFA Algorithm:* In this approach, the user must undergo the basic challenge-response mechanism to prove genuineness using the system administrator-supplied credentials. The second and subsequent factor, as designed in the system, works in the same mechanism to add to the degree of difficulty for guessing or breaking the credential. In such an approach, once the user successfully responds to all the challenges, the user is active through the log-on session until the user logs out of the

system or is actively logged on, as per the allowed user profile time limit set by the system.

2. *Dynamic MFA (D-MFA) Algorithm:* This approach follows a different mechanism, where the user is initially required to undergo the desired challenge-response mechanism described in the static MFA algorithm. However, to ensure the presence of the genuine user throughout the logged-on session, the user is also thrown upon different intermediate challenges. Upon successful response to the intermediate challenges, the system ascertained that the currently logged-in user is the genuine one, and the user continues to work till a subsequent challenge is given to the user to prove user legitimacy and genuineness.
3. *Continuous MFA (C-MFA) Algorithm:* In this approach, the users must undergo the primary challenge-response mechanism of the authentication system. However, the user is thrown upon a series of continuous challenges to re-prove genuineness, time and again, till the log-on session expires or the user manually logs out of the system.

On a detailed analysis of the algorithms, as mentioned earlier, the points observed are represented in Table 3.3.

From Table 3.3, the following points are inferred.

- (i) Repetitive dynamic, or continuous authentication, is effective in dealing with spoofing and impersonation attacks.
- (ii) With the implementation of a Dynamic or Continuous algorithm, the user is disturbed by intermittent or continuous challenge-response procedures while in the active log-on session.
- (iii) An MFA mechanism, operated without disturbing the logged-in user, is essential for the uninterrupted operation of the logged-in user application to have a potential means to deal with a spoofing or impersonation attack.
- (iv) Applicability of a particular factor without disturbing the user while on an active logged-on session must be explored for this research.



**Table 3.3:** Comparative Analysis of MFA Algorithms

MFA Algorithm	Advantages	Limitations
Static MFA	A time-tested approach to enhancing the degree of difficulty for attacker and user is not disturbed in an active session.	System vulnerable to spoofing attack or impersonation attack.
Dynamic MFA	An approach likely to put less load on system process and resources, as additional factor verification by challenging response is conducted randomly. It is an effective approach to deal with spoofing attacks and impersonation attacks.	user likely to get disturbed in an active login session with system prompts or pop-ups to respond to system challenges in random time intervals
Continuous MFA	An approach likely to put more load on system process and resources, as additional factor verification by challenge response is conducted repeatedly and continuously. It's an effective approach to deal with spoofing attacks and impersonation attacks.	user likely to get disturbed in an active login session with system prompts or pop-ups to respond to system challenges in a repetitive manner

- (v) While doing so, eliminating the additional hardware and software requirements must also be considered.

The preceding comparative analysis and explanation describe various MFA algorithms and their effective applicability in tackling spoofing and impersonation attacks on the authentication system.

#### 3.1.4 Applicability of Factors for MFA

With the help of specified factors, an MFA system can achieve desired goals with effectiveness, efficiency, and user satisfaction for stable and secure user identification and authentication. However, the factors applicable to uniquely identify human users differ from those required to identify the devices in a CoT environment.

### 3 Factors of Authentication & Application to Cloud Based Systems

**Table 3.4:** Factors for MFA Implementation of Users

Factor	Description	Advantages	Limitations	Ref
OTPs	This generates a unique, time-sensitive password for user authentication.	Extra layer of security, hard to crack, expire after defined time	Network issue, vulnerable to man-in-the-middle attacks	[62], [63], [64]
Smart Phone Applications	It generate a unique, time-sensitive code that is used to log in, to verify user identity.	Code regenerated in defined time gap, hence safe from attacks	Device power backup and network dependent, clock desynchronization issues in device and service	[48], [64], [65], [66]
Smart Cards	Use a smart card, which usually includes the user's unique identifier and a digital certificate.	More secure using encryption technology	Card may get lost, the chip may be damaged, Require card reader with software	[67], [68], [69]
Voice Recognition	It works by analyzing the unique characteristics of the user's voice to verify the identity.	Convenient, quick and efficient	False negative in loud background, compromised biometric is irrecoverable	[70], [71], [72]
Face Recognition	By capturing and analyzing the unique features of users' faces to verify their identity.	Convenient, quick and efficient	Large storage requirement, may create data vulnerability, vulnerable to spoofing attack	[49], [73], [74]
Ocular-based Methods	It uses biometric authentication techniques to identify the user by analyzing the iris, retina, or sclera of eyes	Very efficient and difficult to spoof	Need high-quality camera and robust mathematical techniques,	[49], [75], [76]
Finger Print Scanner	Uses a sensor to scan the unique ridges and valleys on the user's finger and compare with stored data.	Ease of use, cost-effective	Scanners may fail, easily replicable fingerprint, compromised data is irrecoverable	[70], [77], [78]
Vein Recognition	Uses the unique pattern of veins beneath the skin's surface in a person's hand or finger to identify them.	Efficient and accurate	Expensive, still vulnerable to spoofing attacks, compromised biometric is irrecoverable	[70], [79], [80]
Thermal Image Recognition	Uses infrared radiation emitted by the human body to identify individuals.	Efficient, can be used from a large distance	Different thermal image in case of fever	[49], [81], [82]
Geo-tagged Location	It works by capturing and analyzing the GPS signature of the user's device, which is used to identify the user.	Effective if user be present at a particular location	GPS may not be accurate at some locations	[70], [83]

#### 3.1.4.1 Applicability of MFA for Users

Table 3.2 and Table 3.4 infers that *Face recognition* as a factor of authentication has the potential to deliver efficient and high-precision functionality with modern and advanced research being already conducted in the field of image processing. Hence, this is considered a potential factor for the MFA functionality of our present research.

#### 3.1.4.2 Applicability of MFA for Devices

MFA is an important security measure in the *Cloud of Things (CoT)* to prevent unauthorized access to devices and data. Device identification is one of the factors used in MFA to confirm the identity of the device attempting to access the CoT. Further, they can be combined to provide a more robust authentication process, ensuring that only authorized devices are granted access to sensitive data and services in the Cloud of Things environment. Table 3.5 elaborates some factors for device identification in MFA of CoT and discusses their advantages and disadvantages.

## 3.2 Uniqueness of Users and Devices

Each human as a user can be uniquely identified with the help of specific biometric characteristics. However, data capturing, storage, and processing of biometric data invariably require specialized hardware like retina reader, fingerprint reader, etc., and associated software. At the same time, the Original Equipment Manufacturer (OEM) fitted web-cam operational efficiency may be explored for face recognition system, which is available mainly in today's PCs and mobile Computing devices. All network devices essentially have a unique hardware or MAC address. It also has the potential functional application to be used as a factor of authentication for networked devices connected to the CoT environment. Table 3.2 and Table 3.4 signify that *Face recognition* as a factor of authentication has the potential to tackle spoofing or impersonation attacks efficiently and with high precision functionality. From the tabular analysis depicted in Table 3.4 and Table 3.5, we concluded that Device Unique Identifiers like MAC addresses can be effectively used as

**Table 3.5:** Factors for MFA Implementation of Devices

<b>Factor</b>	<b>Function</b>	<b>Advantages</b>	<b>Limitations</b>	<b>Ref</b>
<i>Biometric Data</i>	Data like fingerprints, facial recognition, or voice recognition can be used to identify a device.	It can do MFA effectively.	Need specialized hardware and software.	[84], [85]
<i>Device behavior</i>	Device behavior like network traffic patterns, login history, and usage patterns can be used to identify a device.	Applicable to High sensitive applications.	Need Machine Learning for pattern reading.	[85], [86]
<i>Device credentials</i>	Device credentials like device ID, serial number, or Media Access Control (MAC) address can be used to identify a device.	Effectively manageable for thickly populated systems.	Need safe and secure credential storage.	[87], [88]
<i>Device metadata</i>	Device metadata like device type, firmware version, and operating system can be used to identify a device.	It can do MFA effectively.	Need Machine Learning for metadata reading.	[89], [90]
<i>Geo location</i>	location of a device can also be used to identify it.	Integration of GPS functionality.	Depend on GPS hardware or IP address for the location.	[91], [14]
<i>Time-based factors</i>	The time of day or week and duration of device usage can be used as factors for device identification.	Provide time precision application.	Not useful all the time.	[92], [93]
<i>User behavior</i>	The behavior of the user accessing the device can also be used as a factor for device identification.	Can effectively incorporate human user behavior.	Need Machine Learning for behavior pattern reading.	[85], [94]
<i>Unique device identifier</i>	Devices in the CoT have a unique identifier, like a serial number or MAC address.	It can do MFA effectively for dense CoT.	Identifiers need protection for the spoofing attack.	[95], [96]

a factor of authentication when safeguarding its MAC address is ensured from spoofing attacks.

#### 3.2.1 Privacy of Users and Devices in Cloud

CSPs are required to employ appropriately tailored privacy controls. NIST published document SP 800-53<sup>‡</sup> provides a set of privacy controls for CSPs to consider when deploying authentication mechanisms. These controls cover notices, redress, and other important considerations for successful and trustworthy deployments. Towards this end, anti-spoofing measures like the liveness test for facial recognition-based user authentication are potent mechanisms to tackle impersonation attacks and protect the user authentication process. Ensuring the protection mechanism against a spoofing attack, like implementing a digital certificate or cryptographic hashing techniques, is required. Also, providing database storage through a distributed database for storing the digital certificates or the corresponding hash value is essential to implement a secure MFA mechanism for the devices in a CoT environment. **This explanation answers the Research Question 1.3.**

### 3.3 Chapter Summary

After establishing the relevance of MFA in secure authentication, this research explored the possibility of having a specific set of factors that do not require additional hardware or specialized software for implementation. An OEM-fitted webcam is selected to perform facial recognition-based user identification for secure authentication. A facial recognition mechanism with the least computing complexity and high reliability is explored for implementing secure user authentication with the MFA approach. Self-Sovereign Identity (SSI) is an approach in identity management known as a user-controlled Personal Identifying Information (PII) preserving mechanism for user authentication. The desired implementation of Public Key Infrastructure (PKI) based encryption is also applied in SSI. Using various MFA algorithms, the amalgamation of SSI and MFA for user authentication is a way to achieve secure authentication in the cloud computing environment.

---

<sup>‡</sup><https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>

## **4 Face Recognition for User Authentication of Cloud Based Systems**

Face recognition involves capturing an image of a person's face using a camera, which is then processed by an algorithm to extract features such as the distance between the eyes, the shape of the nose and mouth, and other facial characteristics. These features are then compared to a database of known faces to determine if there is a match. There are various face recognition applications, such as security and surveillance, authentication, and identification of users in cloud-based systems.

### **4.1 Analysis of Facial Recognition**

Considering the user's face to be utilized as a biometric authentication factor, it essentially requires three mutually related processes for facial recognition. The processes are:

1. Capturing of the image.
2. Detection of the Face in the captured image.
3. Comparison of detected facial features with the registered user's stored credentials.

For capturing a simple image, a system-connected webcam is recommended to be used without asking for additional hardware components. However, the detection of the user's face in the captured image and the recognition of the registered user's facial features need due deliberation.

### 4.1.1 Face Detection

Ali et al. [97] have conducted a detailed comparative analysis of various classical and modern detection methods. In light of their characteristics of near-real-time operations, the authors reviewed some of the most prominent for further discussion and analysis.

1. **Classical Methods** Traditional Rule-based and Appearance-based methods are extremely dependent on the conditions in which images are generated. Such methods are found to be less effective for variations in occlusions, illumination, and pose of the face. Classical methods are widely used, fast, and efficient for detecting faces with high accuracy and precision. However, these methods are sensitive to image variations and perform poorly in unconstrained environments. Some prominent methods are deliberated here.

- a) *Viola Jones*: The Viola-Jones algorithm is a popular face detection algorithm primarily designed to detect frontal face images with minimal variation in the pose. The detection pipeline calculates the integral image, Haar-like features, the Adaptive Boosting (AdaBoost) algorithm, and a cascade filter. Haar-like features are image features such as edge, line, and four-sided features useful for object detection tasks. The nose and eyebrow's horizontal and vertical features are useful for face detection. Calculating these features is computationally expensive, so it is calculated using integral images, where a pixel in the integral image is the sum of all pixels above and to the left in the original image. Once the calculation of Haar features is complete for the training data, it uses the AdaBoost algorithm to train a cascaded detector to enhance the model's speed and accuracy.

Cascading involves taking sub-windows of the image and searching for the best features in the sub-window. The sub-window is further explored for other features only if the best feature is found, saving computation time. Due to the abovementioned optimizations, the Viola-Jones algorithm is a high-speed and lightweight detector. Still, its accuracy is highly dependent on the orientation and pose of the faces in the image [98]. Adaptive Boosting, in turn, produces robust classifiers. Viola Jones has achieved detection with two frames per sec-

ond with a 95 percent detection rate. The Adaptive Boosting classifier has a false positive rate of one in 14084 samples.

- b) *Histogram of Oriented Gradients*: Histogram of Oriented Gradients (HOG) is a feature descriptor with significant usage in object detection for computer vision and image processing. It has a specific application emphasis on face detection in object detection by extension. A feature descriptor represents a picture that separates valuable information and dismisses irrelevant information, enabling the image to be further processed easily. The HOG method is based on counting the occurrence of gradient orientation and magnitude in a portion of an image. The feature description by HOG focuses on the structure and shape of the object. For every localized image region, histograms are generated using both the magnitude and orientation of the gradient.

The images are first preprocessed in the HOG method, and the horizontal and vertical gradients are calculated. At every pixel, the gradient has a magnitude and a direction. The images are then divided into cells, and a histogram of the gradients is calculated for each cell [99]. Supposing the image is divided into 8x8 cells. The information has the shape of 8x8x2 in each cell, with the magnitude and direction of the gradient at each pixel being stored. This vector with 128 numbers is converted into an array of 9 numbers representing angles 0, 20, 40, and up to 160 degrees. This step is followed by block normalization and forming a feature vector. These representations of faces can be learned by a computer and used to detect faces in photos. HOG-based face detection fails to perform well in cases where face photos are at odd angles. Therefore, there are better options for real-world face detection.

- c) *Local Binary Pattern*: Local Binary Pattern (LBP) feature descriptor operates by detecting objects and faces. The LBP operator works by labeling each pixel of an image with a binary number formed by thresholding the 3x3 neighborhood of the pixel, with itself being the central value. It is a straightforward and efficient texture operator with high discriminative power. It can extract a binary number with 8 digits, which can thus be extracted for each pixel. A texture descriptor, a 256-bin histogram consisting of labels generated across the



picture, can be applied with the bins marking different curving edges, spots, flat regions, etc.

A face picture may be considered a compilation of the micro-patterns mentioned above, which can be readily identifiable by the LBP feature descriptor. Face images are divided into several non-overlapping regions, with LBP histograms being calculated for each. These histograms are then concatenated, and the final histogram describes the local texture and global shape of the faces [100]. The feature vectors from these histograms can be used to train a classifier capable of detecting faces in images. The accuracy of this algorithm is highly dependent on the facial image's orientation and illumination conditions.

2. **CNN Based Modern Methods:** Convolutional Neural Networks (CNN) are a category of deep learning algorithms that find significant applications in object detection and recognition. They employ convolutional filters and max-pooling layers to learn complex high-level representations of the input images. The CNN approach for image processing offers much flexibility and a high degree of accuracy compared to the classical methods. Popular CNN architectures are frequently referred to for research and scientific applications, and their features are as follows [101].

- a) *LeNet*: This is designed to classify handwritten digits with 2 convolutional, 2 pooling, and 3 fully connected layers.
- b) *AlexNet*: This architecture adopts consecutive convolutional layers. 5 convolutional, 5 pooling and 3 fully connected layers are employed.
- c) *VGGNet*: This supports up to 19 layers by adopting a deeper network with a smaller convolution window.
- d) *GoogLeNet*: This uses 'inception modules' to employ multiple parallel convolutions, concatenated to decrease overall computational requirements.
- e) *ResNet*: Employs Residual connections, which increase the depth of the network as the residual connections reduce the gradient decay as layers are increased.
- f) *SENet*: This architecture uses block squeezes and excitations. It also recal-

brates channel-wise features by modeling interdependencies between the channels using 'Fire modules'. An improvement in ResNet has also been seen.

CNN-based face detection algorithms are accurate, fast, and highly robust [101]. State-of-the-art models like *You only look once (YOLO)*, *Multi-task Cascaded Convolutional Neural Networks (MTCNN)*, and *Single-shot detector (SSD)* can be employed for detecting faces in unconstrained environments.

- a) *YOLOFace*: YOLO is a state-of-the-art, real-time object detection system that uses one-stage object detection architecture [102]. YOLOFace is based on the YOLOv3 architecture. It uses a single neural network to predict the anchor boxes (predefined bounding boxes). YOLOv3 is the improved and optimized iteration of the YOLO architecture based on darknet-53 CNN. This architecture takes references from both ResNet and darknet-19, being more efficient than ResNet-101 and more powerful than darknet-19 [103]. The anchor boxes drawn around the detected faces in YOLOFace are slim boxes (height is larger than width), as opposed to YOLOv3, which are flat boxes. YOLO uses a multipart loss function. These are regression loss, confidence loss, classification loss, and no object loss. The weight for these losses in the YOLO framework is 1:1:1:1, but it is revised to 2:1:0.5:0.5 to make it more suitable for face detection. Data augmentation, namely saturation, brightness, and hue, was implemented during Model Training. Experimentation was done on the WIDER FACE and FDDB dataset, achieving a higher precision and recall than the YOLOv3 model and exhibiting a better Receiver Operator Characteristic (ROC) and Area under the ROC Curve (AUC) curve.
- b) *FaceSSD*: FaceSSD does face analysis without needing prior preprocessing, such as face registration. Even while identifying multiple faces, it achieves near real-time performance. It's a fully convolutional neural network (F-CNN), having exclusive pooling and convolutional layers [104]. It inherits parameters from the VGG16 network. The face analysis part of VGG16 is frozen, and the face detection part is fine-tuned with Annotated Facial Landmarks in the Wild (AFLW) dataset. Following this, the parameters from the face detection part are copied to the face analysis part, the face detection part is frozen, and the

face analysis part is fine-tuned on specific task-related datasets, keeping the detection part rigid. The face detection portion implements an objective loss function, the weighted sum of the face classification, and bounding box regression losses. Data augmentation approaches such as flipping, downsizing, cropping, and gamma correction are utilized in training. FaceSSD detects faces even while performing one or more tasks, like smile recognition and facial attribute prediction (like face analysis). The model was tested on high-resolution face images, exhibiting good precision-recall and ROC-AUC curves. As per the experiments conducted by Jang et al. [105], the average precision for FaceSSD is 99.88.

- c) *MTCNN*: MTCNN follows a three-stage approach toward face detection and alignment. An input image is resized to various scales and built into an image pyramid, which is the input for the three-stage pipeline. The first stage is a fully convolutional network called the Proposal network, which marks the bounding box regression vectors for the candidate windows and merges the highly overlapped candidates. In the second stage, all candidates are input to another CNN, the Refine Network, which filters out many false candidates. The third stage aims to describe more face details and outputs 5 facial landmark positions [106]. The filters for these networks are changed from 5x5 [source] to 3x3 as face detection is a complex binary problem requiring fewer filters but more discrimination. The training is composed of three main tasks with different loss functions. The Face classification task employs the cross-entropy loss function with the probability of the sample being a face. The second task, bounding box regression, uses an Euclidean loss function for predicting the offset between the candidate window and the ground realities. The third task is to mark the facial landmarks, which also employs the Euclidean loss function. The Face Detection Data Set and Benchmark (FDDB) dataset exhibits an area under the ROC curve of 0.9504. The WIDER FACE dataset shows an area under the precision-recall curve of 0.85, 0.82, and 0.67 for the easy, medium, and hard subsets.

### 4.1.2 CNN Based Face Recognition

The ability of CNN-based architectures to learn complex features in images makes them an excellent choice for calculating image representations and feature embeddings to be fed into a classifier for face recognition and verification. Three popular and highly accurate architectures explicitly designed for face recognition are mentioned below.

- a) *VGG-Face*: This is based on the widely popular VGG-Net architecture, which is trained on a database of 2.6 million face images corresponding to 2600 persons.
- b) *DeepFace*: Developed by Facebook, it is a 9-layer CNN with 120 million parameters. It is a state-of-the-art model but does not perform as well as FaceNet.
- c) *FaceNet*: Built by Google, it focuses on distance metric learning between positive and negative samples. The model has 11 convolutional layers and 3 fully connected layers for recognition. It employs Triplet loss, which is very suitable for face verification

### 4.1.3 Modular Approach for Face Recognition Process

Fernandes et al. [107] described five separate sequential components for a standardized framework for complete face recognition and verification for establishing identity. They are (i) image capture and preprocessing, (ii) detection of face, (iii) extraction of features, (iv) selection of features, and (v) matching of features.

Various research aspects related to Frame difference in captured frames, face detection, and Face recognition have been studied and analyzed as listed below.

1. *Frame Capturing*: The proposal by Valter et al. [108] uses transformation from the RGB colour space to two different colour spaces (YCrCb and CIE L\*u\*v). Histograms are calculated based on the images in these transformed colour spaces and concatenated to serve as input for a tree-based classifier. Transformation to two

different colour spaces makes the input images more suitable for spoofing detection by providing greater separation between luminance and chrominance information.

2. *Frame Difference*: Frame differencing refers to checking the difference between two consecutive video frames. By calculating the similarity score between two frames, frame differencing is an efficient and computationally light method. It reduces the computational load in the proposed facial recognition modular design, as the more computationally expensive steps, such as detection and recognition, are taken only when there is a significant difference in the frames. Susmitha et al. [109] proposed that Mean Square Error (MSE) and Structural Similarity Index Matrices (SSIM) are two commonly used methods to calculate the difference between frames. SSIM score is 1 when the consecutive frames are identical, with a lower SSIM score indicating more difference between the frames. Thus, a threshold can be set such that an SSIM score below the threshold for two consecutive frames indicates a notable difference between them. SSIM score outperforms simple MSE calculation with a negligible increase in computational requirements, so it is chosen as a promising technique for frame differencing.

3. *Face Detection*: Facial detection refers to identifying a human face in a given image. It is a very old technique with rich applications in the modern world, the most common being the autofocus mechanisms used in modern cameras. Many well-developed and documented algorithms employ various mathematical and machine-learning models to identify faces from a given image. The most popular algorithms include LBHP (Local Binary Patterns Histograms), Fisherfaces, etc. Following the scope and objective of this project, instead of searching for the most efficient algorithms, the literature survey focused on collecting the best models developed based on these algorithms.

Various factors were considered while shortlisting the models for further face-to-face comparisons.

- (i) Models should be as lightweight as possible so that a live video feed can be processed in real-time on edge devices.
- (ii) Models should be based on well-researched algorithms.

(iii) Models must be well documented and maintained, preferably available in popular mainstream libraries.

4. *Face recognition*: Face recognition refers to identifying a human face from an image and attributing it to a person from the database. It can be termed as the 1: N problem of identifying a person from N individuals based on the facial features captured. Face verification refers to the 1:1 problem of determining whether or not an image belongs to a particular person. CNN-based techniques have emerged as a good solution for facial recognition and verification tasks due to the ability of convolutional neural networks to understand and extract high-level features from images, which are instrumental in recognizing and identifying human faces. CNN-based facial recognition models tend to have a larger training and prediction time than their non-neural network counterparts, especially without a dedicated GPU card that takes advantage of parallel processing.

The most popular CNN models have been analyzed in detail based on the following criteria.

- (i) Accuracy of prediction.
- (ii) Speed of prediction.
- (iii) Availability of open-source model and weights.
- (iv) Performance on Standard databases like LFW, YTF, CASIA WebFace.

### 4.1.4 Liveness Check & Anti Spoofing Check

The Liveness check or anti-spoofing check is performed with the help of a skeleton code provided in [110], which takes advantage of the DNN module of OpenCV to load a robust anti-spoofing algorithm to check for spoofing. Models based on PyTorch, Keras, and Tensorflow backends can be plugged into the skeleton code to check for anti-spoofing. An OpenCV-based interface for taking in a live video feed and capturing frames from the video has been added to the liveness. CNN models for prevention against replay and print-based attacks are implemented using the backbone, as mentioned in the earlier code.

4.1.4.1 Detailed Analysis

1. Face Detection Methods predating deep learning techniques, such as the VJ and sliding window methods in general, exhibit high accuracy in cases where the face image has been captured in a controlled environment with good pose and illumination. Deep Learning Algorithms for face detection, such as YOLOFace, MTCNN, and Single Shot Detectors (SSD), exhibit more flexibility regarding changes in illumination, pose variation, and scale of the face images.

Table 4.1 compares the eight CNN-based models with the VJ method and lists the advantages and disadvantages of each. The discussion mentioned above on CNN-

**Table 4.1:** Comparative Analysis of Face Detection Models

Model	Advantages	Disadvantages
OpenCV Haarcascade [111]	Very lightweight, capable of running real-time	Highly prone to false-positive detections, therefore has a low accuracy
Viola Jones (Non-CNN) [98]	Fast detection. Simple implementation. No resizing of images required	most effective with a frontal view. sensitive to illumination changes.
OpenCV DNN (Deep Neural Network) [112]	Accurate face detector, Utilizes modern deep learning algorithms	May have unconscious biases in the training set
Dlib HOG + Linear SVM face detector [113]	More accurate than Haar cascades, Extremely well documented	Only works on frontal views of the Face, Not as accurate as deep learning-based face detectors
Dlib CNN face detector [114]	Very accurate model, Expertly implemented and documented	Cannot run in real-time without GPU acceleration
YOLOFace [103]	Most precise of the three CNN models. Best performance than the others on Video Data.	Slower than FaceSSD
FaceSSD [115]	Fastest of the three CNN Based Methods.	Shows poor performance on smaller objects and unconstrained environments.
MTCNN [116]	Provides key points for face alignment.	Slower than FaceSSD. Significantly slower than YOLOFace while using GPU

based Face Detection systems shows that these are the preferred choice for an un-

## 4 Face Recognition for User Authentication of Cloud Based Systems

constrained environment and are more robust and efficient than non-CNN-based methods.

2. Face Recognition Table 4.2 compares the five popular CNN-based models for Face Recognition and lists the advantages and disadvantages of each.

**Table 4.2:** Comparative Analysis of Face Recognition CNN Models

Model	Advantages	Disadvantages
FaceNet [117]	Uses Inception Modules to decrease the number of trainable parameters. Better performance than Deep Face Method. Robust to variations in input image of the three, such as expressions, illumination, and pose. Well suited for verification task due to Triplet loss.	Extremely Deep Network, hence a pre-trained model is preferred. Faces some difficulty with occlusion.
DeepFace [118]	Most state-of-the-art model of its time (2014) Well suited for identification task since the softmax layer is the last layer.	Based on 3D modeling of face, adding to computational overhead.
VGGFace [119]	Best Performance in the case of occlusions without altering the weights or retraining the model. Well suited for verification task due to triplet loss.	Very Heavy Model requiring about 550 MB of Memory. Very Slow Calculation time Slower CPU training time
OpenFace [120]	OpenFace is heavily inspired by the FaceNet project, but this is more lightweight, and its license type is more flexible	Less accurate
Dlib Face Recognition [121]	This model achieved an accuracy of 99.38% on the standard LFW face recognition benchmark for face recognition.	The dlib library is originally written in C++.

### 4.2 Chapter Summary

After analyzing the various factors of user authentication and establishing that face recognition is a potential way to implement MFA without requiring additional hardware, details of modern face recognition systems were explored. The preceding analysis of modern Facial Recognition Techniques and explanation describes the potential and optimal



## 4 Face Recognition for User Authentication of Cloud Based Systems

---

applicability of the MFA factor as part of an authentication system. Using "*something the user is*" towards facial biometrics, these modern approaches are regarded as potential means for cloud user authentication in the form of MFA. The aspect of "*something the device is*" would be analyzed subsequently concerning devices of the CoT environment. The following chapter discusses the secure authentication of users in a multi-cloud environment.

# 5 Secure Authentication of Users In Multi-Cloud Environment

## 5.1 Introduction

Improper authentication may lead to severe consequences for data security, including loss of confidential information, financial loss, and damage to reputation. Individuals and organizations must implement strong authentication measures to protect against these risks. Every user's role is becoming crucial in the entire ecosystem of Information Communication Technology (ICT) since all organizations are moving into the cloud due to the inherent advantages of cloud computing systems. Mostly, user authentication methods and identity-providing systems rely on a centralized approach where credential information storage is always under the threat of a *Single Point of Failure* (SPF) from a security perspective. Hence, a simple failure will likely affect many users' digital identities. Another prominent issue with the centralized approach is that in the centralized systems, individual users do not have control over how much of their private and Personal Identifying Information (PII) is shared in different contexts in different applications or otherwise.

Improper and inadequately strong authentication measures in a cloud environment can severely affect the system's security. Some potential risks like Compromised Infrastructure, Compromised User Accounts, Data Breaches, Denial of Service (DoS) Attacks, and Malware Attacks are the causes of making a cloud system vulnerable. To mitigate these risks, Cloud Service Providers (CSPs) and users must implement strong authentication measures, such as multi-factor authentication, effective password policies, and biometric verification, to ensure that only authorized users can access cloud resources. As a general practice, once a user is successfully authenticated to a system, there are no further checks

until the end of the logged-in session, or the user logs out of the system. However, regular security assessments and audits should be conducted to identify vulnerabilities associated with user authentication and address them promptly.

It is noticed that the COVID-19 pandemic has accelerated the trend toward remote work, which has created new security challenges. With more employees working from home or other remote locations, multiple times authentication over the logged-in session can help prevent unauthorized access and protect against data breaches. Academic universities have also resorted to online education and online examinations and evaluations of students' grades. This case also merits a user authentication solution with a repetitive functionality that must be enforced to check and ensure the presence of only the desired user accessing the system for high security and sensitive applications like online examination. Multiple-time authentication can help prevent unauthorized access and protect against data breaches by providing real-time threat detection and proactive security. Similarly, organizations have resorted to manual or semi-automatic proctoring methods to deal with such proctored examinations and handle user authentications for susceptible applications. Hence, the design and development of an automatic, seamless, proctored authentication scheme is becoming the need of the hour. Association of Computing Machinery (ACM) in [122] has highlighted the desired characteristics, including its security features & requirements for a robust and strong proctoring system ensuring user privacy. This research explores modern approaches to face recognition for user authentication.

Multi-factor Authentication (MFA) has been accessed as a more challenging option for any attacker to break into a secure system through an unauthorized entry by either breaking the account credential protection mechanism or bypassing the authentication factor. Using more than one type of authentication factor in a cascaded manner invariably increases the deterrence level for the attacker. To avoid adding further complexity to cloud computing and its Identity and Access Management (IAM) system, this research endeavor would achieve the desired MFA functionality without requiring additional or specialized hardware or software for implementing MFA towards secure IAM functionality in a multi-cloud environment.

Cloud computing is making its presence felt in all spheres of IT operations due to its inherent advantages associated with this technology. However, it comes at a cost regard-

ing computing complexity related to user *Identity and Access Management* IAM for the cloud. The CSPs extend their services to the clients, called *Tenants* with their respective IAM mechanisms. However, it is vulnerable to suffering the effects of SPF. Hence, a decentralized IAM solution is the need of the hour, which should have the functionality of protecting users' privacy in terms of safeguarding the PII's associated with the users.

Over time, many instances of spoofing attacks and impersonation attacks on secure ICT systems have come to light. As a mitigation measure, the need for repetitive authentication arises because once a user has successfully authenticated, they may still pose a security threat if their credentials or access privileges are later compromised. For example, an attacker may access a user's account by stealing their login credentials or impersonating them using stolen biometric data.

### 5.2 Background Literature Study

In today's digital age, cloud computing has become essential to our daily lives. However, a robust and secure identity authentication mechanism has become more crucial with the increasing use of cloud services. Centralized identity authentication systems, which are currently prevalent, are vulnerable to data breaches and cyber-attacks [123], which can result in severe consequences for the user's privacy and security.

Decentralized cloud identity authentication has emerged as a promising solution to address this issue [124]. Unlike centralized systems, decentralized identity authentication allows users to control their personal information and grants them greater privacy control functionality. Users can choose which personal data they want to share with service providers. Their identity is verified using a more secure, tamper-proof, decentralized, blockchain-based mechanism.

By providing users with greater privacy control and security, decentralized cloud identity authentication can significantly enhance the overall security and privacy of cloud computing, making it more trustworthy and reliable. This technology can ensure that users have more control over their data and identity, ultimately leading to a safer and more secure cloud environment.

Repetitive authentication helps to protect against various types of attacks, including brute-force attacks, password-guessing attacks, and phishing attacks, which can compromise the security of a system. By requiring multiple authentication factors, the system can provide an additional layer of protection and increase its overall security. Nevertheless, the system of challenge-response mechanism follows an authentication procedure by employing a factor of authentication discussed in Chapter 3. It is observed that a user repetitively *responding to challenges* of the authentication system creates a hindrance to the seamless and uninterrupted operation of a successful logged-in user while continuing in an actively logged-in user session.

Dynamic authentication is a security measure randomly requesting additional authentication from a user during a session, even after successful authentication. By adding a layer of security, random authentication can reduce the risk of unauthorized access and protect against data breaches [125]. Seamless random authentication can also provide a better user experience than other forms of authentication or frequent re-authentication.

Continuous authentication is a security measure that involves continuously monitoring a user's behavior and activities to detect any unusual or suspicious activities that may indicate an unauthorized user or an account takeover [126]. The need for continuous authentication arises from traditional authentication methods, such as usernames and passwords, which are no longer sufficient to protect against the growing number of cyber-attacks and security breaches. Seamless continuous authentication is also intended to provide a better user experience than other forms of authentication or frequent re-authentication.

### 5.2.1 Traditional Approach

Traditional approaches in cloud authentication usually involve using usernames and passwords, as well as other authentication factors such as smart cards, biometric authentication, and token-based authentication. Details of their drawbacks are discussed in Table 5.1 for progressing our research.

If the requirement of specialized hardware can be overcome, it can prove advantageous for the unique identification of a user with unique characteristics of the respective cloud user. In addition to these traditional approaches, MFA is becoming increasingly popular

**Table 5.1:** Limitations of Traditional Authentication Approaches

Approach	Functionality	Limitations	Ref
Password Approach (Knowledge)	Users are required to enter a unique username and password combination to access cloud resources. The username is usually the user's email address or a unique identifier, while the password is a secret code that the user creates and uses to authenticate themselves.	Passwords can be easily guessed, stolen, or cracked by attackers, leaving cloud resources vulnerable to unauthorized access.	[127]
Smart-Card or Token-based Approach (Possession)	The smart card or token, with an embedded microchip containing authentication information, is inserted into a card or USB reader, which reads the information on the card or chip and verifies user identity.	Implementing smart cards are costly and complex, requiring additional hardware and software. It can be lost or stolen	[128]
Biometric Approach (characteristics)	Use of unique physical characteristics of the user, such as fingerprints, facial recognition, or iris scans, to authenticate them.	Biometric data is spoofable and cannot be recovered once compromised. It requires specialized hardware and software.	[129]
Geo-tagged Approach (Location)	Use of Global Positioning System (GPS) signature or static IP address are used to authenticate users.	Loss of GPS signal or network failures makes this useless.	[20]

in cloud authentication mechanisms. MFA involves using multiple authentication factors, such as a combination of a password and a biometric factor, to increase security and prevent unauthorized access. Above all, an MFA approach with a decentralized database in the background would facilitate a secure and robust authentication means for a cloud environment.

### 5.2.2 Self-Sovereign Approach

*Self-Sovereign identity (SSI)* is a decentralized identity management approach that allows individuals to own and control their digital identity without needing a centralized authority. The digital identity that is sovereign, decentralized, enduring, and portable

is designed for real-world entities. It allows the owner to securely access various digital services while protecting their privacy and giving them control over the management of their identity [130]. The SSI approach for cloud authentication can provide greater privacy, security, and user control while improving interoperability, trust, and efficiency. The main players of a standard SSI system are identity- *Issuer, Holder, and Verifier*, and their interoperability is depicted in Figure 5.1.

An issuer is an entity that issues credentials to holders who own an identity and receive desired credentials from the issuer. These credentials are held in a digital wallet and presented to a verifier for verification purposes. The verifier is typically a service provider that requests and verifies credentials using a trust relationship facilitated by a blockchain. SSI has three key components: a blockchain or distributed ledger, a Decentralized Identifier (DID), and Verifiable Credentials (VC). The blockchain is used to establish trust without needing a third party, while the DID is a unique identifier linked to an identity created independently and owned entirely by the identity owner. Verifiable Credentials are tamper-evident and privacy-preserving credentials issued by an issuer and are linked to the identity owner's DID. The issuer's digital signature can be verified for validity and verified through their public DID on the blockchain.

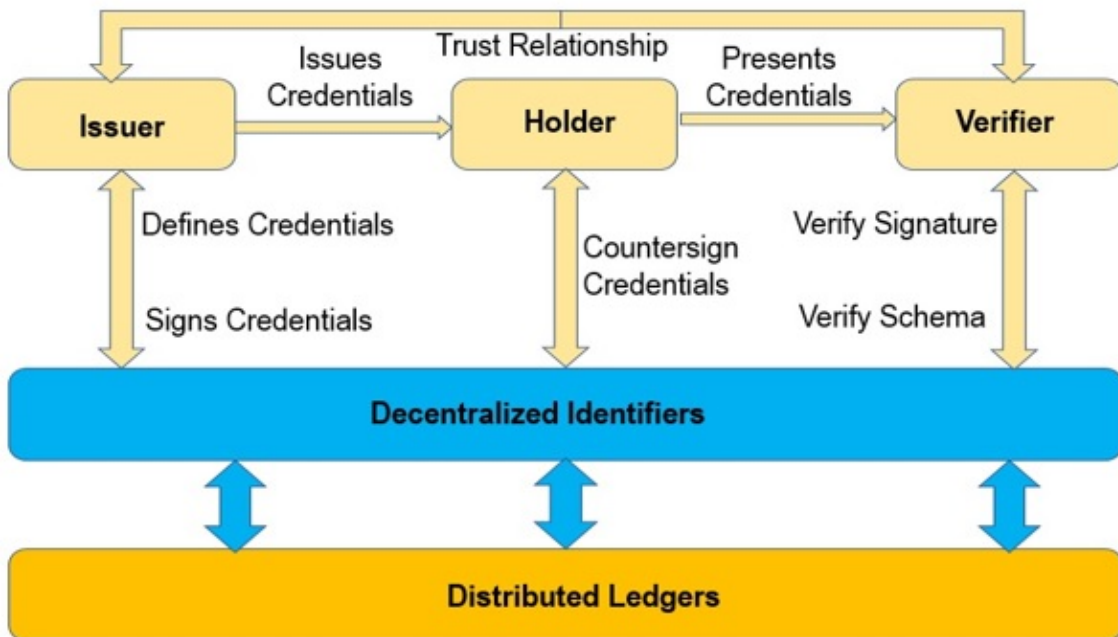
The advantages make SSI an increasingly popular approach for digital identity management and cloud authentication. SSI can provide several benefits for cloud authentication, depicted in Table 5.2.

### 5.2.2.1 Sovrin Networking

The Sovrin Network, operated by a consortium called Sovrin Foundation, is based on a public permissioned ledger, which means that unlike in a public permissionless blockchain, anyone cannot operate the nodes. Instead, trusted institutions called Stewards are the only ones allowed to operate nodes while adhering to the *Sovrin Governance Framework (SGF)*. Through this secure system, identity users can confidently publish their identity, transfer their credentials, sign transactions, and control their keys and data in a peer-to-peer model. The Sovrin blockchain serves as a platform for identity verification, and it operates under a web of trust model that eliminates the need for centralized Certificate Authorities (CAs)

**Table 5.2:** Advantages of SSI Authentication Approach

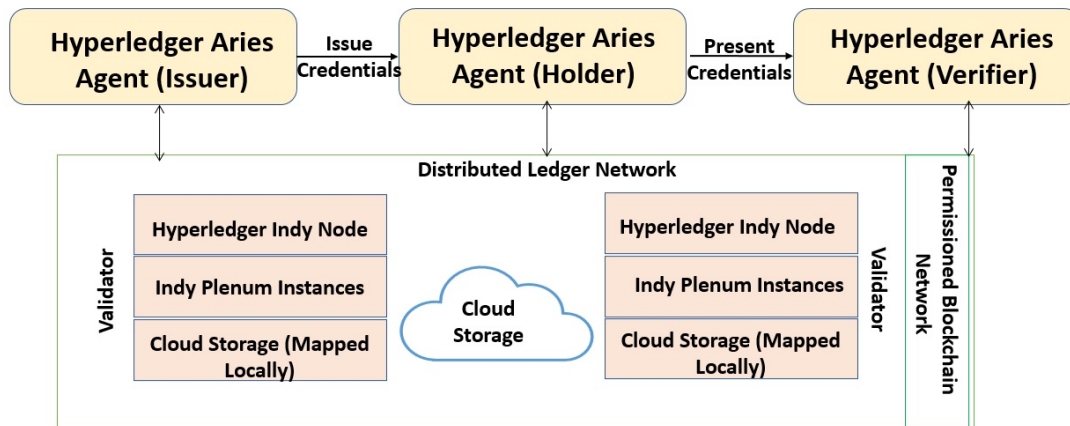
Ser No	Characteristics	Advantages	Ref
1.	Efficiency	SSI can improve efficiency by reducing the need for manual identity verification processes, such as in-person verification or document review.	[131]
2.	Interoperability	SSI enables interoperability between different cloud providers and services, as users can use the same identity across multiple providers without needing to create separate accounts for each one.	[132]
3.	Privacy and Security	SSI can enhance privacy and security for cloud authentication by enabling users to control their own identity data and share only the minimum required information with cloud providers. This reduces the risk of data breaches and identity theft.	[133]
4.	Trust and Verifiability	SSI enables trust and verifiability by allowing users to provide verifiable proof of their identity without revealing unnecessary information. This can reduce the risk of fraud and increase confidence in cloud authentication.	[134]
5.	User Control	With SSI, users have complete control over their digital identity and can choose what information to share with cloud providers. This puts users in charge of their own identity rather than relying on a centralized authority.	[133]



**Figure 5.1:** Functioning of SSI Ecosystem



to provide digital trust. The Sovrin SSI model is adaptable to distributed ledgers that satisfy specific criteria and rely on verifiable claims to establish trust.



**Figure 5.2:** Hyperledger on Cloud for SSI Ecosystem

The Sovrin Foundation manages the Sovrin Network, an open-source framework designed to offer users a decentralized and sovereign digital identity. The framework is based on the Hyperledger Indy platform, which provides a comprehensive set of specifications, design patterns, and terminology for developing decentralized digital identity solutions. The Sovrin Network is a specific implementation of Hyperledger Indy, which is a distributed ledger that is publicly accessible but requires permission to operate. The framework uses other Hyperledger libraries, such as Aries and Ursa, which enable the provision of verifiable digital credentials and a shared cryptographic library, respectively, as depicted in Figure 5.2. Only trusted institutions known as Stewards are authorized to operate nodes and participate in the consensus process, following the SGF. Using the Sovrin Network, individuals can securely publish their identity, transfer credentials, sign transactions, and manage their keys and data in a secure peer-to-peer model. All identity-related operations within the Sovrin Network are governed by the SGF, developed by the *Sovrin Governance Framework Working Group (SGFWG)* [135].

### 5.2.3 Incorporation of MFA to SSI

SSI refers to a commonly accepted structure in digital identity that empowers individuals with authority over their personal data and digital identity. SSI is a decentralized digital identity model that enables individuals to own and control their identity without intermediaries. MFA is a security measure that requires users to provide two or more forms of authentication to verify their identity. SSI can incorporate MFA to enhance security and protect user identity. SSI can incorporate MFA through biometric authentication, one-time passwords (OTPs), or hardware-based tokens to strengthen security and protect user identity.

One way SSI can incorporate MFA is through the use of biometric authentication. Biometric authentication relies on unique physical characteristics, such as fingerprints or facial recognition, to verify a user's identity. SSI can incorporate biometric authentication by requiring users to provide a biometric sample, such as a fingerprint or facial scan, in addition to their SSI credentials. Another way SSI can incorporate MFA is through the use of one-time passwords (OTPs). OTPs are codes valid for a single login session and generated by a device or application. SSI can incorporate OTPs by requiring users to provide a unique code and SSI credentials. SSI can also include MFA through the use of hardware-based tokens. Hardware tokens are physical devices that generate a unique code valid for a single login session. SSI can incorporate hardware tokens by requiring users to have a token in their possession in addition to their SSI credentials.

In our research journey, we inferred that MFA is a user authentication security enhancer, as each user can uniquely identify with biometric characteristics. Similarly, a distributed approach for identification and user authentication addresses the problem of one point of failure. To have a technically sound amalgamation of all the above three aspects, we further explore and orient our research direction.

### 5.2.4 Factors for Re-Authentication

A traditional approach like *Challenge Response* mechanism is disturbing and distracting to repeatedly ask a logged-in user in an active session to prove user authenticity. While

**Table 5.3:** Suitability of Factors for Re-Authentication

Factors for Authentication User	Re-of	Required Extra Hardware and Software	User Disturbed or Distracted
Captcha		No	Yes
Fingerprint Scanner		Yes	Yes
Location Geotagging		Yes	Yes
Ocular Based Method		Yes	Yes
OTP		Yes	Yes
PIN		Yes	Yes
Retina Recognition		Yes	Yes
Smartphone Application		Yes	Yes
Smart Card		Yes	Yes
Thermal Imagery		Yes	Yes
Vein Recognition		Yes	Yes
Voice Recognition		Yes	Yes
Face Recognition		No	No

using the standard classical approach of "what the user knows", "what the user has", "what the user is" or "where the user is", it would entail the requirement and use of additional or specialized hardware and software. However, as concluded in Chapter 3, using a fitted webcam would be a feasible option to achieve user re-authentication without distracting or disturbing the logged-in user during an active session. This way, we can avoid asking the user to remember and apply a password to prove "what the user knows". Also, the user is not disturbed by any kind of pop-up message to read and enter OTP to re-authenticate. Similarly, the user would not be disturbed to use the biometric fingerprint reader to re-authenticate and prove "what the user is". Rather, the user must not be prompted to show the user's face to a biometric camera or retina reader during an active session. As such, reading the user's location from the system would not disturb the user but would entail additional GPS hardware circuitry and associated software to prove " where the user is". An analysis of the suitability of factors for re-authentication (within our objective of not requiring additional hardware & software) is presented in Table 5.3.

Table 5.3 infers that *Face Recognition* can be used as a re-authentication factor for users

without requiring additional hardware or specialized software. It is also expected not to disturb or distract the user while logged in and during an active session. Hence, we explore further details of the usage of webcams for seamless re-authentication of users during an actively logged-in session without any pop-up message to disturb the user, like showing the face to camera for face or iris biometrics. Instead, our further research is about exploring and efficiently using the Original Equipment Manufacturer (EM) fitted webcam for user facial biometric re-verification for repetitive re-authentication.

The preceding comparative analysis and explanation describe various MFA factors and their effective applicability on the authentication system without disturbing the actively logged-on user or device and requiring additional hardware and software. **This meets the defined Objectives mentioned at Chapter1, Objective 2 and Sub-Objectives 2b.**

### 5.3 Research Gaps and Questions

The requirement of specialized hardware and software is an obstacle to adopting biometric-based authentication as a part of MFA. The implementation of *distributed ledger (DL)* approach has experimented in the form of blockchain. In the second or new generation of blockchains, interfacing with user browsers or applications is usually done by deploying smart contracts over the blockchain. *Digital Identifiers(DIDs)* have been promulgated by the World Wide Web Consortium (W3C) to implement and interface with DL in a secure environment. Implementing a Blockchain or DL over a Cloud Infrastructure would enable the deployment of one or multiple Smart-Contract beneath the Blockchain for MFA implementation for user authentication. Research in biometric face recognition has seen many advancements concerning image interpretation, face detection, and face recognition. This research has established the suitability of face recognition without requiring specialized hardware and software to incorporate it as a part of MFA. The possibility of interfacing DIDs to DL for MFA applications is seen as a definite possibility for enhancing the security posture of the user SSI authentication mechanism.

Research in the field of biometric face recognition has seen many advancements [136]. CNN-based face recognition is being explored to handle the drawbacks of classical methods

effectively. The possibility of interfacing DIDs to DL for MFA applications is seen as a definite possibility for enhancing the security posture of the user SSI authentication mechanism.

From the above discussion, the following research questions have been formulated to be answered in this research.

RQ5.1. How would biometric factors be effective in handling the MFA system for users?

RQ5.2. How can biometric user authentication in a repetitive mode in MFA be used for securing user authentication in multi-cloud?

RQ5.3. How do we deploy multiple Smart-Contract in an integrated manner over a Blockchain for IAM functionality, keeping security requirements and design goals in mind?

RQ5.4. How can we implement an efficient and continuous MFA for enhanced security towards user authentication in the multi-cloud scenario?

### 5.4 Proposed Approach of Multi-Factor Authentication

Considering a system that uses MFA with N factors, each factor is denoted by a random variable  $X_i$ , which takes values 0 or 1. The value 1 represents a successful authentication for the i-th factor, while 0 represents a failed authentication. We can assume that each  $X_i$  is a Bernoulli random variable with parameter  $p_i$ , representing the probability of a successful authentication for the i-th factor. The overall probability of successful authentication for the MFA system can be modeled using Equation 5.1.

$$P(\text{AuthenticationSuccessful}) = P(X_1 = 1, X_2 = 1, \dots, X_N = 1) \quad (5.1)$$

This equation represents the joint probability of all N factors being successfully authenticated. Assuming that the authentication factors are independent, we can use the product rule of probability to expand Equation 5.1 as Equation 5.2:

$$P(\text{AuthenticationSuccessful}) = P(X_1 = 1) * P(X_2 = 1) * \dots * P(X_N = 1) \quad (5.2)$$

This equation represents the probability of successful authentication as the product of the probabilities of successful authentication for each factor.

We can use various models depending on the specific authentication method to calculate the probability of successful authentication for each factor. For example, we can use a statistical model for password authentication or a cryptographic model for token-based authentication. In this research, we prefer using cryptographic means for added security in the overall MFA process.

### 5.4.1 Use of Biometric

Since it uses users' unique characteristics, biometric authentication can be highly secure. However, it can also be subject to spoofing attacks. Similarly, malicious insiders (users) can pose threats for impersonation attacks. It can also be triggered for the negligent sharing of the respective user credentials among the users.

Biometric authentication systems can incorporate various techniques, as listed, to handle spoofing attacks:

1. *MFA*: Biometric authentication can be combined with other forms of authentication, such as a password or a security token, to increase the system's overall security.
2. *Liveness Detection*: This technique uses advanced algorithms to ensure that the biometric data presented to the system is from a live person and not a static image or a recorded video.

Similarly, To handle impersonation attacks, biometric authentication systems can use techniques such as:

1. *Identity Verification*: This involves verifying a user's identity against a trusted identity database, such as a government-issued ID, to ensure they are who they claim to be.
2. *Repeatative Authentication*: This technique involves monitoring the user throughout

the authentication process, using machine learning algorithms to detect anomalies or suspicious behavior.

3. *Multi-modal biometric authentication*: This involves using multiple biometric modalities, such as fingerprints, facial recognition, and voice recognition, to increase the accuracy of the authentication process and make it harder for attackers to impersonate a user.

Incorporating these techniques can help increase the system's overall security and mitigate the risks associated with these types of attacks. Within the declared limitations of the scope of this research, features like repetitive re-authentication as well as liveness check features related to facial recognition are further explored. **This answers the Question No 5.3.**

### 5.4.2 Dynamic MFA (D-MFA)

Spoofing attack is a known concern with facial recognition. It is needed to ensure that it only authenticates the trusted individual user and that they are active for the whole active system log-on session. To deal with this impersonation attack, we propose to use the approach of Dynamic-MFA (D-MFA) authentication, as depicted in Equation 5.3

$$F(x) = k(a) \bigcap \sum_{t=0}^{t=rand} l(b) \bigcap \sum_{t=0}^{t=rand} m(c) \quad (5.3)$$

$F(x)$  is Multi-factor System Authentication Function over user  $x$  with a resultant Boolean Value.  $k(a)$  is the First Authentication Factor Function with input value  $a_1$  and so on.  $l(b)$  is the Second Authentication Factor Function with the input value  $b_1$  and so on.  $m(c)$  is the Second Authentication Factor Function with the input value  $c_1$  and so on, for its applicability in general, to third and subsequent factors for authentication.

It is the process of authenticating users frequently and dynamically during live sessions. The dynamic authentication system would also check for the liveness factor during each image recognition step, which in turn verifies the authenticity of the user. The system architecture of the D-MFA System is depicted in Figure 5.3.

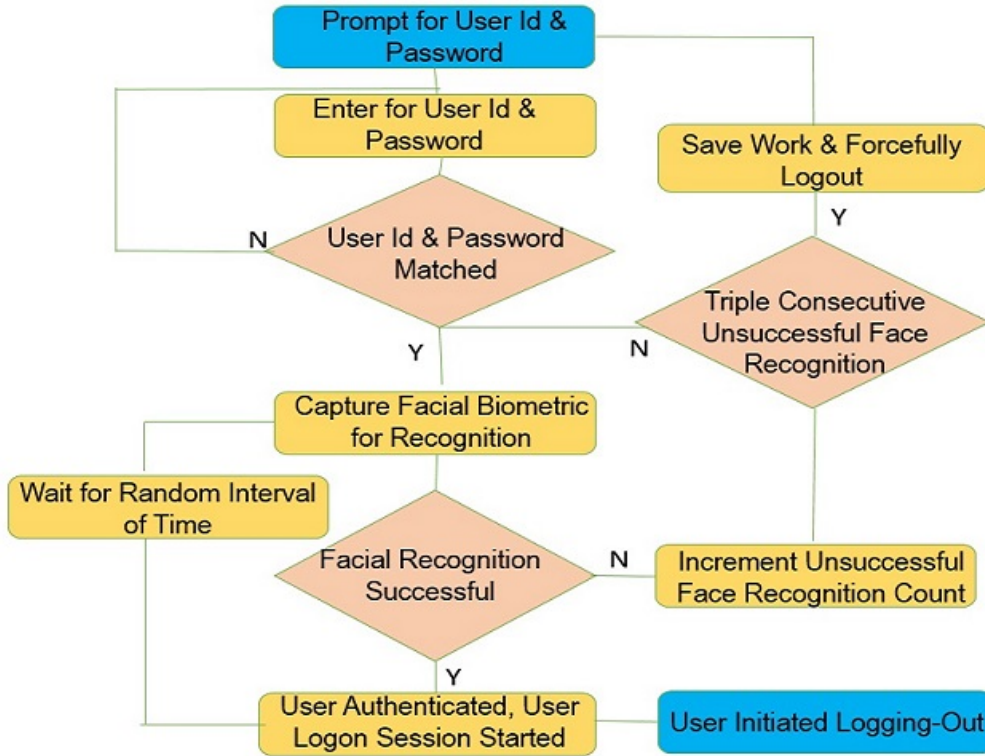


Figure 5.3: System Architecture of D-MFA

### 5.4.3 Continous MFA (C-MFA)

Impersonation attacks, which fall under phishing attacks, are a known concern with facial recognition. It is needed to ensure that it only authenticates the trusted individual user and that they are active for the whole active system log-on session. To deal with this impersonation attack, we propose using the Continous-MFA (C-MFA) authentication approach, as depicted in Equation 5.4. The uniqueness of this face recognition-based MFA is that the face recognition operation is modular. Depending on the need for its function, a particular module like *Frame Difference*, *Face detection*, *Face Recognition* operations, as depicted in Figure 5.4 are performed without disturbing the logged in user in any manner.

$$F(x) = k(a) \cap \sum_{t=0}^{t=\infty} l(b) \cap \sum_{t=0}^{t=\infty} m(c) \quad (5.4)$$

$F(x)$  is Multi-factor System Authentication Function over user  $x$  with a resultant Boolean Value.  $k(a)$  is the First Authentication Factor Function with input value  $a_1$  and so on.  $l(b)$  is the Second Authentication Factor Function with the input value  $b_1$  and so on.  $m(c)$  is the Second Authentication Factor Function with the input value  $c_1$  and so on, for its



applicability in general, to third and subsequent factors for authentication.

The research conducted by [120], [121], [137], and [138] aims to design architectures that

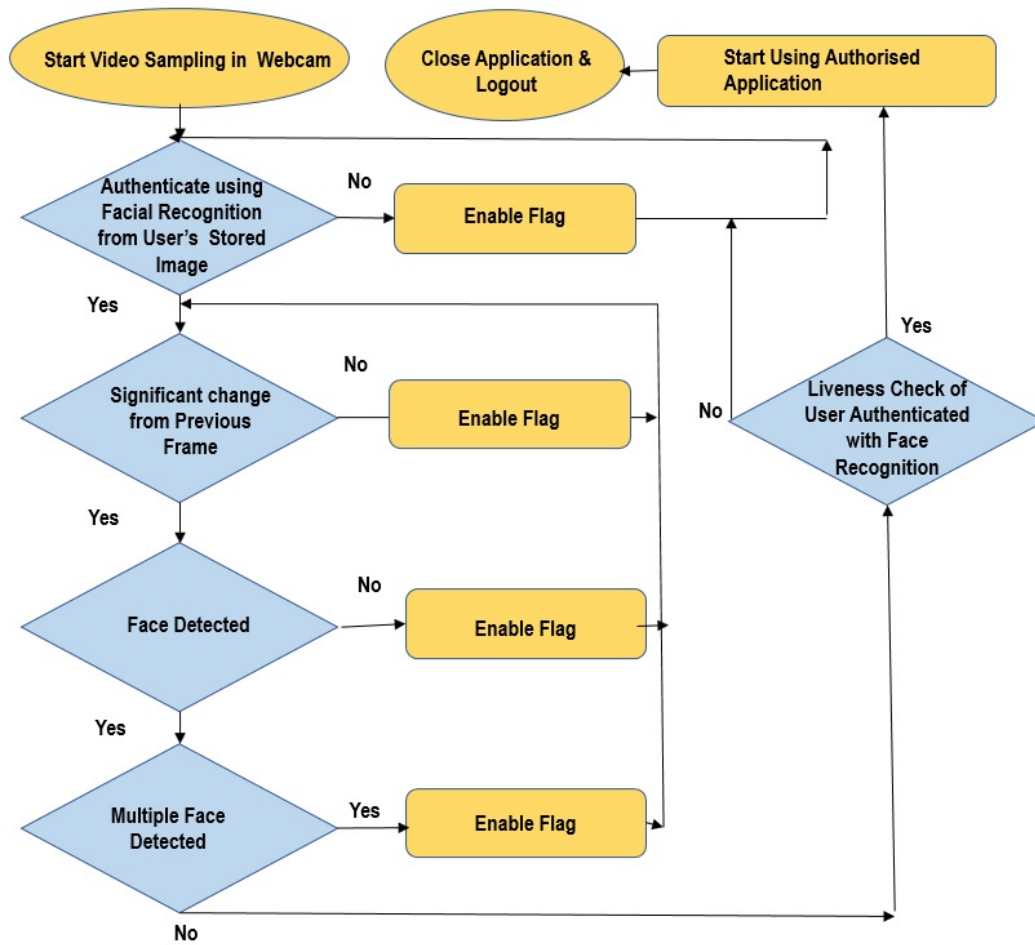


Figure 5.4: System Architecture of C-MFA

are lightweight and accurate for face verification in unconstrained environments, with a special emphasis on speed of detection and verification.

The above explanation describes the usage of biometric user authentication in a repetitive mode in MFA, which can be used for secure user authentication in multi-cloud. **This answers the Research Question 5.3 mentioned in Chapter4.**

Table 5.4 compares these architectures, stating the distinct advantages and disadvantages. In this research, it is proposed that a permissioned blockchain-based MFA implementation uses OEM-fitted webcam for user recognition with facial biometrics. The proposed approach is DID-enabled, integrating multiple smart contracts for interfacing with the DL of the Blockchain network.

**Table 5.4:** Comparative Analysis of Optimised Face Verification Models

Model	Advantages	Disadvantages
FaceNet+ [120]	Shows good performance in unconstrained conditions	Performance of Facenet models are not as good as VGG-based models.
2-tier Face [121]	Lighter Mode with 9.2M parameters Faster speed for positive case	More computation on Edge device Private training database
Face verification without alignment [137]	Sift Descriptors offers robustness to rotation.	2 weeks training time with GPU.
MixFaceNets [138]	Extremely Fast and Lightweight model (1 to 3M parameters depending on the version)	Less focus on robustness to unconstrained settings

### 5.4.4 Selection & Implementation of Blockchain

Second-generation blockchain is chosen to take advantage of the inherent security aspects of the blockchain and for effective interfacing with the DL in the dynamic environment. Many organizations have progressed towards implementing identity solutions, namely Sovrin, uPort, OLYMPUS, SelfKey, Blockstack, Civic, ShoCard, lifeID, and MultiChain. We preferred choosing an open-source solution that has been well documented and the availability of research papers and white papers on the particular solution approach. Opensource implementation of second-generation blockchain is done using Hyperledger and is well documented. As a stable solution, it is incorporated into our present research work. Different flavors of Hyperledger customized for specific operations are also proposed to be used, as depicted in Figure 5.2.

### 5.4.5 Interfacing with DL for MFA System

Interfacing with a distributed ledger for implementing Multi-Factor Authentication (MFA) can be done through various mechanisms depending on the specific distributed ledger technology being used. One common approach is using a blockchain-based identity management system incorporating MFA functionality. In this system, the distributed ledger

is used to store and manage identity data, such as public keys and authentication tokens, while MFA provides an additional layer of security for identity verification.

To implement MFA using a DL and DIDs, users may be required to provide multiple forms of authentication, such as a password or PIN, a fingerprint or face scan, and a cryptographic key or token. This information is then stored on the distributed ledger and can be used to verify the user's identity when they attempt to access a resource or service.

Another approach is to use a smart contract-based system that incorporates MFA functionality. In this system, the distributed ledger executes and enforces the MFA rules and policies specified in the smart contract. The smart contract can be programmed to require multiple forms of authentication, such as a password and a cryptographic key, before granting access to a resource or service. For our research, we have opted to use a smart contract.

The above describes interfacing between the MFA system using the DL approach with SSI towards MFA implementation.

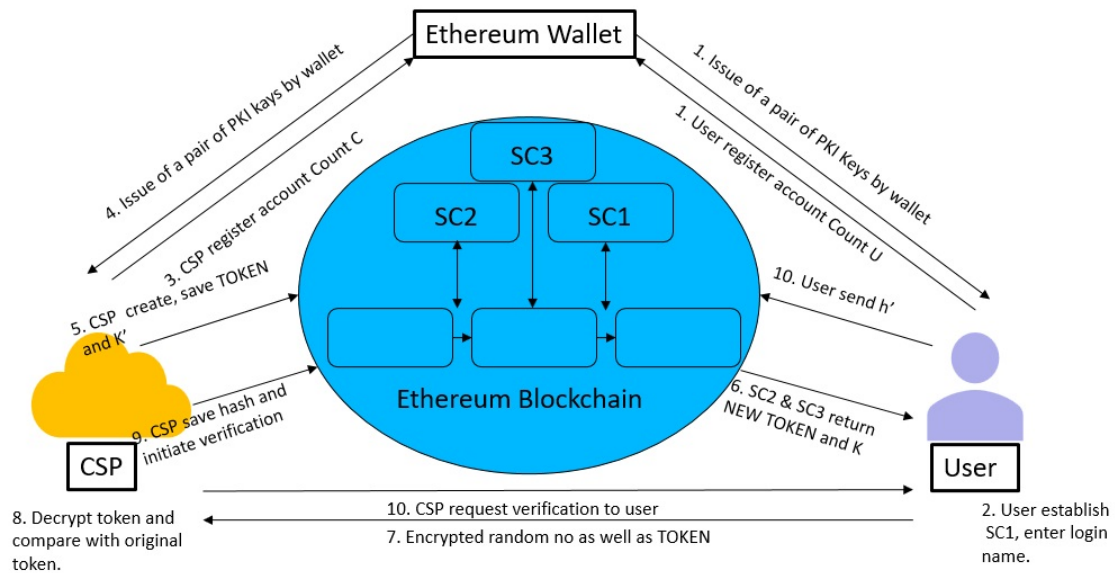
### 5.4.6 Implementation & Integration of Multiple Smart-Contract

A smart contract is a self-executing contract with the terms of the agreement between multiple entities being directly written into lines of code. The code and the agreements contained therein exist on a blockchain network, allowing for automated execution of the terms of the contract without the need for intermediaries. In the present research, we have adopted a smart contract each for the respective tasks to be handled independently in an integrated manner. Various tasks that are expected to be handled by smart contracts are (a) Storage and retrieval of user credentials for verification operation in the authentication process. (b) Storage and execution of authentication in terms of facial biometric verification. and (c) Accessing the storage location for the associated resources access control as part of the authentication process.

A unique mechanism for integrating the execution of smart control deployed over the DL containing the user details regarding credential and facial biometrics is proposed for

## 5 Secure Authentication of Users In Multi-Cloud Environment

implementation, as depicted in Figure 5.5. This unique mechanism is based on RFC 7519-based protocol, which describes the *JSON Web Token (JWT)* format. JWT is a compact, URL-safe means of representing claims to be transferred between two parties. These claims can be used to authenticate, authorize, or share information between parties, which answers RQ1. The JWT format consists of three parts, namely, a header, a payload,



**Figure 5.5:** Integration of Multiple Smartcontract Deployed over Ethereum

and a signature. The header contains information about the token type and the algorithm used to sign the token. The payload contains the claims or statements about the entity that is authenticated. The signature is used to verify the token's integrity and ensure that it has not been tampered with. The JWT format is widely used in modern web applications and APIs to transmit authentication and authorization information between parties. JWTs are often used with OAuth 2.0 and OpenID Connect to enable secure and decentralized authentication and authorization flows. **This answers Research Question 5.3.**

### 5.4.7 Security Requirements & Design Goals

The design goals of an MFA solution using Ethereum and Smart Contracts should prioritize security, privacy, decentralization, accessibility, interoperability, scalability, and auditability to provide a robust and reliable system for identity verification and authentication as enlisted in the Table 5.5. Our design goal is to suggest a method that offers a comprehensive structure to guarantee the utmost protection of cloud users' privacy. Pre-

**Table 5.5:** Design Goals of Blockchain-Based MFA Solution

Ser No	Criteria	Desirable Details
1	<i>accessibility</i>	The MFA solution should be accessible and easy to use for all users, regardless of their technical expertise or physical ability.
2	<i>Auditability</i>	The MFA solution should provide a transparent and auditable system of record for all authentication requests and transactions, which can be used to detect and prevent fraud or unauthorized access.
3	<i>decentralization</i>	The MFA solution should leverage the decentralized nature of Ethereum and Smart Contracts to eliminate the need for a centralized authority or intermediary for identity verification and authentication.
4	<i>interoperability</i>	The MFA solution should be interoperable with other blockchain-based identity management systems and should adhere to industry standards and best practices.
5	<i>privacy</i>	The MFA solution should protect the user's privacy by minimizing the collection and storage of personal data and ensuring that any data that is collected is encrypted and secure.
6	<i>scalability</i>	The MFA solution should be scalable to handle a large volume of authentication requests and should be able to support multiple use cases and applications.
7	<i>security</i>	The MFA solution should provide a high level of security by incorporating multiple factors of authentication, such as something the user knows (Password), something the user has (Smart-card), and something the user is (Biometric Data).

serving users' privacy, including their personal information and identity, is critical in the cloud environment. Moreover, the usage of OEM-fitted webcams fulfills the condition of not requiring specialized hardware and software, which answers RQ2.

### 5.5 Implementation and Evaluation

For face recognition, we followed the integration of face recognition and livenessnet to achieve simplicity, better accuracy, and liveness detection all in one go for the reasons mentioned below.

1. *Face Recognition*: This is the simplest face recognition library for Python based on deep learning. It achieves an accuracy of 99.38 % on the LFW dataset.
2. *Livenessnet (liveness detection model)*: This model is based on deep convolutional neural networks and has an accuracy of almost 100 % <sup>§</sup> in liveness detection.

The system implementation starts with simple steps for user registration, as mentioned in system modeling. The detailed procedure followed for User Registration is presented in Algorithm 1.

D-MFA implementation is the core concept of our work, which dynamically implements MFA in a randomly chosen time interval, which helps ensure the detection and prevention of impersonation attacks. The algorithm of D-MFA is presented in Algorithm 1.

As a separate approach for this research, C-MFA implementation is the core concept of our work, which continuously implements MFA without disturbing logged-in users. It also helps ensure the detection and prevention of impersonation attacks. The conceptual algorithm of C-MFA is presented in Algorithm 2. The schematic diagram of C-MFA is presented in Figure 5.4.

---

<sup>§</sup><https://pyimagesearch.com/2019/03/11/liveness-detection-with-opencv/>

---

**Algorithm 1** Dynamic Multi-factor Authentication (D-MFA)

---

- 1: Initialize,  $H \leftarrow \text{Boolean}$ , &  $(F, G, \text{RAND}, \text{PCOUNT}$  and  $\text{FCOUNT}) \leftarrow 0$
- 2: Prompt for User Id and Password
- 3: Read User Id and Password (as  $x1$ )
- 4: If  $(F(x) = f(x1))$
- 5: Password Verification success
- 6: Set  $F \leftarrow 1$  and switch to Step 12.
- 7: Else  $\text{PCOUNT} = 1 + \text{PCOUNT}$
- 8: EndIf
- 9: While  $\text{PCOUNT} \leq 4$
- 10: Go to Sep 3.
- 11: Authentication Failed and Prompt Auto Save Current Work and Forceful Logging Out of User.
- 12: Initiate camera for Face Recognition
- 13: Read User Id and Face Biometric (as  $y1$ )
- 14: If  $G(y) = g(y1)$  Set  $G \leftarrow 1$
- 15: Else Set  $G \leftarrow 0$
- 16: EndIf
- 17: If  $F \geq 1$
- 18: Authentication Successful if  $H = F \cup G$  is TRUE and wait until Logging out by User
- 19: Generate Random No between 60 and 180
- 20:  $\text{RAND} \leftarrow \text{RandomNo}$
- 21: While  $\text{FCOUNT} \leq 4$
- 22: For  $\text{FCOUNT} = 1 + \text{FCOUNT}$  Do  $\text{RAND} = \text{RAND}/2$
- 23: Go to Step 20.
- 24: EndWhile
- 25: EndIf
- 26: While  $\text{RAND} \geq 2$ ,  $\text{RAND} = \text{RAND}-1$
- 27: Go to Step 13.
- 28: EndWhile

---

---

**Algorithm 2** Continous Multi-factor Authentication (C-MFA)

---

```

1: Initialize  $H \leftarrow$  Boolean, & (F, G, RAND, PCOUNT and FCOUNT)  $\leftarrow$  0
2: Prompt for User Id and Password
3: Read User Id and Password (as x1)
4: If ( $F(x) = f(x1)$ )
5: Password Verification success
6: Set F=1 and switch to Step 21.
7: Else PCOUNT=1+PCOUNT
8: EndIf
9: While  $PCOUNT \leq 4$ 
10: Go to Sep 3.
11: EndWhile
12: Authentication Failed and Prompt Auto Save Current Work and Forceful Logging Out
    of User
13: Initiate camera for Face Recognition
14: Read User Id and Face Biometric (as y1)
15: If Change in captured – frame detected, then check for Face Detection in New –
    Frame,
16: Else generate RANDOM – TIME and wait for RANDOM – TIME for face image
    capture activation.
17: If Face Detected in captured – frame, Do Face – Recognition for  $G(y) = g(y1)$  AND
    Single – Face is detected in frame.
18: Else  $RAND \leftarrow 0$ , and Initiate – Image – Capture
19: EndIf
20: If  $G(y) = g(y1)$  Set G = 1
21: Authentication Successful if  $H = F \cup G$  is TRUE and wait until RANDOM time.
22: Go to Step 14.
23: Else
24: FCOUNT = 1+FCOUNT
25: While  $FCOUNT \leq 4$ 
26: Authentication Failed and Prompt Auto Save Current Work and Forceful Logging Out
    of User
27: EndWhile
28: EndIf

```

---



### 5.5.1 Experimental Setup

The experimental setup for this research implementation and testbed provisioning is done in a phased manner. In the first phase, Local implementation is done. Subsequently, its porting to the cloud environment is done as discussed below.

1. Local Implementation: Every application is first implemented on a local system before it is ported to the cloud. A similar procedure is followed for implementing our solution. The testbed of the system on which the application is implemented is as follows:

- Operating System: Windows 10 (Version: 21H1).
- Processor: Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
- RAM: 16.00 GB
- HDD: 1 TB

A Linux virtual environment (WSL 2) is also used. The *Windows Subsystem for Linux (WSL)* lets the user run Linux command-line applications, tools, and utilities directly on top of Windows. The Linux distribution used for our implementation is Ubuntu 20.04.

2. Hosting the application on the cloud. For ease of use and convenience of implementation, we decided to host the application on Azure Cloud and Google Cloud environments. To host the application to the cloud, we first built our ReactJS code. We have developed the code as a part of different files, and when the user opens a webpage on our app website, it will require an additional HTTP request to access the additional JS/CSS file, which will decrease the overall performance. To deal with that, we built our app that merges all JS into one file, CSS into one file, and HTML into one file. It will also compile the JSX components of the react application into native HTML and CSS. A DockerFile is created inside the root folder of the app. It establishes the fact that we will need Python 3.7 for our app. It then creates a directory inside the docker container as the app and copies all files from the current

directory into it. It then sets up the system and installs necessary packages like *cmake* and *wheel*, which are needed before a required file is installed. Then, we built an image from that Docker file and pushed that file into the Azure and GoogleCloud container registry. We executed the Docker Desktop app running for that which is running on the WSL2 backend.

### 5.5.2 System Implementation

The technical stack followed for the application is as follows:

1. Client End:
  - (i) A JavaScript library called ReactJS is used. It is a declarative, component-based library that uses XML-like syntax called JSX(JavaScript XML).
  - (ii) Most of the React code is written in JavaScript.
  - (iii) HTML code is written as a part of JSX inside React.
  - (iv) CSS is used for styling the components and is imported inside the React file.
  - (v) Node is required to run JavaScript outside the browser.
  - (vi) The node version used for this implementation is v14.18.1
  
2. Server End:
  - (i) Hyperledger Indy and supported APIs for DL and Ethereum environment implementation.
  - (ii) A Python-based web framework called Flask is used. It is a WSGI web app framework to support DDI, a unique identifier.
  - (iii) The Python version used is 3.7.0
  - (iv) OpenCV and Dlib face recognition package.
  - (v) Amazon EC2 instance and Azure virtual machine for installing and configuring HyperledgerIndy over Apache CloudStack 4.17.2 implementation for multi-cloud environment testbed.
  
3. Database:
  - (i) Azure storage account Table Storage for storing the user details.
  - (ii) Server/Local File System to store binary data like images, PDFs

### 5.5.3 System Model

The reasons for choosing a secure web-based interface for this MFA-based solution are its versatility and ease of use in a cloud system environment. A broad description of the system model is as follows.

1. For Sign-up: Registration details of the user as per Algorithm 3.

Capture the face image of the user in the user database.

Register the face of the user.

2. For Log-in(With MFA): Enter Credentials.

Capture face and verify user face to make user logged in.

The user is not logged in if either the credential is incorrect or face verification does not match.

---

**Algorithm 3** User Registration

---

- 1: Prompt for User Id
  - 2: Verify for Unique User Id
  - 3: If ( $UID(is - NOT - Unique)$ ) Prompt as User Exists and Enter Unique UID
  - 4: Else ( $UID(is - Unique)$ ) Prompt and capture Password as per Policy
  - 5: EndIf
  - 6: If  $Password(is - NOT - (as - per - Password - Policy))$  Prompt to re-enter password as per Policy
  - 7: Else  $SaveEncrypted - Password - in - Server$
  - 8: EndIf
  - 9: Prompt for User Facial Biometric
  - 10: Initiate camera for Facial Image Capture
  - 11: Capture User Facial Biometric Images in desired angles AND Map with its User Id.
  - 12: Prompt to Save Captured Biometric Picture Set of User in Server and Exit
- 

In addition, aspects of secure web access and associated distributed databases, along with their connectivity for the secure web interface, is a crucial consideration while implementing this research work.

### 5.5.3.1 Multi-Cloud Testbed & Distributed Database Mapping

The testbed was implemented in a phased manner. Phase I comprised of In-Premise Cloud; in Phase II, AWS cloud infrastructure was integrated. In this phase, 16 nodes of Ethereum mapping were done over the EC2 instances, as depicted in Figure 5.6.

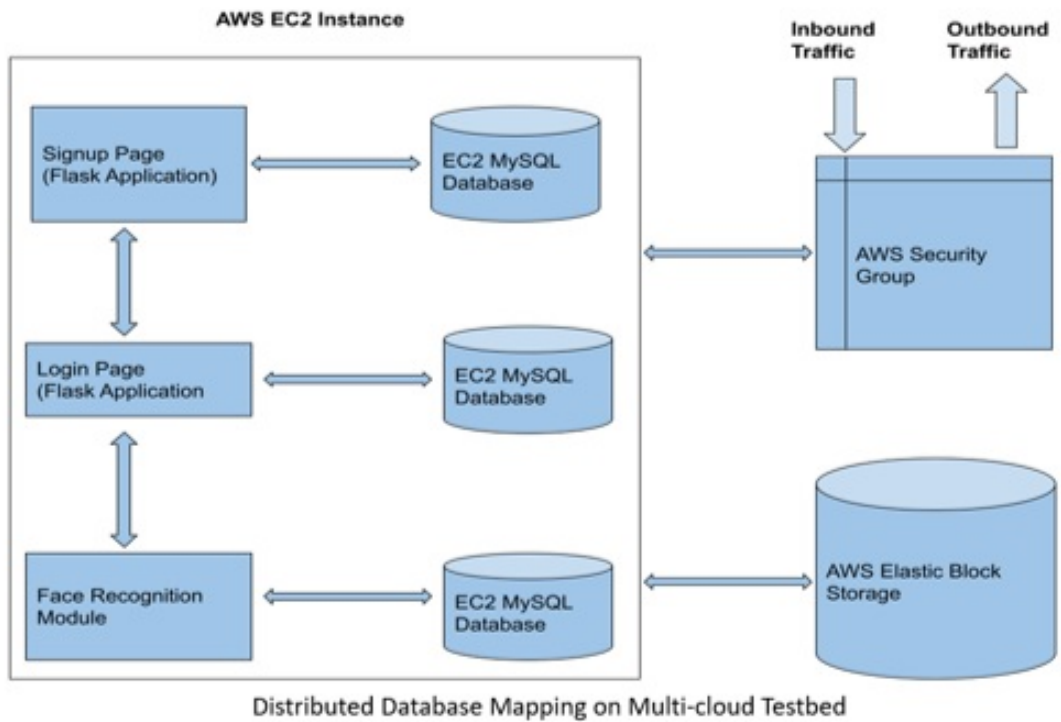
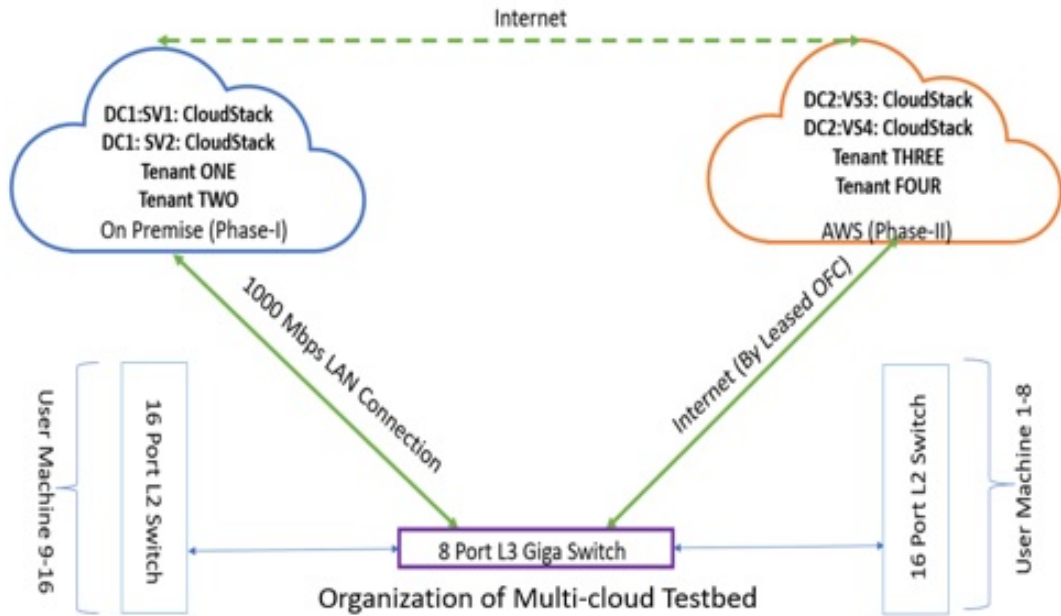
### 5.5.3.2 User Interface

The designed user interface provides a facility for user registration and Dashboard. The user registration process comprises two passes. Firstly, the user credential checks for uniqueness and storing credentials and subsequently, as verified by Algorithm 3, captures and stores the facial characteristics of the associated user. The Dashboard is presented to the user after successfully verifying credentials in the login process. Secondly, facial verification of the user takes place along with a liveness check by the system. The face recognition check phase of the authentication process interface is depicted in Figure 5.7.

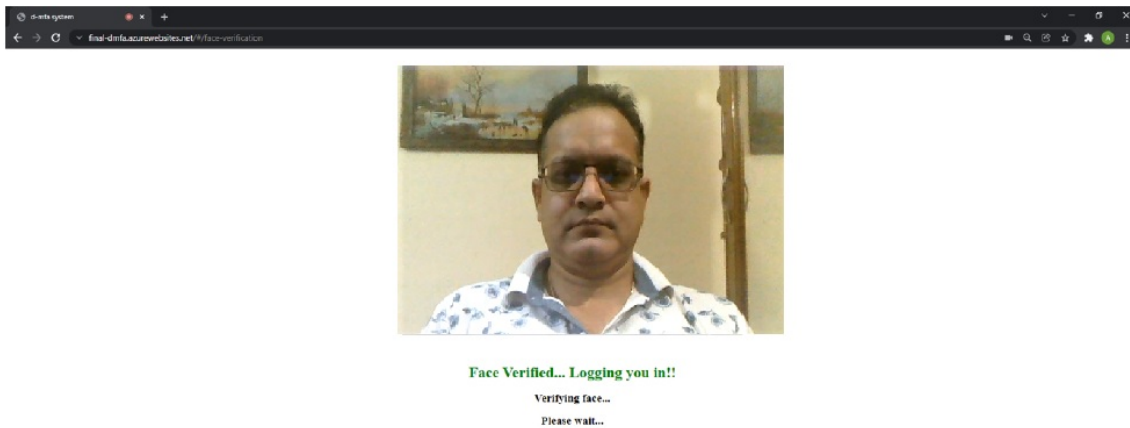
### 5.5.4 System Threat Analysis

Threat analysis for the system was done by envisaging the different situations regarding MFA using the system. The following scenarios are created and followed with specific reference to face recognition and its effect on users' login using MFA:

1. *Login with an actual face:* The user gets authenticated and logged in to the system. Subsequently, the user gets successfully verified every time.
2. *Login with user's face image:* The user gets authenticated and logged in to the system. Subsequently, the user gets successfully verified every time.
3. *Login with the wrong face image:* The face recognition algorithm tries to match the user with the stored image, but the face does not match, so the server sends a negative response, and the user cannot log in.
4. *Login with an actual face, but the user switches to a face image after login:* The



**Figure 5.6:** System Model of D-MFA & C-MFA Over Multi-Cloud Testbed with Distributed Database Mapping



**Figure 5.7:** Login interface for Face Recognition using D-MFA as well as C-MFA

Verification succeeds successfully. The swapped user continues to be logged on and access the system resources. However, the swapped user can't log on afresh after logging out or after the previous successful login session ends.

Threat modeling was done by envisaging the different situations regarding MFA using the system. The following scenarios are created and followed with specific reference to face recognition and its effect on users logging in using MFA.

1. *Login with an actual face (live user):* The user gets authenticated, dynamic & repetitive authentication runs flawlessly in the background every few seconds, and the user gets successfully verified every time.
2. *Login with user face image:* The liveness detection algorithm detects that the person in the image is not an actual person but an image of the user. It sends a negative response to the client side, and the user cannot successfully log in (complete both authentication steps).
3. *Login with the wrong face:* The face recognition algorithm tries to match the user with the stored image, but the face does not match, so the server sends a negative response, and the user cannot log in.
4. *Login with an actual face (live user), but user switches to a face image after login:* The Verification succeeds successfully. However, when the verification endpoint is called every few seconds, the liveness detection fails every time now. Once it fails

three consecutive times, the system logs out the user, and the user log-on session ends.

### 5.5.5 System Evaluation

Initially, the proposed solution for the SSI-based MUF was implemented in an in-premise private cloud testbed with OpenCV as a face recognition algorithm as part of MFA. During implementation, after successful registration and authentication with one and the first user, the registered user density was incremented in phases to observe the system performance. This scalability aspect of our SSI-based MFA implementation was studied with 2, 8, 64, 512, and 2048 registered users. However, due to lab limitations of available 16 terminals, running 04 virtual machines each, in our testbed, 64 users could concurrently log in through the respective terminal browsers to study the system performance while the evaluation was done for 512 and 2018 registered users. The system was implemented in a phased manner, with SFA (Username-password-based) in the first phase, and the system performance was studied. Subsequently, the MFA (with Face Recognition) was implemented after integrating the third smart contract assigned to handle the face recognition factor of authentication on being deployed over the user authentication system blockchain in the Ethereum environment. The observed performance of the system in terms of time taken for user authentication is depicted in Figure 5.8. The authentication time obtained for SFA ranged up to 69 milli sec, whereas with SSI-based MFA in action, the time required for MFA was 70-71 milli seconds, respectively.

The processor load of the in-premise cloud server was checked for SFA and MFA separately by incrementing the registered users from 2 to 2048 in a phased manner with a maximum 64 number of active logged-in users at any time. The processor utilization % obtained for SFA ranged up to 4.6%, whereas with SSI-based MFA in action, processor utilization for MFA was also 4.6%, showing consistent performance at almost all the stages as depicted in Figure 5.9. Subsequently, the solution was graduated for implementation over multiple commercially available clouds testbed for the SSI-based MUF. The virtual servers of the commercial CSPs were used to avail the IaaS facility to install and configure the HyperledgerIndy for cloud setup over CloudStack with Ethereum Blockchain. The installed cloud utilized the secure storage and retrieval of user credentials and biometric

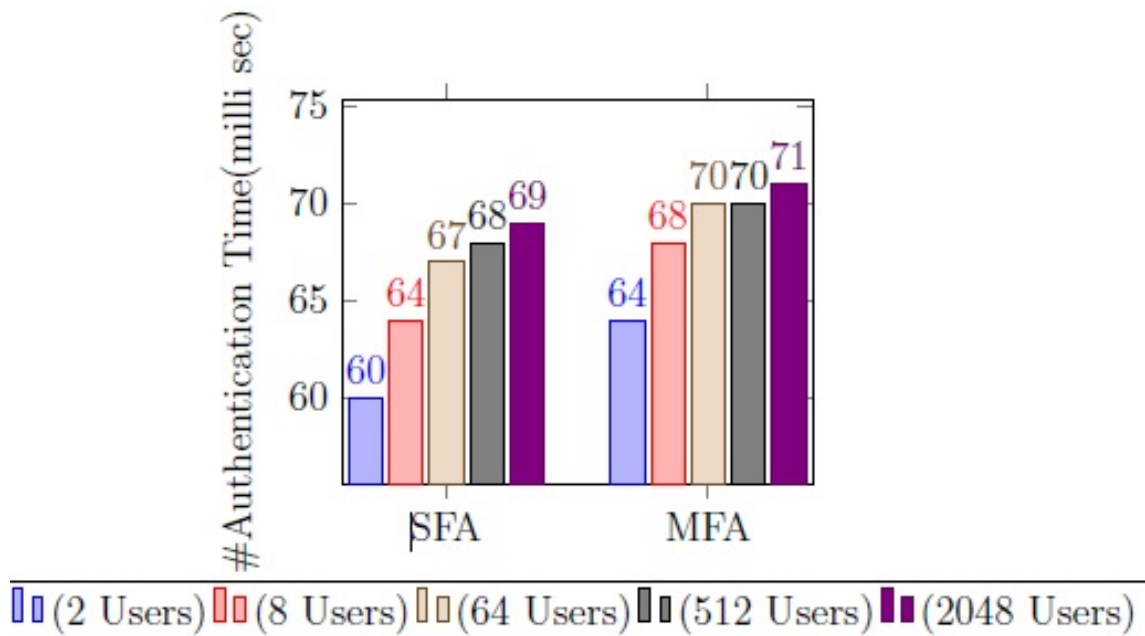


Figure 5.8: On-premise Authentication time Performance Comparison

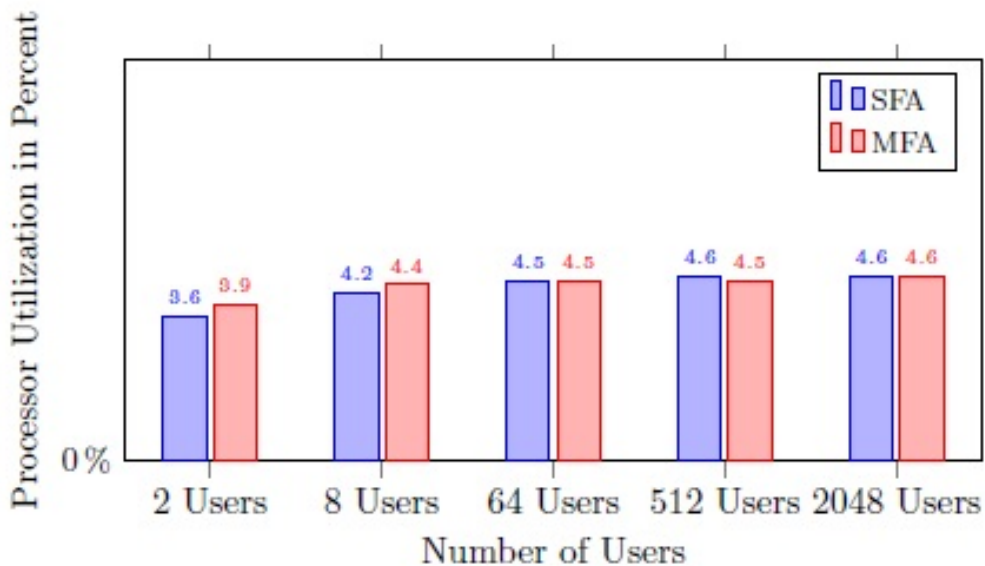
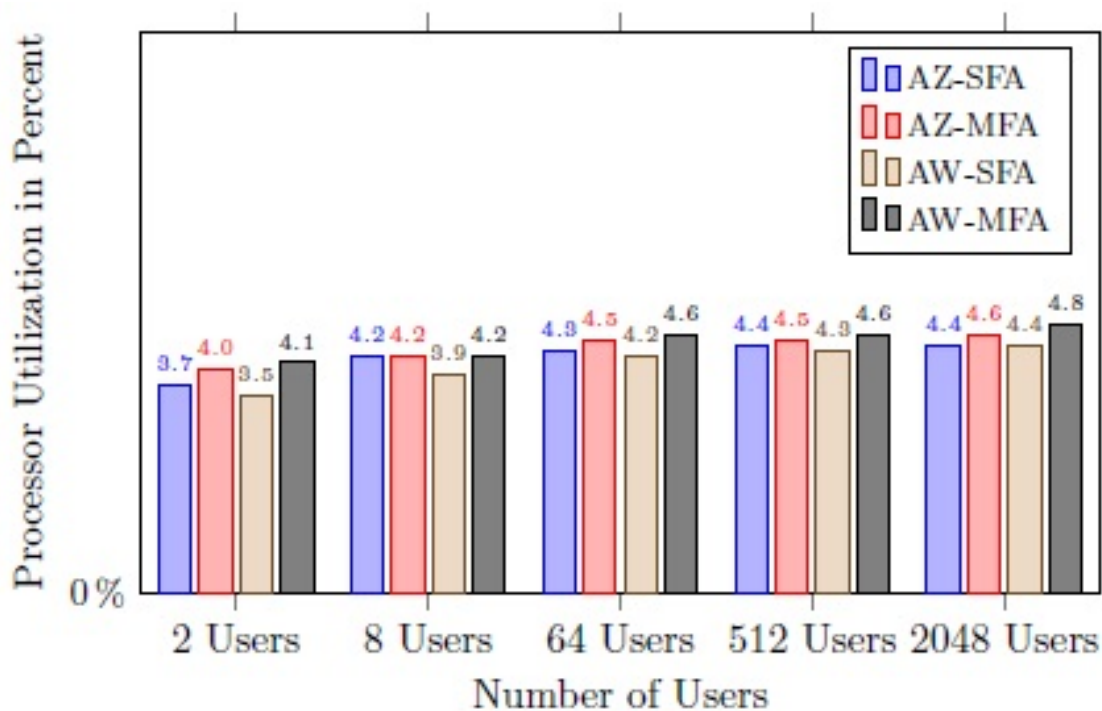


Figure 5.9: On-premise Authentication Processor Utilization Comparison



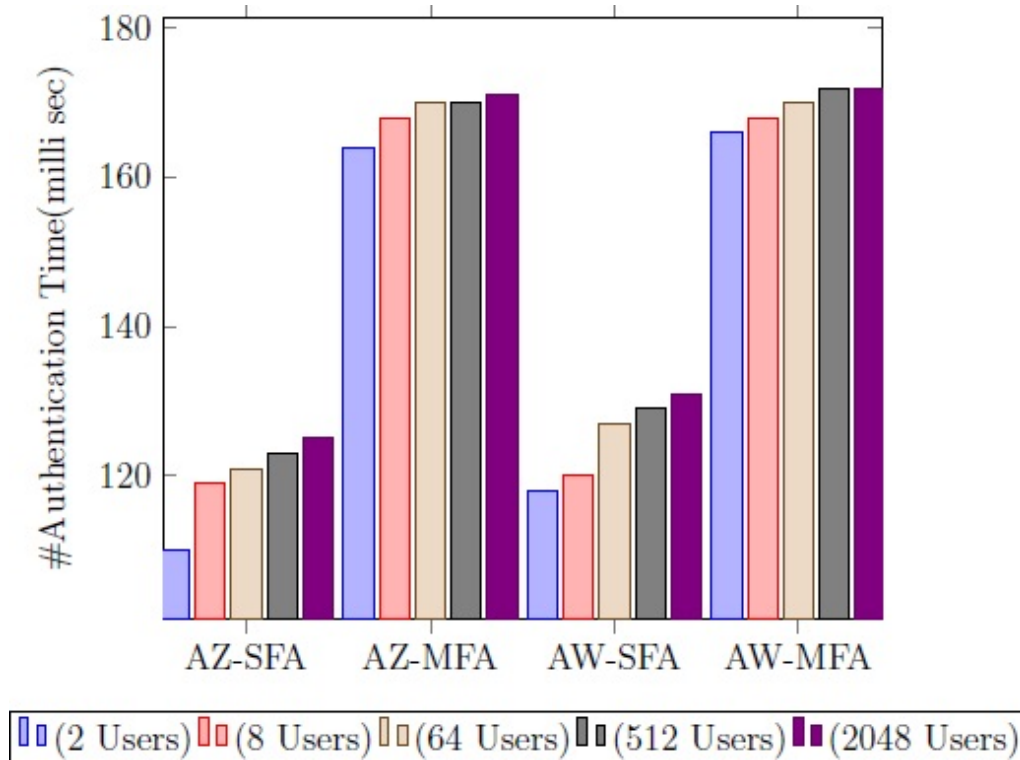
photos of the corresponding users. Processor utilization of the virtual machines configured as cloud servers over the Azure and AWS setup was also monitored with Azure Monitor and Amazon CloudWatch, respectively, in a phased manner with scaling up of registered users in phases with a maximum of 64 users being concurrently logged in as depicted in Figure 5.10.

Similarly, the latency involved in the MFA of Multi-cloud setup over the internet connection with a 100MBPS leased line internet connection was also observed. Observed latency for the CSPs configured locally over Azure and AWS are depicted in Figure 5.11. It is observed that both Azure and AWS virtual server-based configured CSPs provide SSI-based MFA for user authentication in the time range of 164 to 172 milliseconds, including network latency for MFA in the testbed. Simultaneously, on the same setup, SFA, with latency time, was in the range of 110 to 131 milli sec for 2-2048 users registered setup.



**Figure 5.10:** Processor Utilization with Azure & AWS Based Cloud Server for MFA

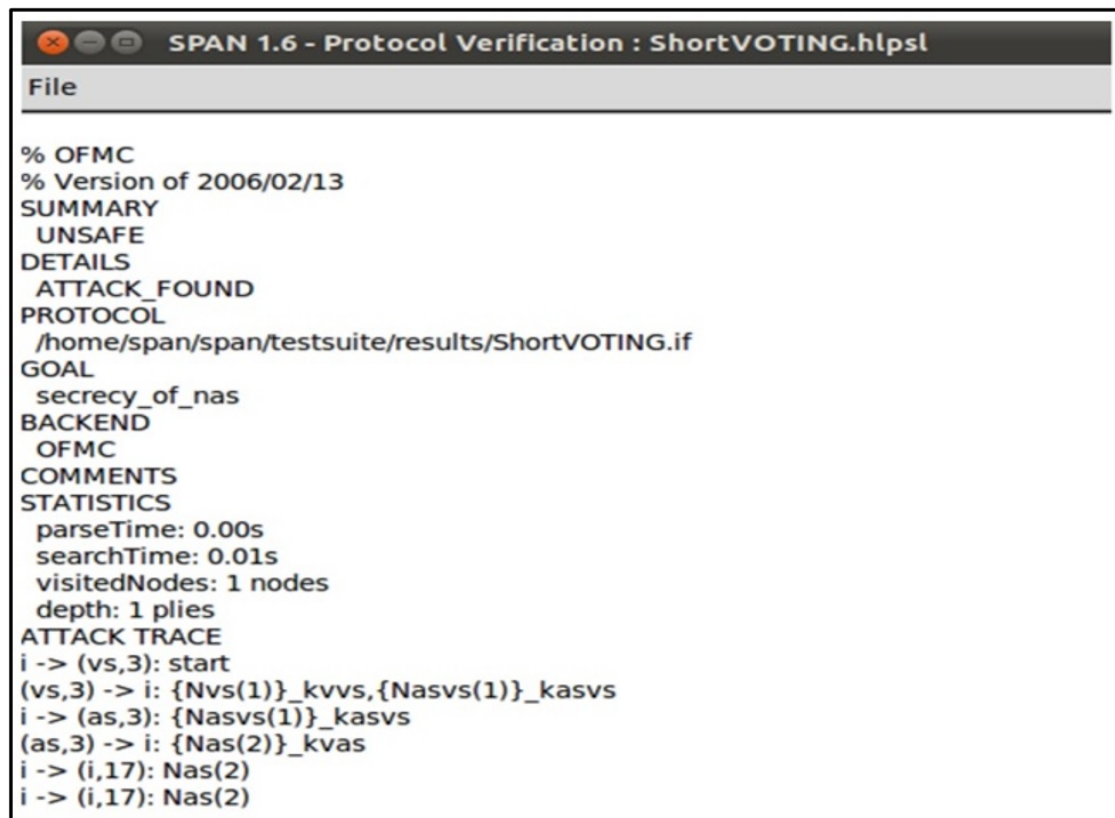
The proposed scheme's security is evaluated using a formal analysis of Burrows, Abadi, and Needham (BAN) logic, informal security analysis, and model checking using the AVISPA tool. It uses Transport Layer Security (TLS), a security protocol that provides privacy and data integrity for Internet communications. Implementing TLS is a standard practice for



**Figure 5.11:** Authentication Time Over Azure & AWS Based Cloud Server for SFA & MFA

building secure web apps. Security Protocol ANimator (SPAN) for AVISPA was used for the analysis. A relevant screenshot is presented in Figure 5.12. The analyzed observations from the testing and its results discussed above are:

1. The Web application is quite secure from various security attacks like injection, cross-site scripting, etc.
2. The application is optimized for unique identification with DID, having SSI functionality
3. The application is optimized concerning response times and its continuous flow
4. The combination, blockchain powered multi-factor authentication (B-MFA system) is successful both with regards to performance and security
5. The approach can deal with a DoS attack scenario
6. The approach is not able to deal with an impersonation attack scenario



```
SPAN 1.6 - Protocol Verification : ShortVOTING.hlpst
File
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/ShortVOTING.if
GOAL
  secrecy_of_nas
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i -> (vs,3): start
(vs,3) -> i: {Nvs(1)}_kvvs,{Nasvs(1)}_kasvs
i -> (as,3): {Nasvs(1)}_kasvs
(as,3) -> i: {Nas(2)}_kvas
i -> (i,17): Nas(2)
i -> (i,17): Nas(2)
```

**Figure 5.12:** SPAN Output Screenshot of AVIPSA Security Analysis Tool

The implemented system has been extensively tested following detailed threat modeling. The results have been analyzed for a deterministic conclusion based on its outcome. Below are the testing mechanisms that are followed.

### 5.5.6 Security Scanning using ZAP

OWASP ZAP is an open-source security scanner for web applications. It can run active scans of various kinds on a website. A screenshot of the test results obtained is presented in Figure 5.13. The following tests were done to test the application, which resulted in no security alerts.

- (i) Client Browser
- (ii) Information gathering
- (iii) Injection

(iv) Server Security

(v) Other Miscellaneous aspects

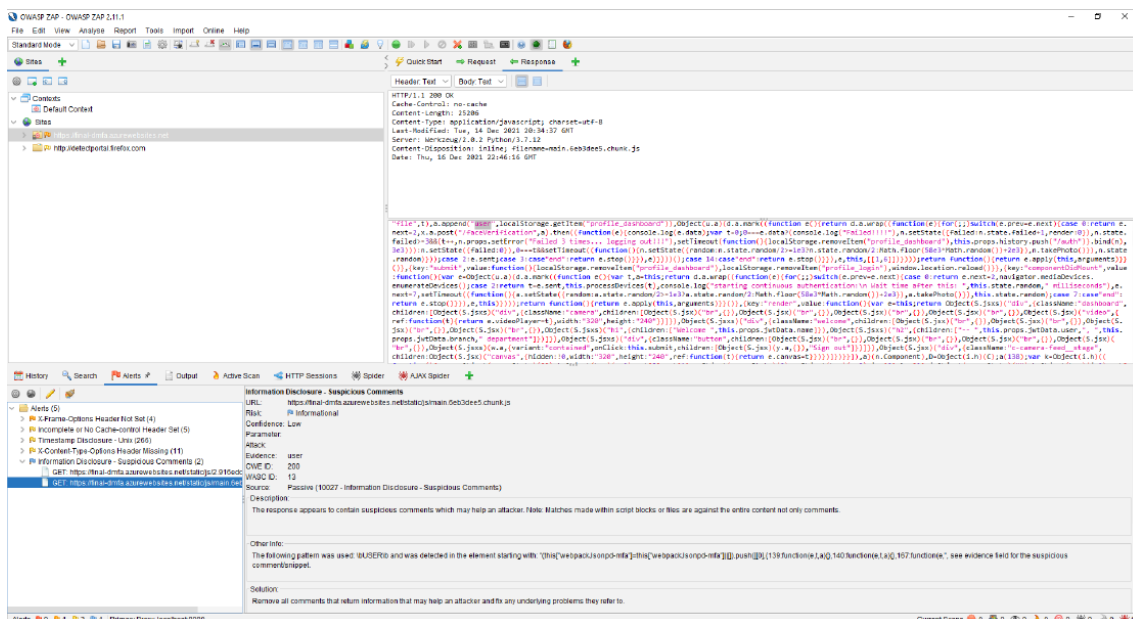


Figure 5.13: ZAP Test Report of D-MFA & C-MFA with Multi-Cloud Testbed

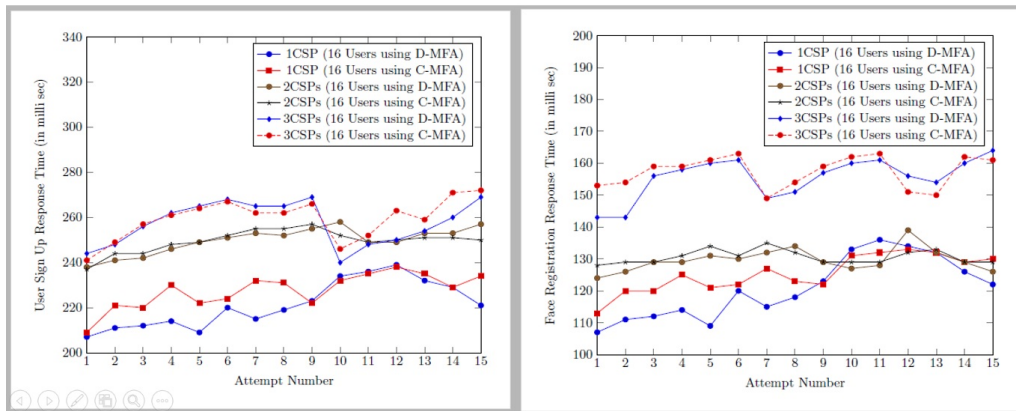
### 5.5.7 API Response Time Patterns

To find the response time of the APIs, we called the APIs 15 times with different users logging in. The response times were noted to get the average response time in milliseconds (ms). It shows that face verification takes the maximum time with an average of 1000 ms (1 second). All of the other APIs take less than 1 second to respond. This face verification endpoint is called during logging in (second step of MFA) and dynamically in the backend during the user's logged-in time. So, on average, it takes around 1 second to verify the authenticity of the user. Details of respective patterns for various phases and constituents of D-MFA and C-MFA are presented below.

The response to a set of fifteen different users was obtained to study the system's overall response towards user sign-up comprising both the discussed phases. The user sign-up took an average of 250 microseconds. For face registration as a part of User registration, we captured five different input images of the concerned user. We studied the capture patterns and time of response for different users, considering fifteen other instances. Each face registration as part of user registration takes an average of 150 microseconds. Figure 5.14

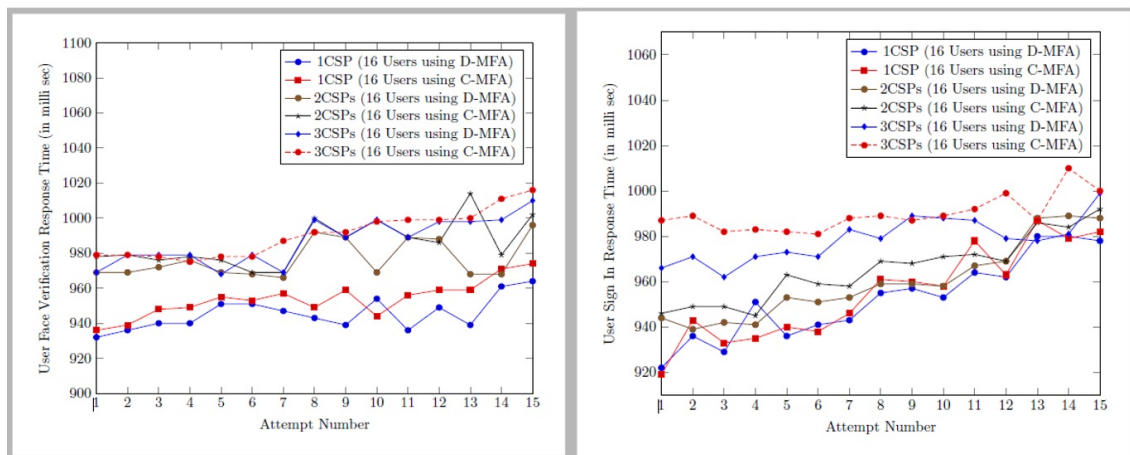
## 5 Secure Authentication of Users In Multi-Cloud Environment

gives the user registration Sign-up response trend and the Face registration response trend in our testbed for D-MFA and C-MFA as a response comparison. We studied the D-MFA



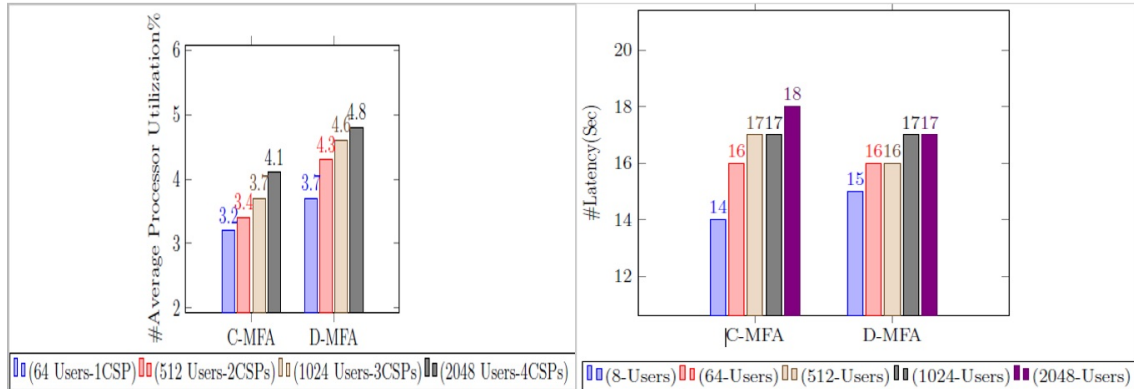
**Figure 5.14:** User Registration Response Time & Face Registration Response Time for D-MFA & C-MFA with Multi-Cloud Testbed

and C-MFA signing-in responses of fifteen different users. Once registered, the average response time for user sign-in is approximately 400 milliseconds. D-MFA and C-MFA activate the face verification functionality in a repetitive or continuous manner to check for the genuineness of the user present in the camera frame and process the facial recognition function. Specifically, the response of this function that runs in the background, without disturbing user functionality of logged-in users during the login session, is studied for fifteen different users. Figure 5.15 presents the response trend of signing-in of users obtained for our system and repetitive face verification. Performance monitored the processor



**Figure 5.15:** User Sign-in Response Time & Face Verification Response Time for D-MFA & C-MFA with Multi-Cloud Testbed

utilization using D-MFA and C-MFA at various registered users and for varied CSPs configured for the Multi-Cloud environment being explored. We also monitored the latency



**Figure 5.16:** Processor Utilization and Latency for D-MFA & C-MFA with Multi-Cloud Testbed

observed in our testbed of the Multi-Cloud environment by uploading an 8 MB pdf file to the cloud storage. Figure 5.16 presents the response of the Multi-Cloud testbed regarding processor utilization as well as latency noticed.

Evaluation of the security of the proposed scheme is conducted using formal analysis of BAN logic, informal security analysis, and model checking using the AVISPA tool. The results obtained are almost the same as that covered in Chapter 4.

The solution using the D-MFA method can handle spoofing and impersonation attacks to a greater extent. The tests were conducted under varying ambient light during face recognition, users with and without spectacles, and the same user with and without beards & mustaches. However, the need to enhance the mechanism's efficiency arises to have a full-proof solution to this problem and secure user authentication for Multi-cloud scenarios. From the output of C-MFA in terms of processing load, it is evident that C-MFA is more efficient than the D-MFA approach. However, latency is almost similar in both cases. It is also inferred from the discussed fifteen instances of experiment that both D-MFA and C-MFA approaches are scalable, and performance is consistent with scaling up the number of registered users along with configured CSPs over the Multi-Cloud testbed. **This answers Research Question 1.3 mentioned at Chapter1, under the preview of user authentication.**

### 5.6 Chapter Summary

The implemented system operated flawlessly with consistent performance in terms of processor load, registration time, and authentication time taken, as designed to provide secure user authentication and secure access to the system resources in a user-friendly manner. The SSI approach for efficient user authentication as part of MFA was checked. From the research mentioned earlier, we observed that despite all SSI and DL-based security measures in place, there was a successful login with legitimate users' photos when used as a part of the MFA process. It is akin to bypassing all effective system measures and requires a liveness test as an anti-spoof measure for the users trying to log on to the system. Similarly, after a successful log-on, when the legitimate user, who initially logged in successfully with desired credentials and with verification of the original face, got swapped with another (duplicate) user, the system could not detect such an act. Accordingly, re-verification of logged-in users has been performed with the CNN-based MixFaceNets method, which was incorporated as a part of the C-MFA and D-MFA process to ensure the liveness check of logged-in users along with an effective means required to detect and check impersonation attacks on the system.

In the present scope of research, biometric face recognition has been used efficiently as a factor for authentication in the MUF process. It has been able to prevent spoofing and impersonation attacks effectively. Strong and inherent security mechanisms of DL using the secure interface, incorporating the DIDs, have been successfully implemented for secure user authentication for a Multi-Cloud environment.

After achieving the scope and desired results of MFA-based secure user authentication with DLT, further identification details for devices in the CoT environment are also explored. With MFA for secure authentication of users in the present chapter, it is observed that similar mechanisms and advanced techniques are needed to develop a secure authentication system for the secure authentication of devices of the CoT ecosystem. Networking Device *Media Access Control (MAC)* number-based unique identifier has been identified as a unique factor to be used for MFA for devices in the CoT environment.

Nowadays, with the advent of new technologies, an additional factor for authentication is also emerging. It can be explained in terms of "What the User does". The research

## 5 Secure Authentication of Users In Multi-Cloud Environment

---

community is exploring this aspect regarding various machine learning algorithms for pattern generation and pattern recognition associated with usual user behavioral patterns like mouse movement, eye blinking, the pattern of browsing web pages, etc.



# 6 Secure Authentication of Devices In Multi-cloud Environment

## 6.1 Introduction

The rise of Internet of Things (IoT) has led to an increase in the number of devices accessing the Internet, including wireless sensor networks (WSNs) that collect data from the environment and transfer it to central locations. WSNs are used in smart cities, agriculture, healthcare, and more. These sensors have limited resources and are mobile, typically consisting of a microcontroller, communication devices, and a power supply unit. A central node is often required to connect the WSN to the Internet using the IP protocol. IoT applications require adaptability, scalability, collaboration, and lightweight yet effective security measures. Combining IoT with Cloud computing can effectively manage the large volume of data generated by these sensors. All IoT applications require certain features, such as the ability to be highly adaptable, scalable, and able to collaborate with various stakeholders. Additionally, there is a need for security measures that are lightweight yet effective [139]. The sheer volume of data generated by these sensors can be overwhelming, but combining IoT with Cloud computing can provide an effective solution to this problem.

From the security point of view, IoT encounters several factors which have a direct relationship with its functional efficiency. Security measures show exponential effect if the IoT devices are used as constituents of a Cloud of Things (CoT). Table 6.1 highlights some of the prominent factors with respect to IoT and its security.

The majority of IoT devices face limitations when it comes to memory, computation,

**Table 6.1:** IoT Security Factors

IoT Security-Factors	Description	Examples
Authentication	The process of verifying the identity of a device or user.	Passwords, biometric authentication, two-factor authentication.
Authorization	The process of determining what actions a device or user is allowed to perform.	Role-based access control, attribute-based access control, mandatory access control.
Encryption	The process of converting data into a secret code to protect it from unauthorized access.	Advanced Encryption Standard (AES), Secure Sockets Layer (SSL), Transport Layer Security (TLS).
Data Privacy	The protection of sensitive data from unauthorized access or disclosure.	Data anonymization, data masking, and data encryption.
Device Management	The process of monitoring and controlling IoT devices to ensure their security and proper functioning.	Firmware updates, device inventory management, device access control.
Network Security	The protection of networks from unauthorized access and attacks.	Firewalls, intrusion detection systems, virtual private networks (VPNs).
Physical Security	The protection of physical assets and infrastructure from theft, damage, or unauthorized access.	Access control systems, surveillance cameras, and physical barriers.
Incident Response	The process of detecting, responding to, and recovering from security incidents.	Security incident and event management (SIEM), incident response planning, forensic analysis.

and power. As a result, conventional security protocols are inadequate in ensuring their protection. The 2016 Mirai botnet attacks were a clear illustration of how vulnerable IoT networks can be without proper security measures in place [140]. The widespread implementation of IoT in sectors such as the Military and Healthcare has made it even more essential to safeguard data privacy and guarantee top-notch security. In IoT-cloud systems, it is necessary to have an Identity and Access Management (IAM) protocol that is lightweight, scalable, and secure.

### 6.1.1 IoT Authentication

The process of IoT authentication is utilized to ensure the security of data and manage access as it moves through an unsecure network by verifying the identities of IoT systems and technologies. This involves identifying and authorizing users, devices, and applications and restricting access to only those who are authorized and trustworthy. The use of authentication in IoT also helps prevent unauthorized access and impersonation of users by attackers, thus protecting sensitive data such as medical health records and sensor readings [141]. To secure IoT communications, there are many different methods available for achieving effective and reliable authentication.

#### 6.1.1.1 Authentication with One Factor

Authentication using only one factor for credential verification in IoT applications and platforms has a significant drawback. This type of authentication is susceptible to brute force attacks, which can lead to the theft of user data. Users or devices often reuse PIN or verification factors across different platforms, which puts them at risk since hackers who gain access to one platform can use the same password to access other platforms. Therefore, "what user knows" based authentication alone cannot guarantee the security of users accounts and data. To address this issue, access control mechanisms like Single Sign-On (SSO), Multi-Factor Authentication (MFA), and Open Authentication (OAuth) are essential alternatives. For example, SSO allows a user to be authenticated using a single set of login credentials and access multiple applications and services on a cloud platform without additional prompts. On the other hand, MFA requires the use of multiple identification factors and access management credentials, making it a more secure option [142]. However, implementing these authentication models requires scalable security management systems and security evaluations.

#### 6.1.1.2 Authentication with Multiple-Factors

Device identification is one of the factors used in MFA for CoT. The following are some of the factors for device identification in MFA for CoT:

1. *Device MAC address*: Every device has a unique Media Access Control (MAC) address that can be used to identify it on a network. The MAC address can be used as a factor in MFA to verify the identity of a device.
2. *Device type and model*: The type and model of a device can also be used as a factor for device identification in MFA. This information can be obtained from the device's firmware or operating system.
3. *Device location*: The location of a device can be used as a factor in MFA. For example, if a user is trying to access a CoT application from a new location, the device can be verified based on its location.
4. *Device fingerprints*: Device fingerprints are unique characteristics of a device that can be used to identify it. These characteristics can include information about the device's hardware, firmware, and operating system.
5. *Device behavior*: The behavior of a device can also be used as a factor in MFA. For example, if a device is behaving in an unusual way or is accessing a CoT application from an unusual location, it may trigger additional authentication measures.

Referring to the above list, the aspect of the requirement for additional and specialized hardware is considered. Considering this, it is inferred that the MAC address of any networked device is an option that is burnt into the hardware by the manufacturer and uniquely identifies the device, which can be used as a suitable factor within the scope of this research. When the first-factor authentication is combined with a second-factor authentication, it's referred to as two-factor authentication. Similarly, multiple factors of authentication can be combined for MFA, Common forms of two-factor authentication for devices include using an IP address or a combination of IP address and MAC address. However, this type of authentication requires extra information from the device, which can be provided through tokens or Pseudorandom PIN [143]. Nevertheless, two-factor authentication that relies on third-party generated tokens is vulnerable to attacks, such as Man-in-the-Middle (MITM) attacks, which can result in token theft.

The Public Key Infrastructure (PKI) has been a reliable method for authentication for a long time. It uses strong cryptographic techniques that are more dependable than pass-

words or tokens. Password or token-based methods are susceptible to DDoS attacks, as seen in the Mirai botnet attack. PKI offers security against various types of attacks. However, the conventional PKI method requires a trusted entity, like a Certificate Authority (CA), which can be vulnerable to the single point of failure attacks [140].

Hence a decentralized approach to address this issue and solve this trust problem using Distributed Ledger Technology (DLT) is the need of the hour to eliminate the need for CA by establishing a trustless environment [144]. Continuous authentication is a security strategy that entails persistently observing the actions of a connected device and conduct to identify any atypical or dubious actions that could indicate an unauthorized device or device profile breach. This approach is necessary due to the inadequacy of conventional authentication techniques, like MAC address or IP address verification, to defend against the rising incidence of cyber-attacks and security violations in CoT environments.

Random authentication is a security strategy that involves requesting additional device authentication at random intervals during a session, even after successful authentication. This approach enhances security by adding an extra layer of protection and decreasing the possibility of unauthorized access and data breaches. Random authentication may also offer improved performance compared to other forms of authentication that could require frequent re-authentication.

### 6.2 Background Literature Study

Authentication of devices in the Cloud of Things (CoT) requires a few key steps:

1. *Device registration*: Devices need to be registered with the CoT platform before they can be authenticated. During registration, the device should be assigned a unique identifier or key, which will be used for subsequent authentication.
2. *Authentication protocol*: Choose an authentication protocol that is suitable for the device and the CoT platform. Popular protocols for CoT include OAuth 2.0, OpenID Connect, and Message Queuing Telemetry Transport (MQTT).
3. *Device authentication*: When the device attempts to connect to the CoT platform, it

**Table 6.2:** Comparison of Communication Protocols for Authentication & Authorozation

Protocol	Purpose	Authenti- cation	Author- ization	Transport	Format
X.509 cer- tificates	Device au- thentication	Yes	No	TCP/IP, HTTP/HTTPS	Certificate
OAuth 2.0	User and de- vice autho- rization	Yes (for users)	Yes (for users and devices)	HTTP/HTTPS	JSON
MQTT with TLS	Secure mes- saging pro- tocol for IoT	Yes	No	TCP/IP	Binary
JSON Web Tokens (JWT)	User and de- vice autho- rization	Yes (for users)	Yes (for users and devices)	HTTP/HTTPS	JSON

should present its unique identifier or key, along with any other required credentials (such as MAC address or IP address). The CoT platform will verify these credentials and determine whether the device is authorized to connect.

4. *Authorization:* Once the device is authenticated, the CoT platform will determine what level of access the device should have. This may include permissions to read or write data, access certain resources, or perform specific actions.

A detailed study and analytical comparison of communication protocols used for authentication and Authorization have been made to find out the suitability of a particular protocol for our present research. The protocol comparison is depicted in Table 6.2. Due to the limited resources of IoT devices, the conventional communication and cryptographic protocols are unsuitable for use in IoT environments. Even when they can be implemented, their performance is often inadequate [145]. Therefore, lightweight communication and cryptography has become necessary to address these challenges. Its goal is to provide a solution for resource-constrained devices by reducing the key size, cycle rate, throughput rate, power consumption, and area. Lightweight communication and cryptography algorithms have been extensively researched in recent years due to security requirements in resource-constrained devices such as Radio Frequency Identification (RFID) and smart cards. Researchers have done a significant amount of work related to lightweight cryptography, including efficient implementation of traditional cryptography algorithms and the

design of new lightweight algorithms and protocols. Additionally, many academic communities and international organizations, such as Internet Engineering Task Force (IETF), European Telecommunications Standards Institute (ETSI), and IEEE, have contributed significantly to normalizing and developing IoT security standards. IEEE802.15.4, Zig-Bee, WirelessHART, ISA100.11, 6LoWPAN, and Bluetooth Low Energy are among the standards that have been established.

In the context of devices, the protocols have slightly different purposes and capabilities. X.509 certificates are used for device authentication, ensuring that the device is who it claims to be. OAuth 2.0 is used for user and device authorization, allowing devices to obtain access tokens that can be used to access cloud services on behalf of the user or device. MQTT with TLS is a secure messaging protocol for IoT, providing encrypted communication between devices and the cloud server. JSON Web Tokens (JWT) can be used for both user and device authorization, providing a secure method for transmitting authorization data between parties. The choice of protocol depends on the specific needs of the device and the system it is part of. For example, a device that needs to securely communicate with a cloud server might use MQTT with TLS, while a device that needs to access cloud services on behalf of a user might use OAuth 2.0 with JWT. X.509 certificates being the lightweight among these, is considered to be used for device authentication, considering the MAC address as the input for issuance of the digital certificate.

### 6.2.1 IoT Environment & Security Requirements

IoT environment and the different security requirements needed to design a proposed solution for meeting the needs of secure authentication needs due deliberation.

#### 6.2.1.1 IoT Environment

In IoT network configuration mostly comprises of a cloud (CoT) as the umbrella over it. In this, a number of IoT devices or sensors are linked to a single gateway. Communication between devices is only possible through the gateway via wireless channels. These IoT devices are small in size and rely on batteries, which limits their energy capacity. The gateway and devices are designed to operate in an offline environment, meaning they

are not connected to a central device or a trusted third party. The devices have limited memory, but manual configuration is possible, allowing an administrator to manually set up a shared secret key in both the device and the gateway. To store the shared private key and the hash of the sensor's and gateway's IDs, non-volatile memory like electrically erasable programmable read-only memory (EEP-ROM) is necessary. The communication between the device and gateway requires a few megabytes of RAM to store information shared during previous communication sessions, which is used to update the session key. Finally, the network is established in a secure area where unauthorized access to the devices is not possible, ensuring the confidentiality of the stored secret keys.

### 6.2.1.2 Security Essentials

With the exponential increase in the number of IoT devices connected to the Internet, there are now more opportunities for potential security vulnerabilities to be exploited. When a device is connected to the Internet without proper security measures, it becomes an easy target for cyberattacks, potentially resulting in data theft for the user. Ensuring the security of IoT devices should be a top priority, and manufacturers must implement security mechanisms that guarantee user privacy and security. Building trust between users and their connected devices is critical to protecting online activity from cyberattacks. If users lack trust in the security of their IoT devices and information, it can lead to real problems. Therefore, a new IoT protocol is necessary to address these security challenges and provide secure authentication, message integrity, and confidentiality. Though Adi Shamir [146] introduced the concept of identity-based cryptography (IBC), where individuals can utilize their identity information as public keys for direct communication. However, it wasn't fully implemented until 2001, when Boneh and Franklin utilized bilinear pairings on elliptic curves to create identity-based encryption (IBE). Although IBE is a great alternative to PKI, it still has issues with the key escrow and a single point of failure. Since the key generation center (KGC) generates and sends private keys to users, it has the ability to decrypt messages and manipulate data, which is still a difficult challenge to overcome [147].

Many researchers have experimented with PKI in various DLT applications for Blockchain implementations. A comparison of some of the recent works (from the year 2019 to 2021)



**Table 6.3:** Comprative Analysis of Research Works on PKI-Based Blockchain

Ref	Platform	Consensus	Type	On-Chain Storage	Off-Chain Storage	Time Copm-lexity	Trust Model	Certi-ficate
[148]	Ethereum	PoW	Permi-ssion Less	NA	Public	$O(n)$	WoT	Custom
[149]	Ethereum	PBFT	Permi-ssion Less	Hash	Public Data	$O(\log(n))$	X.509	Hiera-rchical
[150]	Ethereum	PoS	Permi-ssioned	NA	Public	$O(\log(n))$	Custom	Hiera-rchical
[151]	Ethereum	PoW	Permi-ssioned	NA	Public Data	NA	X.509	WoT
[152]	Ethereum	PBFT	Permi-ssion Less	Hash	Public Data	$O(n^2)$	Custom	WoT

depicted in Table 6.3 highlights that PKI is well suited for Blockchain implementation in many flavors. From the Table 6.3, the following are inferred.

- (i) Ethereum Blockchain is checked for stability.
- (ii) It supports all most all consensus methods.
- (iii) Extensive use of hashing algorithm for storage.
- (iv) It supports Off-Chain storage.
- (v) However, certificates are dependent on a CA, which, if DoS attacked could affect the entire system

In the context of PKI, a CA is a reliable third-party or device manufacturer that can securely store cryptographic keys. However, if a malicious actor gains access to these keys, the network becomes susceptible to insider attacks. Additionally, the process of reissuing or renewing certificates can be complicated. The expense of certificate signing can become a significant obstacle for implementing the conventional Transport Layer Security (TLS)

or Secure Sockets Layer (SSL) protocol in IoT networks, especially when considering a large number of IoT devices. The signing and verification times for a CA as a third party can be quite lengthy. Although it is possible to decrease these times. But by doing so, it could make the network more vulnerable to security threats. Managing root certificate lists can be challenging, especially as the number of devices in the network increases due to the involvement of multiple CAs based on the devices present.

These types of drawbacks are noticeable in the CA-based PKI environment [153]. Numerous authentication protocols have been suggested for IoT, including those employing public-key cryptography such as traditional or elliptic curves as documented in references [154–157]. However, the implementation of this type of cryptography can be challenging for resource-constrained devices due to their limited memory and power supply [158]. Alternative proposals, such as those using traditional encryption algorithms like AES, have also been suggested but can still have a significant impact on the limited memory and processing power of IoT devices. A research [159] reveals that the execution time of an encryption algorithm is 75.93% higher than that of a hash function. In contrast, our proposed protocol solely utilizes lightweight operations such as xor, addition, subtraction, and hash functions, resulting in a minimal impact on the limited computing and battery resources of IoT devices.

A comparative analysis of lightweight cryptographic hash functions has been carried out and presented in Table 6.4. In the Table 6.4, Pre-image resistance is a property of the hash function that it is computationally infeasible to get input that maps to a given hashed value. For the QUARK hash function,  $(O)2^{128}$  computations may be required. As indicated by power and throughput, the first three hash functions are more lightweight than SHA3/Keccak. In the case of IoT networks, the number of devices is expected to be large, and the size of each individual device is expected to be small. If high power is produced in these small devices, it is likely to cause heating and increase the temperature of the entire device. Any such rise in temperature will decrease the throughput of the device. For this reason, importance is given to power as compared to throughput while choosing the hash function. Hence we conclude to use SPONGNET as the hash function for this research.

A solution for managing identities on a blockchain should allow for selective storage of iden-

**Table 6.4:** Comparison of Lightweight Hash Functions

Name	QUARK [160]	PHOTON [161]	SPONGNET [162]	Keccak (SHA3) [163]
Digest/Output Size (bits)	136	128	128	128
Pre-image Resistance	$2^{128}$	$2^{112}$	$2^{120}$	$2^{64}$
Second Preimage Resistance	$2^{64}$	$2^{64}$	$2^{64}$	$2^{64}$
Collision Resistance	$2^{64}$	$2^{64}$	$2^{64}$	$2^{64}$
Power (Micro Watt)	2.44	2.29	2.20	11.50
Throughput (Kb/s @ 100kHz)	1.47	1.61	0.34	14.4

tities. These identities must be verified by authorities or other entities on the blockchain. The process typically involves an entity making a verifiable claim and having it attested to based on certain distinguishing attributes such as phone number, email, government-issued IDs, or biometrics. In the management of identities on a blockchain, it is important to differentiate between the digital identifier (which uniquely identifies the entity) and the associated attributes. It is crucial to handle the storage of attributes with well-defined principles, as unauthorized or uncontrolled disclosure of attributes can lead to security and privacy breaches.

### 6.3 Research Gaps and Questions

Many researchers have tried exploring the details identity and security measures in the area of IoT device. It has also happened several times that a significant research contribution has further given rise to many subsequent research questions, yet to be answered.

#### 6.3.1 Cryptographic Infrastructure

The majority of authentication procedures currently in use rely on PKI which utilizes digital signatures and asymmetric algorithms. However, traditional PKI systems require

a CA to be trusted, which creates a single point of failure. This reliance on a trusted third party to issue certificates presents challenges for PKI systems. Additionally, the limited resources of IoT devices make it difficult for them to undergo lengthy and complex authentication procedures, and storing and computing keys may be inefficient for these devices. IBE requires a private key generator (PKG), who is a trusted entity responsible for providing private keys to clients in the system. Similarly, IBC also employs a PKG to distribute private keys to users, which creates a single point of failure and adds a third-party trust factor to the system. This reliance on the PKG results in an inherent key escrow problem since the PKG can generate private keys and decrypt all encrypted messages.

The above drawback of IBC and PKG can be overcome by digital certificates. Nevertheless, the digital certificate itself is dependent on a CA. Hence to avoid dependency on CA, there is a need to work towards a concept best described as *Self Signed Digital Certificate* [164].

### 6.3.2 Device Biometric Authentication

Biometric authentication uses a physical characteristic of the device to verify its identity. Biometric authentication can be highly secure, but it can also be subject to spoofing attacks. In terms of devices, MAC address, manufacturer coined RFID Tag, or Global Unique Device Identification Database (GUDID) number are some of the alternatives which can uniquely identify the devices. In recent times several instances of MAC address spoofing attack [165] incidents, specifically in public networks involving sensors and IoT devices, have been noticed. The criticality and adverse effect of the spoofing attack on an IoT sensor cloud-based smart-city cloud-based system can be well imagined in terms of the chaotic situation and panic it can trigger. MAC address being an integral part of any smart connected device, essentially part of any IoT device of CoT. This can act as a factor of device authentication following *What the device is* concept.

The vulnerability associated with MAC address spoofing and its adverse effect [166] can be avoided by using a cryptographic hash function over the MAC address. Similarly, multiple factors like MAC and Associated digital certificates can be combinedly subjected to SHA algorithm-based function to make it virtually unbreakable.

### 6.3.3 Storage of Credentials

The disadvantages and storage vulnerability associated with centralized storage and retrieval mechanism has been described. To avoid an SPF situation and to have an additional layer of database security, the use of DLT is a recommended choice. To make it interactive and enhance its functional capability from the end user point of view, the same needs to be empowered with suitable interfacing programmability.

Usage of Ethereum and smart contract deployed over the DL is a technically recommended option to avoid an SPF situation as well as to avail the inherent cryptographic and auditability features of Ethereum blockchain as an application of DLT on peer-to-peer networking.

From the above discussion, the following research questions have been formulated to be answered in this research.

1. RQ6.1. How to develop a lightweight mutual authentication method that suits resource constraint devices to work on a peer-to-peer networking model?
2. RQ6.2. What approach is to be followed for system modeling for implementing a lightweight, secure authentication for IoT, and what methodology be followed for its security analysis before its implementation?
3. RQ6.3. How device biometric authentication in a repetitive mode in MFA be used for securing device authentication in multi-cloud?
4. RQ6.4. How to deploy multiple Smart-Contract in an integrated manner over a Blockchain for IAM functionality, considering security requirements for threat modeling?
5. RQ6.5. How an efficient and continuous MFA be achieved for enhanced security towards device authentication in the multi-cloud scenario?

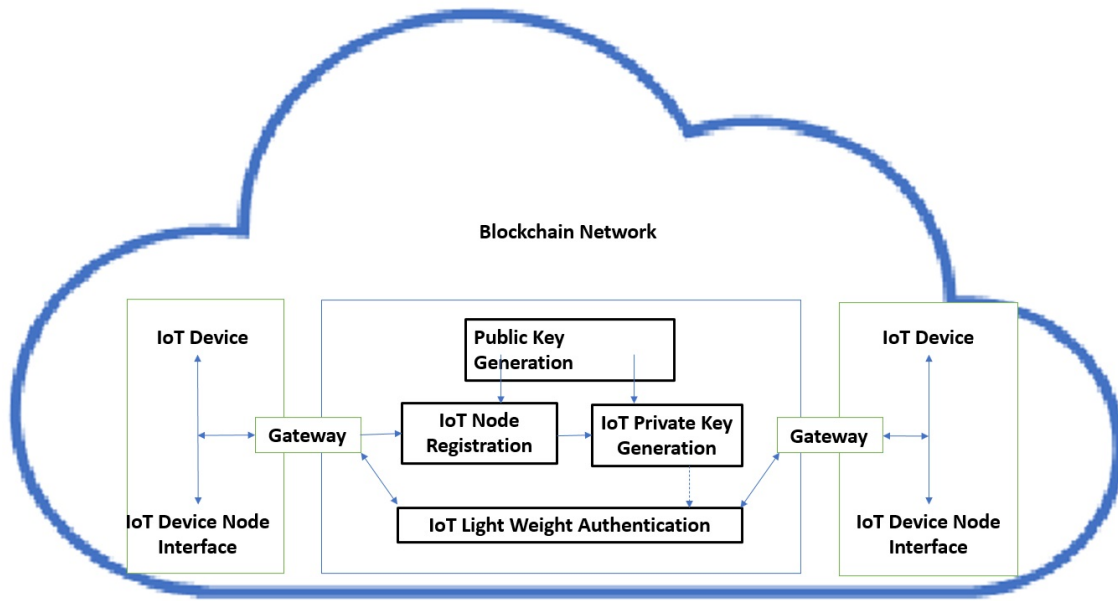
**Table 6.5:** Notations Used for Lightweight Secure Authentication of Devices

Ser No	Description	Used Notation
1	A set of Random Sequence numbers used in key update of i-th session	(seq-Nb <sub>i</sub> )
2	A set of Random Sequence numbers used in key update of (i+1)-th session	(seq-Nb <sub>i+1</sub> )
3	Based on (seq-Nb <sub>i</sub> ), a random frame from (i-1)-th session	R-frame <sub>i</sub>
4	First update-key	K <sub>i-k</sub>
5	Hashed Gateway-ID	h-(G <sub>id</sub> )
6	Hashed Message Authentication Code	H-MAC
7	Hash-key derivation function	H-KDF
8	Hashed Sensor-ID	h-(S <sub>id</sub> )
9	Length of Key	K <sub>lnt</sub>
10	Message authentication code	M-A-C
11	Permanent key burnt into all the devices (MAC Address)	K <sub>p</sub>
12	Random numbers used for a challenge (used Once for each session)	K <sub>1</sub> and K <sub>2</sub>
13	Time of updating session (i-th) session-key)	sesn <sub>time<sub>i</sub></sub>
14	Time of updating next session (i+1)-th session-key	sesn <sub>time<sub>i+1</sub></sub>
15	Update key for symmetric session-key	K <sub>u</sub>

### 6.4 Proposed Approach for Multi-Factor Authentication

To address these security challenges and provide secure authentication, a lightweight authentication for message integrity and confidentiality is proposed. A schematic diagram of various interactions among the stakeholders using a distributed ledger approach deployed over the cloud is represented in Figure 6.1. Details of various notations used in this secure and lightweight device authentication is depicted in Table 6.5. The lightweight secure authentication approach encompasses the following:

1. *Mutual Authentication:* Before two devices can communicate, mutual authentication must take place. To achieve this, the sensors send an encrypted mutual authentication message to the gateway, including the hash of the sensor’s ID and a challenge.



**Figure 6.1:** Lightweight Secure Authentication for Devices of Cloud

To ensure message integrity and authenticity, the sensors calculate a message authentication code (M-A-C) for the entire message. Upon receiving the message, the gateway decrypts it and compares the received hash with the stored hash of the sensor's ID. If the hashes match, the gateway proceeds to calculate the M-A-C for the message. If the calculated M-A-C matches the received M-A-C, the gateway authenticates the sensors and sends an encrypted reply message containing its own ID hash, a new challenge, a set of random sequence numbers, the session time, and the M-A-C of the entire message to the sensors. The sensor then decrypts the message and compares the received gateway ID hash with the stored hash. If they match, the sensor proceeds to verify the received M-A-C against its own calculated MAC. If the M-A-Cs match, the sensor authenticates the gateway, and secure communication between the two devices can begin. During normal communication, attackers cannot intercept the data because all authentication credentials are encrypted and hashed using a secret pair key manually inserted by the administrator into the devices.

Mutual authentication process has the following steps:

- a) *Step-1 (Sensor to Gateway):* The process of mutual authentication begins with the IoT device, or sensor, sending a message to the gateway that includes the hash of its ID ( $h(S_{id})$ ) and a challenge value ( $K_1$ ) that was randomly generated.

This initial message is encrypted using a permanent key ( $k_p$ ). To ensure the message's integrity and authenticity, the sensor calculates the M-A-C for the entire message (M-A-C  $K_i k(msg1)$ ) using the initial update key ( $k_{i-k}$ ).

- b) *Step-2 (Gateway to Sensor)*: Upon receiving the mutual authentication message from the sensor, the gateway proceeds to decrypt it and compare the hash of the sensor's ID received with the stored hash. If the hashes match, the gateway then moves on to the next step of calculating the M-A-C for the entire message. However, if the hashes do not match, the authentication process fails. If the calculated M-A-C matches the received M-A-C, that means  $M - A - CK_{i-k}(msg1)_{calc} = M - A - CK_{i-k}(msg1)_{recd}$ , then the gateway authenticates the sensors. On the other hand, if the M-A-Cs do not match, the authentication process fails. The gateway then selects a sequence of numbers (seq-Nb<sub>i+1</sub>) that will be used to update the session key and generates a new challenge K2, the predefined session time  $sesn_{Time_i}$ , K1, and the hash of its ID ( $h(G_{id})$ ). These are all encrypted with  $k_p$ , and the gateway calculates M-A-C  $K_{i-k}(msg2)$  using  $k_{i-k}$  and sends it to the IoT device.
- c) *Step-3 (Sensor to Gateway)*: Once the gateway's message is received by the sensor, the message is decrypted and the received hash of the Gateway's ID is compared with the stored hash ( $[h-(G_{id})_{stor}] = [h-(G_{id})_{recd}]$ ). If the two hashes match, the authentication process continues by verifying the received MAC with the calculated M-A-C ( $[M-A-C- K_{i-k}(msg2)_{calc}] = [M-A-C-K_{i-k}(msg2)_{recd}]$ ). If the two MACs are the same, sensor device successfully verifies the gateway's authenticity and confirms that the gateway has the correct initial session key and has received K1 from the first message. If the two M-A-Cs are not identical, authentication fails. To confirm that it has successfully received K2 and  $sesn_{Time_i}$  from the previous message (message 2), the sensor calculates the hash of K2, (seq-Nb<sub>i+1</sub>),  $sesn_{Time_i}$  ( $h(K1 || K2 || (seq-Nb_{i+1}) || sesn_{Time_i})$ ) and the M-A-C for the entire message (M-A-C  $K_{i-k}(msg3)$ ) all encrypted with  $k_p$ , and sends it to the gateway. The pseudorandom numbers K1 and K2 are randomly generated and contribute to the process akin to different factors of authentication. However, this does not significantly enhance the computing complexity of the process



d) *Step-4 (Gateway to Sensor)*: After receiving message 3 from the sensor, the gateway will first decrypt the message and check whether  $[h-(S_{id})_{stor}] = [h-(S_{id})_{recd}]$ . If they're not equal, authentication will fail. If they are equal, the gateway will proceed to calculate the M-A-C for the received message and compare it with the received M-A-C  $[M-A-C K_{i-k}(msg3)]_{calc} = [M-A-C K_{i-k}(msg3)]_{recd}$ . If the two M-A-Cs match, the gateway will authenticate the sensor since it knows that the sensor has received K2, (seq-Nb<sub>i+1</sub>), correctly from message 2. To confirm that it has received message 3 from the sensor correctly, the gateway will then send an acknowledgement (ACK), the encryption of the hash of its ID, and M-A-C  $K_{i-k}(msg4)$  to the sensor. The permanent key ( $k_p$ ) will be used as the secret key.

Upon receiving message 4 from the gateway, the sensor will decrypt the message and validate both  $h-(G_{id})$  and M-A-C  $K_{i-k}(msg4)$ . If they're valid, the sensor will confirm that the gateway has received message 3 successfully, which means that the mutual authentication process has been completed. After mutual authentication, the sensor and gateway will generate a session key ( $k_u$ ) using the Hash function, with K1 and K2 serving as inputs, the key length determined by Klth, and  $k_{i-k}$  serving as a Hash-key. The session key will be used for encryption and hashing until the predetermined session time expires. Once the session time is over, the session key will be updated using the previous session information.

2. *Communication Between Device & gateway*: In the diagram shown in Figure X, the sensor communicates with the gateway by sending encrypted and hashed session frames using a session key. Upon receiving a message from the sensor, the gateway first decrypts it and then calculates the M-A-C for the entire message. If the received M-A-C matches the calculated M-A-C, the gateway authenticates the sensor since it assumes that the sensor has the correct session key. The gateway then sends an encrypted and hashed acknowledgment message to the sensor, indicating that it has received the previous message correctly. Once the sensor receives the acknowledgment message, it confirms that the gateway has the correct session key and proceeds to send the next frames to the gateway. This process is repeated until the predefined session time expires. During this phase, the sensor and gateway keep all the infor-

mation needed for the next session key update secret. Once the predefined session time has elapsed, the next phase for updating the session key can begin.

3. *Exchange of Key & Updation of Session Key:* To establish secure communication, the sensors, and gateway use a secret key that they share. There are two primary shared secret keys: a permanent key ( $k_p$ ) and an update key ( $k_u$ ). The permanent key is used to encrypt the authentication messages, while the update key is used to encrypt and hash communication during sessions. To hash authenticated messages between devices and the gateway, an initial update key ( $k - i - k$ ) is required. Both the permanent key and initial update key are stored in non-volatile memory, such as EEPROM, on all devices. The update key ( $k_u$ ) is generated using an H-KDF for the next session. After a predetermined session time, the update key ( $k - u$ ) is updated by using the previous session information.

The session key update process is described in the following steps.

- a) *Step No1-Sensor to Gateway:* After the predetermined session time has elapsed, the IoT device will use the H-KDF of  $R-frame_i$  and  $k_p$  as a salt input to compute the new session key ( $k_u$ ). The current  $k_u$  is utilized as an H-KDF key. The IoT device will then transmit encrypted information to the gateway, including  $h(S_{id})$ , a 'key update' command to indicate the key update process, a new challenge K1, and M-A-C  $K_{i-k}(msg1)$ , all encrypted using  $k_p$ .
- b) *Step No2-Gateway to Sensor:* After receiving the key update message, the gateway decrypts it and checks if the predetermined session time has elapsed. If it has, the gateway calculates the next session key using the H-KDF of  $R-frame_i$  as an input, along with  $k_p$  as the input salt. The current session key is used as the HKDF key. The gateway then encrypts  $h(G_{id})$ , K1, K2, (seq-Nb<sub>i+1</sub>), and  $sesn_{time_{i+1}}$  using  $k_p$  and sends it to the IoT device.
- c) *Step No3-Sensor to Gateway:* After receiving the message from the gateway, the IoT device decrypts it and computes the M-A-C. If the computed M-A-C matches the received M-A-C, the IoT device authenticates the gateway. Then, the IoT device sends  $h(S_{id})$ , K1, K2, (seq-Nb<sub>i+1</sub>), and M-A-C  $K_{i-k}(msg3)$  using  $k_p$  to the gateway. Upon receiving the message, the gateway calculates

the M-A-C and checks if it matches the received M-A-C. If it does, the gateway authenticates the IoT device. This marks the completion of the key update process, and the new communication session begins.

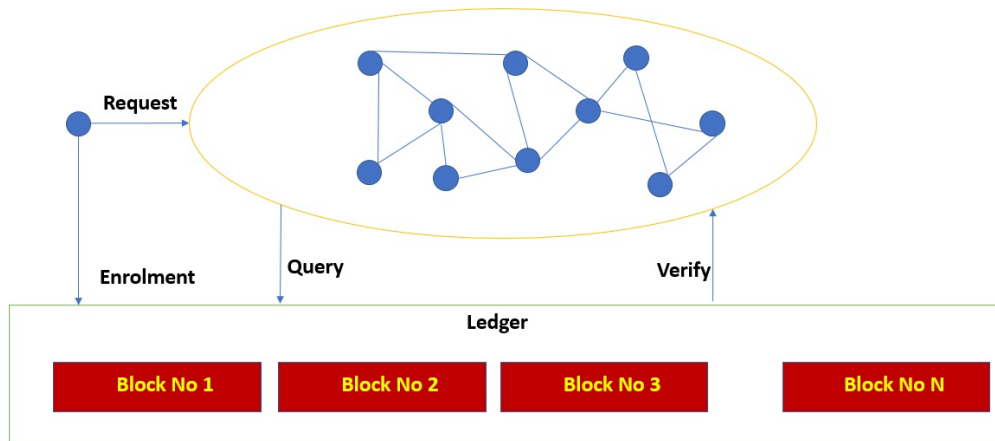
**The above communication and authentication protocol discussion answers Research Question 5.3.**

### 6.4.1 System Model

Blockchain technology is a secure, open, and distributed system that can adapt to complex and dynamic network environments. The system remains stable even if some nodes fail, as distributed authentication between nodes prevents malicious nodes from attacking the network. Additionally, even if some nodes are compromised, the ledger remains secure and cannot be tampered with. In a multi-node network, each device's identity information must be registered in the blockchain ledger. Hence it is chosen as a base for the application of DLT for the system model due to DL functionality and associated cryptographic functionalities. The system model is required to include the device's ID, public key, and a hash of critical data, among other information. Each device acts as a node in the blockchain network, and the consensus mechanism ensures that all nodes store the same information. When devices communicate with each other, public key cryptography is used for device identity authentication in the CoT environment as depicted in Figure 6.2. The system process flow involves three distinct steps, namely (i) Enrolment of the Device, (ii) Device Authentication, and (iii) Integrity Verification.

### 6.4.2 Security Analysis

A theoretical appraisal of security for the described mechanism is conducted to find its effectiveness against breaches. Our assessment is that incorporating two confidential keys, as done in MFA, enhances the robustness of the proposed protocol since it would be challenging for an attacker to decipher them. Possessing only one key would be insufficient to compromise the confidentiality and integrity of the messages. The security analysis indicates that the developed protocol can provide security features, such as protection



**Figure 6.2:** System Architecture of Blockchain-Enabled Device Authentication

against brute force attacks, data confidentiality, data integrity, man-in-the-middle attacks, mutual authentication, replay attacks, and impersonation attacks, as described in detail.

1. *Brute Force Attack:* A brute force attack is a type of cybersecurity breach where the perpetrator attempts to uncover the password or secret key to decrypt a message. However, in the proposed protocol, a brute-force attack is improbable for three reasons. Firstly, the secret keys' length is lengthy, making it difficult for an attacker to crack them using a brute force attack. Secondly, the secret keys are stored in the sensor and gateway's non-volatile memory (EEP-ROM) and are never transmitted over the network, making them challenging to obtain. Lastly, the attacker lacks information about the secret keys, making them challenging to guess. Furthermore, as the attacker cannot provide all the information required for mutual authentication, they will be unable to initiate the process to log into the sensor or gateway. Thus, the protocol is resilient to brute-force attacks.
  
2. *Data confidentiality:* Data confidentiality pertains to safeguarding transmitted data between nodes of communication, guaranteeing that only the sender and receiver can access and modify the message. To provide data confidentiality, we utilized two secret keys, which encrypt traffic from the sensor and gateway. These secret keys are never transmitted over the network, making it challenging for an attacker to obtain them. As a result, we can confidently assert that the protocol ensures data confidentiality.
  
3. *Data integrity:* Data integrity is the assurance that messages are received in the same

state they were sent, without any duplication, insertion, modification, or reordering. We employed the H-MAC function to ensure data integrity. Before transmission, both the sensor and gateway compute the M-A-C for the entire message. If the received M-A-C matches the computed M-A-C, they can be sure that the message was not tampered with during transmission. If an attacker intends to tamper with transmission messages, they must possess the  $k_u$  encryption hash key used during the session. Additionally, during mutual authentication, the attacker would require knowledge of the  $k_p$  and  $k_{ik}$  before tampering with the messages. These measures affirm that the protocol upholds data integrity.

4. *Impersonation attack*: In this type of attack, the attacker attempts to pose as a legitimate device. For an attacker to successfully impersonate a sensor or gateway, they would need to acquire knowledge of  $h(S_{id})$ , K1, K2, and  $h(G_{id})$  before launching an impersonation attack. All of this information can not be obtained without having access to the secret keys. As a result, we can conclude that the protocol is capable of withstanding impersonation attacks.
5. *Man-in-the-middle attack*: A man-in-the-middle attack is a cryptographic attack on a communication channel wherein an active attacker covertly takes control of a confidential communication channel between two parties. The attacker can intercept, read, modify, and replace the communication traffic between the victims. In the proposed protocol, if an attacker intercepts the communication traffic between the sensor and gateway, they must know both the permanent keys ( $k_p$ ) and session key ( $k_u$ ) to decrypt and modify data. However, even if an attacker knows  $k_p$ , they cannot calculate  $k_u$ , as it is dependent on the previous session key and random session number. Furthermore, the intruder cannot obtain  $k_u$  as the keys are never exchanged over the network. Even if  $k_p$  is discovered during mutual authentication, a man-in-the-middle attack cannot succeed as the attacker must learn the initial session key to prove their identity. As a result, an attacker cannot decrypt or modify data, and the protocol can resist man-in-the-middle attacks.
6. *Mutual authentication*: Before initiating a standard communication process, it is necessary to establish a secure mutual authentication between the two devices. When the mutual authentication process is successfully completed, a secure communica-

tion channel is established, and it becomes impossible for attackers to intercept data during this process. To ensure secure mutual authentication, the administrator manually inserts a pair of secret keys into the devices, which encrypt and hash the authentication credentials. Hence, the protocol achieves secure mutual authentication.

7. *Replay attack*: A replay attack is a type of cyber-attack in which an attacker intercepts legitimate messages exchanged during an authentication phase and later replays these messages to impersonate a legitimate party and establish an authentication session. In the proposed protocol, replay attacks are prevented by using  $h(S_{id})$ , K1, K2, and  $h(G_{id})$ . The receiver first verifies the validity of all this information before authenticating the sender. Moreover, once mutual authentication has been established between the sensor and gateway, all previous information becomes invalid and cannot be used to initiate a new authentication process. If an attacker attempts a replay attack, the gateway and the sensor can detect the invalidity of these messages, and authentication fails. Therefore, we can conclude that the proposed protocol is resistant to replay attacks.

**The above communication and authentication protocol discussion answers Research Question 2.**

### 6.5 Implementation and Evaluation

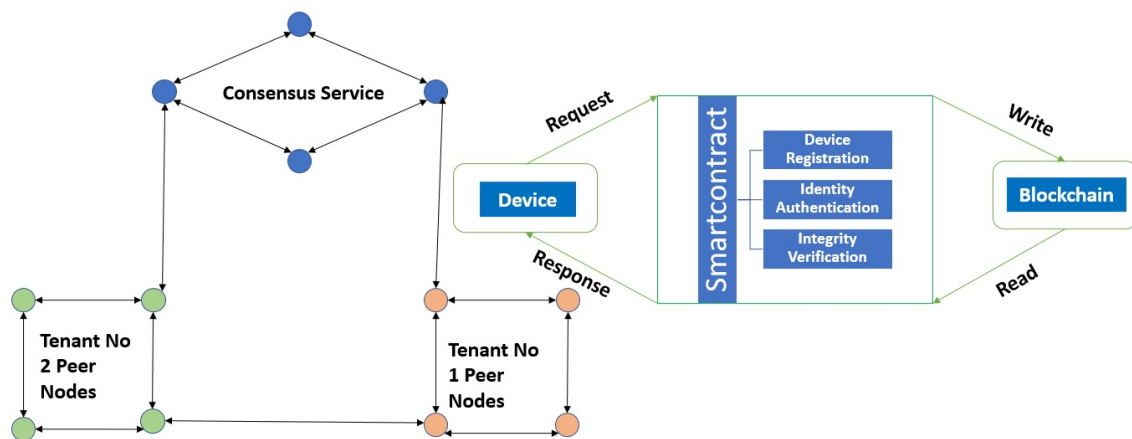
The system implementation involves various aspects, such as deploying the blockchain, generating asymmetric keys, and defining a storage structure for data to maintain its integrity in the blockchain. Additionally, the system security is evaluated in the form of a security analysis.

The independent functioning prototype was utilized in the GETH virtual machine of the open-source Ethereum platform. The smart contract is called using the Solidity v0.4.24 scripting language together with the GANACHE truffle suite, which facilitates the deployment of the smart contract on the blockchain platform. At the outset, the network's Gas limit was established as 4000000, and each of the four nodes created had 100 ETH in their account. The experiment was conducted on a Windows 11 operating system, with a

machine having an Intel i5 processor with a 2.8GHz clock speed, 8 GB of RAM, and a 1 TB HDD. Using this the B-MFA concept discussed in Chapter 4 has been implemented with MAC and digital certificate being used as separate factors for the MFA functionality to study their performance characteristics.

### 6.5.1 Deployment of the Environment

To meet the requirements of our IoT application, we opted to use a permissioned blockchain on the IoT cluster. Our implementation is based on deploying a blockchain network on sixteen numbers of Windows 98 based virtual machines, leveraging the Hyperledger Fabric open-source project. Each virtual machine serves as a node in the blockchain network, which is established in an ad-hoc manner. Thanks to the multi-chain and channel technology, the blockchain can be partitioned into several sub-chains, allowing IoT devices to create different subnets based on their business needs. A subnet can interact with its corresponding sub-chain with no interference from other subnets. Figure 6.3 illustrates the network structure of the blockchain along with the block configuration. All transactions occurring in the blockchain network are stored in the blockchain, and the data structure of the block is similar to that of Bitcoin, except that transactions refer to events such as device registration, identity authentication, and integrity checks. The stan-



**Figure 6.3:** Structure of the Blockchain & Block Configuration

alone test for blockchain and DID-enabled smartcontract was implemented first. This also used the hashed key of the device MAC along with held certified digital certificate. Subsequently, the testbed was implemented in a phased manner. Phase-I comprises of In-

Premise Cloud, and in Phase-II, AWS cloud infrastructure was integrated. In this phase, 16 nodes of Ethereum mapping was done over the EC2 instances, as System Architecture of Device MFA with Multi-Cloud Testbed as depicted in Figure 5.6. To depict the IoT Devices (Resources Constraint Devices with a sensor), a windows 98 based dual-core (1.0 G Hz) based processor with 1 GB of RAM have been virtualized. 16 number of such virtual machines with the OEM fitted webcam as the sensors have been emulated for the test environment in a multi-cloud environment.

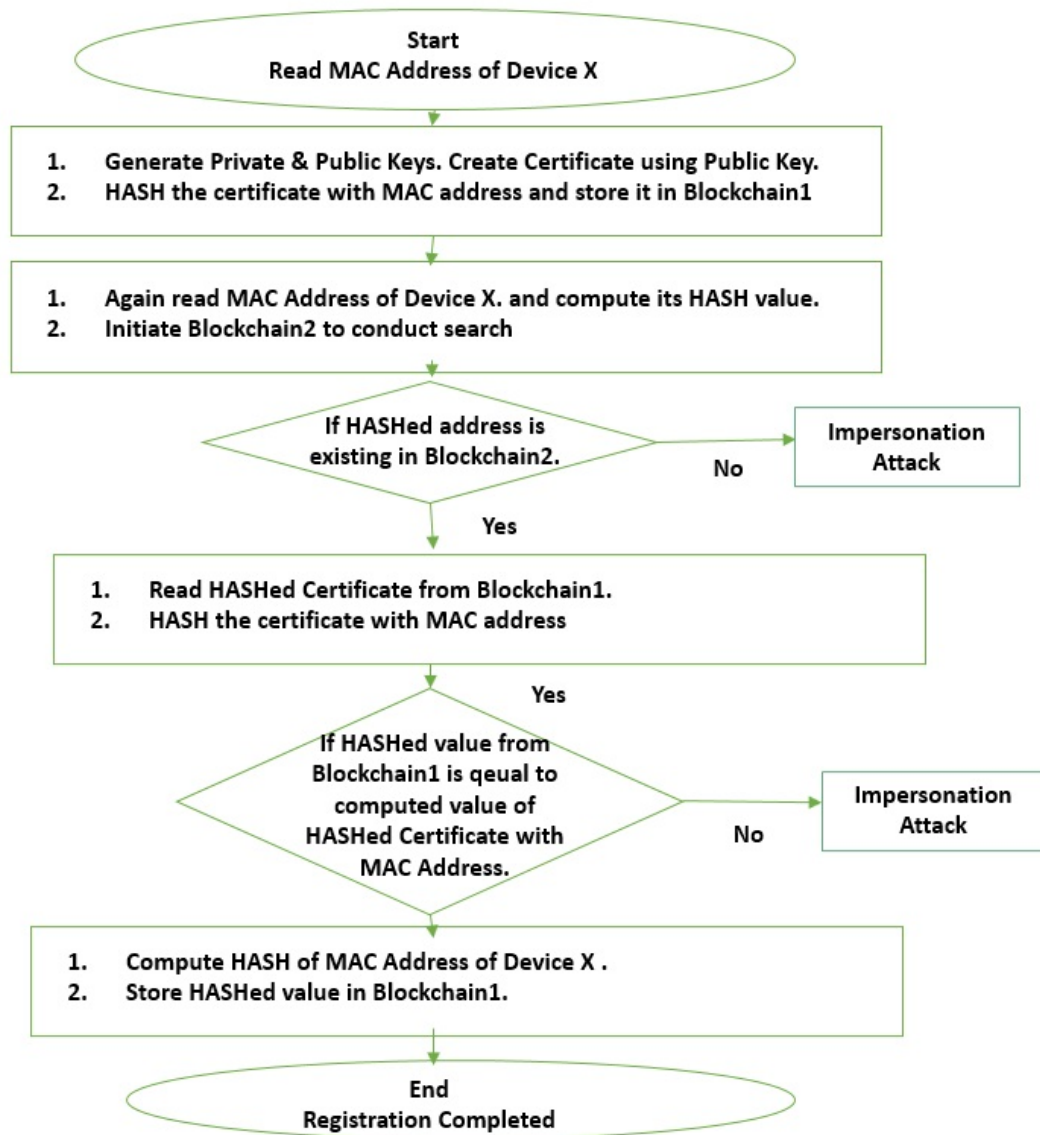
The communication between the devices and the blockchain occurs through transactions, which we classified into three types, namely (i) Device Registration, (ii) Device Login, (iii) Device-to-device Communication, and (iv) Integrity Verification. The transactions are requested and responded to by using smartcontracts. The smartcontracts receive requests from devices and execute distinct functions, such as writing and reading to/from the blockchain, based on the specific requests.

### 6.5.1.1 Device Registration

Figure 6.4 depicts the steps for registration of new devices into the network. First, the network administrator will read the MAC address of the device and generate public-private key pair. The private key generated for each device is unique, and the public key will be shared between the devices. The public key certificate is generated, and the hash of the certificate and MAC address is computed using SPONGNET. Store this in blockchain\_1. The MAC address of the same device is read again, and the hash is computed using SPONGNET. Hashed value is compared with every value in blockchain\_2. If it is already present, then it means a device with this MAC address exists in the network, and the device currently trying to register is using a falsified MAC address. This is an impersonation attack on the network, and the administrator will end the registration procedure for the current device. If the hash is not present, there is no reason to suspect the device's authenticity, so the administrator can perform the following steps. A Hash of MAC and certificate is produced and compared with the hashed value stored in blockchain\_1. If these values do not match, it is the case of an impersonation attack, and thus the process is terminated. If these values are equal, then the device is authenticated and authorized to



enter our network. The hash of the MAC address computed using SPONGNET is stored in Blockchain\_2. The relevant process is presented in Algorithm 4.



**Figure 6.4:** Device Registration Process

### 6.5.1.2 Device Login

After registration of the device is complete, it will need to log in to the network. First, the administrator will read the MAC address of the device and compute its hash value. This hash value is compared with every value in blockchain\_2. This is the first factor of authentication. If the hash does not exist in blockchain\_2, then the device is denied

entry into the network. Read the certificate of the device. Hash the certificate and the MAC address of the device. This is compared with all the values in blockchain\_1. This is the second factor of authentication. A schematic process for the same is depicted in Algorithm4. If the hash does not exist, the login procedure is terminated. If the value exists, then multi-factor authentication of the device is complete. Now, the device can connect to the network, being properly authenticated. The relevant procedure is presented in Algorithm5.

---

### Algorithm 4 Device Login Process

---

```

1: Initialize exists ← Boolean
2: Function Login(macAdd)
3: if (if-macInBlockchain2(macAdd))
4: read the certificate of the device
5: Endif
6: invoke SC to check if hash(cert+macAdd) exists in Blockchain1, exists ← 1 Else exists
  ← 0.
7: Endif
8: If(exists == 1)
9: Multi-Factor Authentication Successful.
10: EndIf
11: Else its Impersonation Attack Condition
12: OR Device Doesnot exist in the Network.
13: Multi-Factor Authentication NOT Successful.
14: EndElse and Exit

```

---



---

### Algorithm 5 Communication for Authentication

---

```

1: Initialize S, PT and CT ← NULL.
2: Function CommunicationSender()
3: if (livenessCeck(macAdd-A, macAdd-B))
4: S = diffieHellman()                                ▷ To generate session key
5: PT = concat(PT, timestamp)                          ▷ PT is Plain Text
6: CT = encrypt(S, PT)                                ▷ CT is Cipher Text
7: Function CommunicationReveiver()
8: S = diffieHellman()                                ▷ To generate session key
9: Receive CT
10: PT = decrypt(S,CT)
11: PT, timestamp = PT, split()
12: If (timestamp > threshold), Generate Alert        ▷ Generate new session key for new
  session

```

---

**This answers Research Question 3.**

### 6.5.1.3 Device to Device Communication

When Device A wants to communicate with Device B, it will broadcast request to network engine. Network Engine will check the availability of both devices using the liveness check. Both devices will generate public-private key pair using Elliptical Curve Cryptography (ECC). Schematic detail of the process is depicted in Algorithm 5 Device A will send its Public Key to Device B, and Device B will send its Public Key to Device B. Both devices will calculate Shared Symmetric Key

$$S = Pubkey(A) * PrivKey(B) = PubKey(B) * PrivKey(A) \quad (6.1)$$

This process is known as Elliptical Curve Diffie Hellman (ECDH) Key Exchange protocol [167]. This is a session key that will last till either of the devices is disconnected. When devices get reconnected new key will be created every Time. After key S is created, both devices can start communicating. When Device A wants to send a message, it will append the time stamp. This will be encrypted with key S. Device B will receive an encrypted message. It will decrypt it using the same key S which was previously computed. It will compare the timestamp with the current Time. If it is not within the given threshold, the network is under attack. It could be a delay introduced by an adversary listening to communication and forwarding a message to B. This is a man-in-the-middle attack. It could also be a replay attack, where the adversary sends a previously recorded message to get a response from Device B. If the timestamp is valid, Device B will send an acknowledgment back to A. In a similar manner, devices will communicate with each other.

### 6.5.1.4 Cryptographic Key

In the context of IoT, each device is associated with a key pair, consisting of a private key and a public key, which serves as the device identity. The private key is a randomly chosen number, while the public key can be obtained via elliptic curve multiplication. The most crucial aspect of private key generation is using a sufficiently secure entropy source to ensure the selected random number is both unpredictable and non-repeating. Crypto-

graphically secure pseudorandom number generators (PRNGs) are commonly employed to provide a source of randomness that meets these criteria. Unlike statistical and weaker PRNGs, cryptographically secure PRNGs generate pseudorandom numbers that possess additional properties of randomness. Using a cryptographically secure PRNG also necessitates a seed value derived from an entropy-rich source. In our system, we gather diverse information from emulated Windows 98 virtual machines, which serve as IoT devices, such as memory usage status, free hard disk space, I/O delays, the number of processes, CPU clock speed, etc., to obtain an approximate random seed.

The elliptic curve algorithm can be used to derive the public key from the private key. This process is one-way and irreversible, given by  $K = k * G$ , where  $k$  is the private key,  $G$  is the generator point constant, and  $K$  is the resulting public key. If an attacker attempts to find the private key  $k$  by discovering the discrete logarithm of the public key  $K$ , it becomes extremely challenging to identify the correct key by bruteforce searching through all possible values.

To authenticate a certificate using this technique, we adjust the OpenSSL's certificate verification function and send a request to the Ethereum blockchain instance, which is operated on a server located within the same local area network. This server is equipped with the Metamask wallet implementation. This is usually the most likely scenario in which a network administrator will have the blockchain node hosted on the same network as the IoT devices that perform the authentication. The request includes the certificate ID, and in response, we receive the certificate hash. Subsequently, we compare the hash obtained from the response with the calculated hash of the certificate that we are verifying. If the hashes match, the certificate is authenticated successfully.

### 6.5.1.5 Blockchain Mechanism for Data Integrity Verification

In blockchain-based systems, files are signed using distributed ledger technology, which provides stronger protection against data forgery and theft compared to traditional file signatures that rely on asymmetric key encryption. These systems use hash encryption techniques for integrity protection, with the security of the hash functions and the validity shared ledger is crucial for ensuring integrity. The storage structure of files in the

blockchain is described by the files that are first stored as a hash value and then combined to calculate a hash until a root hash value is obtained. This structure is known as a Merkle Tree (hash tree), and the path from the hash of each file to the root hash is called the file's signature. To verify a file's integrity, one only needs to insert the hash of the file to be verified into the signature path and compare the resulting root hash value with the original hash value.

### 6.5.2 Threat Modeling Performance Analysis

The performance metrics of the system, such as throughput and delay, are mainly influenced by the blockchain platform used, which in this case is Hyperledger Fabric. Therefore, the specific details regarding performance are expected to align with the performance demonstrated by Hyperledger Fabric. For the consensus algorithm, the system uses PBFT (Practical Byzantine Fault Tolerance), which allows for the detection of abnormal behavior and the synchronization of data in the ledger to achieve consistency in the blockchain network. The security of the system depends on the strength of PBFT. In a Byzantine fault tolerance system with  $n$  nodes, if the number of traitors (fault nodes or colluders of attacker) is  $t$ , as long as  $n > 3t$ , the agreement will conclude within a limited time, and the honest nodes will ultimately reach a consensus.

#### 6.5.2.1 Threat Modeling Analysis

The proposed approach for secure authentication is found to be effective in the following scenarios:

1. *Brute force*: An attacker attempting a brute force attack on a 256-bit elliptic curve cryptography (ECC) key would need to try an astronomical number of possible key combinations - specifically, 2 to the power of 256 different combinations. Compared to traditional RSA keys, ECC keys have an advantage in that a 256-bit ECC key is equivalent in strength to a 3072-bit RSA key. However, even with a relatively small key size, it is impossible to discover the key through brute force methods.
2. *Man in the middle*: The attack involves the interception and alteration of messages.

Since each message is encrypted with the recipient's public key, the attacker is unable to modify its content. Additionally, the hash of the data is included in each packet, so any modifications made to the message can be detected by comparing the received hash to the calculated hash.

3. *Replay attack*: This is a variant of a Man-In-The-Middle attack, where the attacker may or may not alter the message content. Instead, the attacker holds onto the message for a period of Time before sending it to the receiver, who believes it to be a new message. However, if time stamping is used in each data packet, this type of attack can be prevented.
4. *Identity spoofing*: Considering node A has the ability to falsely assert that it is node B and can ask for an update of the public key/identity. However, this can be prevented by checking that the IPv6 address registered with node A is different from the one belonging to node B. Therefore, if the IPv6 address claimed by node A does not match that of node B, public key/identity spoofing cannot be carried out. Another method to address this issue is to utilize a challenge-response mechanism.
5. *DDoS attack*: The issue can be addressed by utilizing a decentralized peer-to-peer communication network and a strong authentication protocol. Furthermore, private keys are utilized for device login, making it impossible for anyone to gain unauthorized access through brute-forcing keys. Any firmware updates for the device are signed and transmitted by the trusted manufacturer registered with the blockchain network.
6. *Sybil attack*: The attack entails a single entity asserting multiple identities. However, in our proposed system, this is not possible since every node must register its identity and IPv6 address with the blockchain. Therefore, if any node claims an identity that doesn't match its own, it can be detected by querying the ledger and utilizing a challenge-response mechanism similar to the one used for the IPv6 spoof attack.
7. *Malicious Insider attack*: It is possible for the validator node to act dishonestly and send an incorrect secret share to the IoT node, which would result in an erroneous computation of the private key. However, this scenario is prevented because the IoT node has the ability to verify each secret share it receives.

8. *Single Point of Failure*: This proposed system operates in a decentralized manner where the ledger is distributed among all the validator nodes responsible for maintaining the blockchain. Hence it is not a single point of failure-prone system.
9. *Loss of private key*: In the event that the private key of the IoT node is lost, a key request transaction could be initiated. This would allow all the validators to send their shared secrets to the node, enabling it to recompute its private key.

### 6.5.2.2 Liveness Check

In a public IoT network like that of a Smart-City IoT network, sensor devices are mostly deployed in a public place, having relatively less provisioning of physical security. Hence there is a chance of malicious swapping of the device by an attacker, which could lead to a chaotic situation in the functioning of the IoT network. Accordingly, even if the device is successfully registered and logged in, there is still a requirement for its liveness check to ascertain the following to ensure smooth and secure access of the device in the IoT Cloud. This is designed to detect spoofing attacks as well as device impersonation attacks.

1. If the device is powered off or the device battery is drained out.
2. Any fault with the device networking adopter or interface.
3. If the genuinely registered device is maliciously swapped with a faulty one or even embedded with malicious software.

There is a requirement to detect a non-networked device , if it malfunctions or even powered off due to any reason, to take it out of the actively logged in devices list. To ascertain and re-authenticate for device genuineness, there is a requirement for periodic or regular re-authentication of the registered devices. To accomplish this task liveness check is done with various test conditions. A schematic view of such test cases is depicted in Figure6.5. **This answers Research Question 4.**

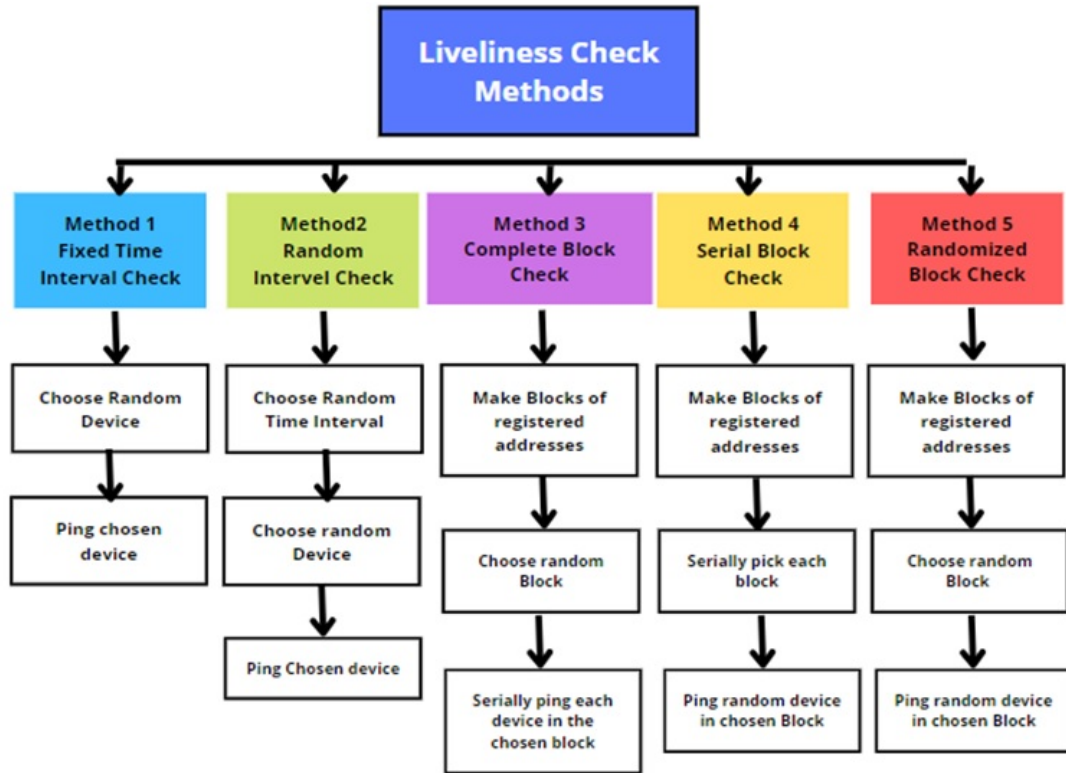


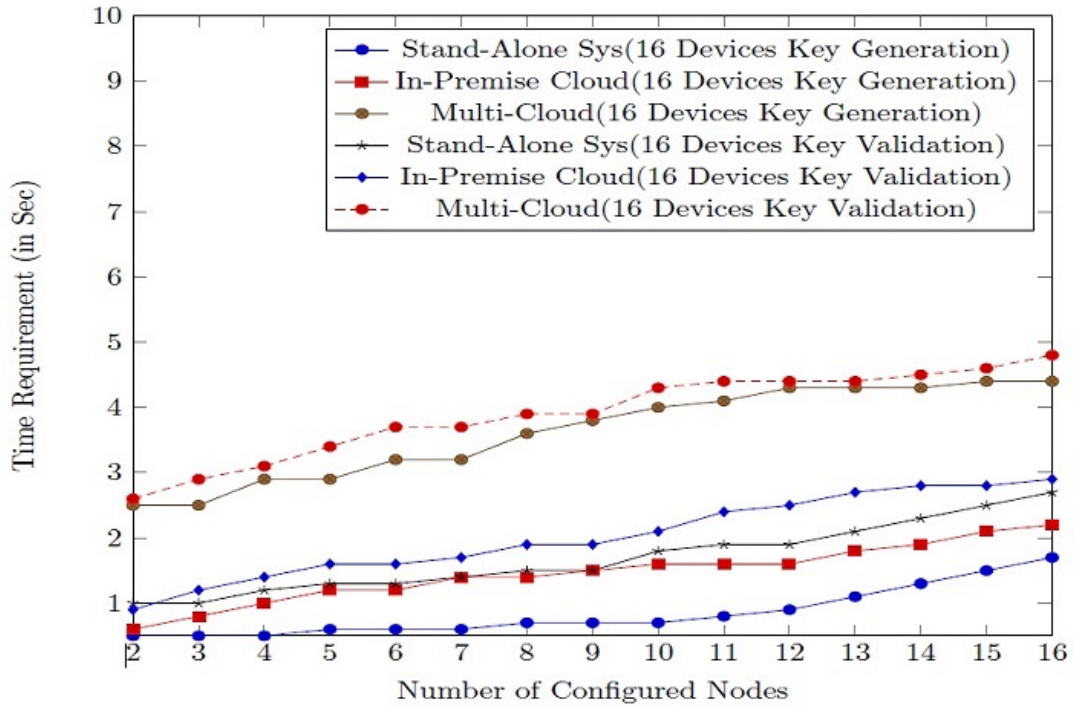
Figure 6.5: Test Cases for Liveness Check of IoT Devices

### 6.5.2.3 Performance Analysis

Initially, the system of secure device authentication was deployed in a standalone machine to check the operational requirements. After successful verification of the operation of the designed system, the same was ported to the in-premise cloud, and in the subsequent phase, the proposed secure device authentication solution was deployed in over multiple commercially available clouds forming multi-cloud testbed having 16 Blockchain nodes each for Blockchain1 and Blockchain2. The Devices of CoT were emulated by lower resource-configuration Windows 98 based PCs with OEM-fitted webcams.

System performance in terms of Key Generation and Key validation is found out to be a key factor towards system efficiency. Timing taken at various stages of implementation infers that the present system being implemented is scalable and consistent in terms of Time -performance. A detailed analysis of time performance for Key generation and Key validation is depicted in Figure 6.6. Similarly, the authentication time taken by this device MFA approach has been studied. Time Taken for all three test scenarios, namely standalone, in-premice MFA, and also multi-cloud MFA doe device authentication,





**Figure 6.6:** Key Generation and Key Validation Performance Analysis

have shown consistency in time performance, tested with 16 concurrent login devices in a progressive phased manner. Details of this time analysis are depicted in figure 6.7. All the test cases of Liveness check have been carried out in a multi-cloud environment comprising 18 PCs depicting as 18 devices of the CoT, and their performance analysis in terms of re-authentication times is depicted in Figure 6.8. This liveness check is virtually implementing the concept of C-MFA and D-MFA discussed earlier in the context of device MFA. The above analysis of C-MFA and D-MFA of Device authentication infers that the second method of the test case (*Random Interval Check*) is most optimal, considering the time complexity of our researched solution. Experimental results show that IoT chain can maintain high throughput and effectively reach consensus in a distributed system.

**The above modelling and implementation answers Research Question 5.**

## 6.6 Chapter Summary

The specialty of device authentication in a CoT environment lies in providing a secure and efficient authentication mechanism that can handle a large number of devices with differ-

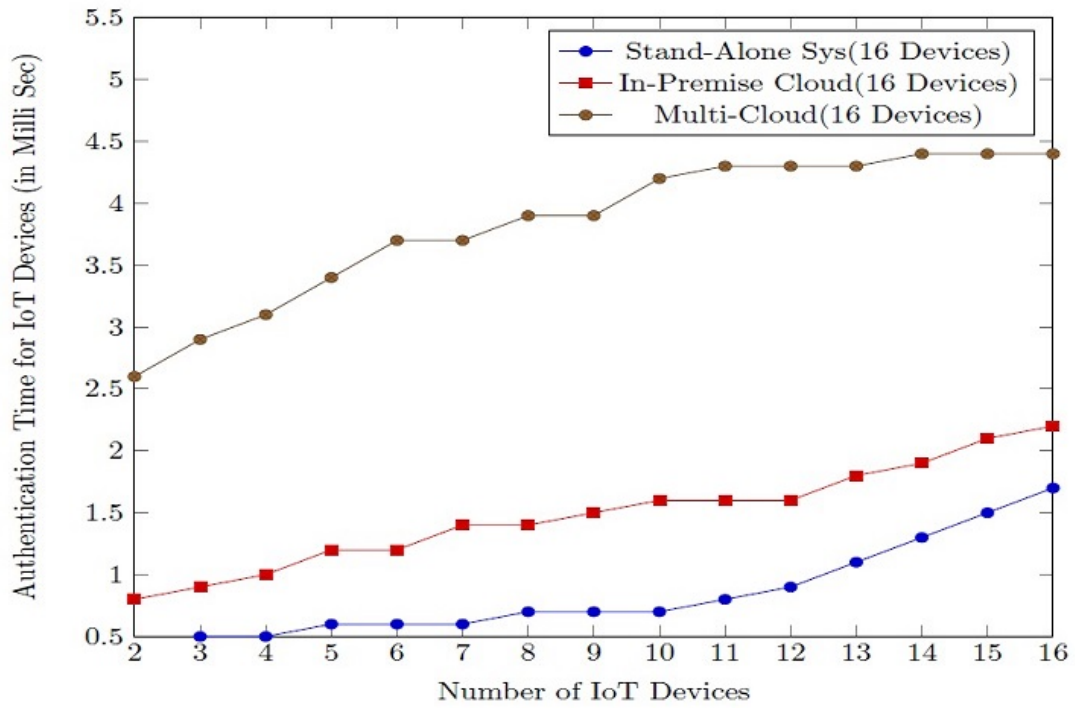


Figure 6.7: Authentication-Time Performance Analysis

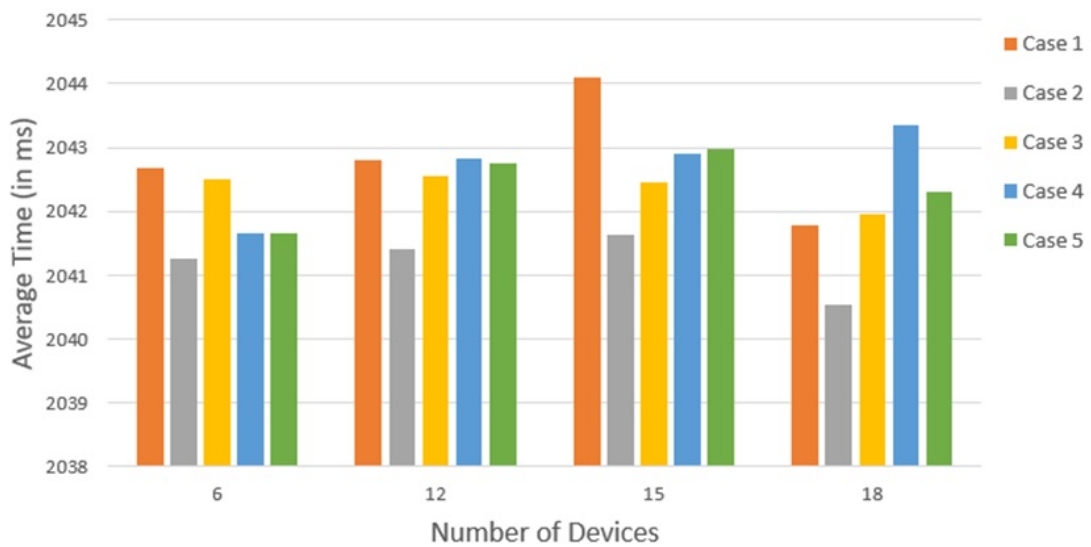


Figure 6.8: Time Interval for Test Cases of Liveness Check

ent capabilities and security requirements. This requires a combination of robust identity management, mutual authentication, strong encryption, and scalable authentication protocols. In the limitation of laboratory conditions, 18 number of devices were subjected to various test conditions. A secure authentication protocol has been proposed and validated with different conditions for device registration, mutual authentication, and data communication. DLT has been incorporated to take inherent cryptographic security advantages. Smartcontract have been deployed over two different blockchain in an integrated manner. The concept of Blockchain Enabled MFA (B-MFA) was initially tried on a standalone system. Subsequently, the concept of Continuous MFA (C-MFA) is implemented as the first test case for the liveness check. Subsequently, Dynamic MFA (D-MFA) for device authentication has been implemented in a multi-cloud environment for verifying liveness in terms of four different test scenarios. The system performance in terms of scalability consistency has been inferred. With uploading of 4MB video MP4 file from the sensor webcam to the cloud storage and downloading of such files from cloud storage to Windows 98 based, resource constraint machines have demonstrated data confidentiality, integrity, and availability.

# 7 Augmented Multi-Factor Authentication

## 7.1 Introduction

The authentication process is critical in today's world, where cyber attacks are becoming more sophisticated and frequent. In the modern era, trustworthiness is essential for making synchronization with technical advancements in this direction. To maintain trust, there are several important concepts, such as Authorization, Authentication, and data values. Nowadays, information is critical and often stored digitally, so it's essential to be cautious about the environment in which we access the data and ensure that the person accessing the data is authorized and authentic. Identity and Access Management (IAM) is a process that involves Authentication and Authorization, and it is primarily used to ensure that people follow certain rules or have specific permissions to keep information safe.

The Zero Trust Network (ZTN) as a concept can potentially do value addition to IAM, which operates with a different approach to identity and accessibility. In this approach, the concept of "trust" is completely eliminated from the network. The network is viewed as potentially untrustworthy and compromised, and every access request must be verified. Combined with ZTN, Software Defined Perimeter is a net segmentation methodology that can effectively split a relatively larger network into smaller network segments, which can logically represent several domain-specific network. Such segmentation paves the way for better and more efficient enforcement of access control policies.

Augmented authentication is a modern approach to identity verification that combines multiple factors to increase security and accuracy. Augmented authentication is becoming increasingly important as more and more sensitive information is stored online and cyber threats continue to evolve. It offers a way to stay ahead of hackers and other malicious

actors and provides a more secure and user-friendly experience for individuals and organizations. Hence an innovative technical amalgamation of IAM and ZTN, along with SDP, is a potential means for enhancing the identity process of users or devices with an umbrella of trust over the desired confidentiality, integrity, and availability (CIA) requirements.

## 7.2 Background Literature Study

Before the actual implementation of ZTN and SDP concepts for this research, a brief review of the background processes and their interoperability are considered.

### 7.2.1 Augmented Multi-Factor Authentication

Augmented Multi-Factor Authentication (A-MFA) is considered an advanced security technology that combines traditional multi-factor authentication methods with additional layers of security to provide enhanced protection against unauthorized access to sensitive data and systems. As a concept, A-MFA uses a combination of multiple authentication factors, such as passwords, biometric data (such as fingerprints, facial recognition, or voice recognition), smart cards, tokens, and contextual factors (such as the user's location, device, or behavior) to verify the identity of the user. In addition, it also incorporates advanced technologies, such as artificial intelligence, machine learning, and behavioral analytics, to detect and prevent fraudulent activities. This includes analyzing user behavior patterns and detecting anomalies in real-time to identify potential security threats. In addition, evolving technological approaches like Zero Trust Architecture (ZTA), ZTN and SDP are also potentially viable means to further enhance the effectiveness. This way, A-MFA is expected to provide an additional layer of security to the MFA process.

### 7.2.2 Zero Trust Architecture

The National Institute of Standards and Technology (NIST) has published a report on Zero Trust Architecture (ZTA) [168] which states that ZTA is not a single network architecture that can be achieved with a single technology. Instead, it is a collection of guiding

principles that must be strategically implemented to protect enterprise assets, including data, devices, users, and other infrastructure components. The key principles for achieving ZTA are authentication and access control, which establish the user's identity and determine privileges for various operations involving protected resources.

To implement ZTA in a critical infrastructure context, a strong authentication scheme that identifies both users and devices is required. Additionally, rather than relying solely on entry-point authentication, a context-aware and continuous authentication scheme that considers the user and device contexts on an ongoing basis is recommended. Risk-aware access control schemes that assess the risk associated with access requests should also be part of ZTA strategies.

In addition to these primary principles, ZTA implementation requires the use of lightweight encryption schemes for resource-constrained devices in cyber-physical systems. NIST also recommends micro-segmentation and software-defined perimeters as core ZTA implementation strategies, although customization is necessary to secure the edge network of IoT devices. Threat intelligence is also critical, as it can be used as a feedback mechanism to drive automated security technologies in the defense environment. A customized and reliably responsive system is necessary for continuous trust evaluation and access control.

### 7.2.3 Zero Trust Network

Zero Trust Network is a security concept that assumes that every user or device that connects to a network is potentially hostile and thus must be authenticated and authorized before being granted access to any network resource. Zero Trust Network can effectively contribute to maintaining user or device identity in an authentication system [169–172] by implementing the following measures:

1. *Continuous authentication:* Zero Trust Network continuously monitors user and device activity on the network and authenticates them based on their behavior. This helps to prevent unauthorized access to network resources.
2. *Device Authentication:* By requiring devices to authenticate themselves before ac-

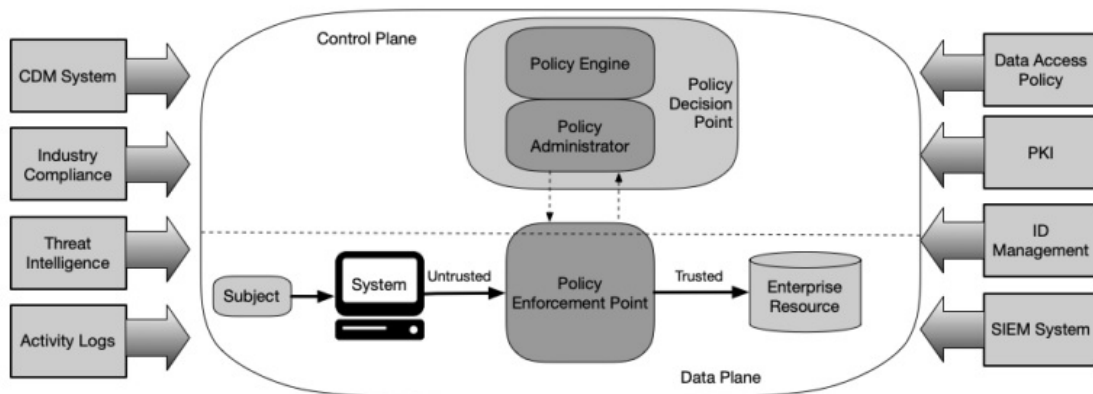
cessing the network, zero trust authentication can help ensure that only trusted devices are granted access to the network.

3. *Identity and Access Management (IAM)*: By implementing a centralized system for managing user and device identities, zero trust authentication can help ensure that access rights are granted only to authorized users or devices.
4. *Least privilege access*: Zero Trust Network restricts access to network resources based on the principle of least privilege. This means that users and devices are granted access only to the resources they need to perform their authorized tasks and nothing more.
5. *Multi-factor authentication*: Zero Trust Network requires multi-factor authentication for all users and devices before granting access to any network resource. Multi-factor authentication involves using two or more authentication factors, such as a password and a smart card, to verify the identity of a user or device.
6. *Network segmentation*: Zero Trust Network segments the network into smaller, more secure zones, with each zone having its own security controls. This helps to limit the damage caused by a security breach, as the breach will be contained within the affected zone.

A Zero Trust Network can be implemented using a combination of various technologies and techniques such as multi-factor authentication, identity and access management, network segmentation, micro-segmentation, encryption, continuous monitoring, threat intelligence, and automated response. The implementation of a Zero Trust Network involves a comprehensive security strategy that requires ongoing assessment, testing, and optimization. Organizations need to consider their unique security requirements, the types of assets they need to protect, and the risks they face. They also need to ensure that all security solutions and practices work together to create a cohesive and effective security posture. Basic fundamental steps to be followed for a Zero Trust Network implementation as depicted with its components in Figure 7.1. \* can be summarized as follows:

---

\*<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf>



**Figure 7.1:** Zero Trust Network Components

- (i) Authenticate all users, devices, and applications that request access to network resources.
- (ii) Verify the identity of users, devices, and applications using multi-factor authentication and other identity verification techniques.
- (iii) Authorize access to network resources based on the principle of least privilege.
- (iv) Segment the network to isolate sensitive data and applications and limit lateral movement by attackers.
- (v) Continuously monitor network activity, including user behavior, network traffic, and device behavior.
- (vi) Detect and respond to security incidents in real time using automated response and threat intelligence.
- (vii) Implement ongoing assessment, testing, and optimization of security controls, policies, and procedures to ensure continued effectiveness.

### 7.2.4 Software Defined Perimeter

Network segmentation is the practice of dividing a network into smaller subnetworks or segments, each with its own security policies and access controls. This approach can



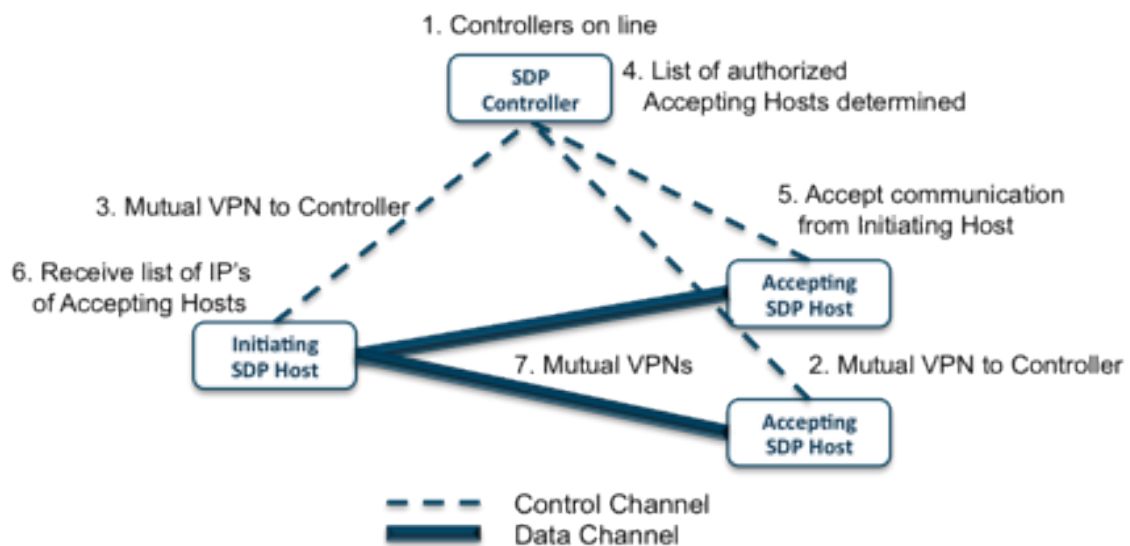
help improve network security by limiting the scope of a potential breach and reducing the impact of a security incident. Software-defined perimeter (SDP) is a security framework that allows organizations to create secure, segmented network environments using software-defined networking (SDN) technologies [168,173,174]. SDP provides a flexible and scalable way to achieve network segmentation by enforcing identity and application-level access control, creating dynamic access, following zero-trust security principles, and using encryption to protect network traffic. By implementing SDP, organizations can enhance their network security and reduce the risk of unauthorized access or lateral movement by attackers. SDP can help achieve network segmentation by following these steps:

1. *Application-level access control*: SDP can be configured to provide access control at the application level, allowing organizations to limit access to specific applications, servers, or services, and to ensure that only authorized users or devices can access them.
2. *Create network-segment security policies*: SDP follows a zero-trust security model, which means that all access attempts are considered untrusted until they are verified and authorized. This helps ensure that only legitimate users or devices are granted access to the network. Based on the resource classification, access control policies can be created. SDP policies can be granular and based on a user's identity, device security posture, or location.
3. *Deploy the SDP controller*: The SDP controller is a central component that manages network traffic and enforces access control policies. It is responsible for authenticating users and devices and assigning them to the appropriate segments of the network.
4. *Dynamic access*: SDP creates a dynamic and secure connection between users and resources, which means that access is only granted when it is needed and for the duration of the session.
5. *encryption*: SDP uses encryption to protect network traffic, which means that data is secure even if it is intercepted by attackers.
6. *Enforce traffic filtering*: To further improve network security, SDP can be used to

filter traffic between segments. This can be achieved by configuring firewall rules or using a network overlay to encapsulate traffic and limit visibility across network segments.

7. *Identity-based access control*: SDP provides granular access control to resources based on user or device identity, making it easier to segment the network and control who has access to what.
8. *Identify the resources to be protected*: The first step is to identify the critical resources that need to be protected, such as sensitive data, applications, or servers.

The fundamental process of Software Defined Perimeter emphasizes dynamic, on-demand access controls that grant or deny access to network resources based on user and device attributes and the use of a secure connection broker that serves as the gateway to network resources. The architecture provides a highly secure and flexible way to protect network resources from unauthorized access while also allowing authorized users and devices to access those resources as needed. Essential steps to be followed for an SDP implementation as depicted in Figure 7.2<sup>†</sup> can be summarized as follows:



**Figure 7.2:** Software Defined Perimeter

- (i) Identify and inventory all network resources, including those that are not managed by the organization.

<sup>†</sup>[https://en.wikipedia.org/wiki/Software-defined\\_perimeter](https://en.wikipedia.org/wiki/Software-defined_perimeter)

- (ii) Categorize network resources based on their sensitivity, criticality, and value to the organization.
- (iii) Implement a secure connection broker that serves as the gateway to network resources.
- (iv) Use a policy engine to dynamically grant or deny access to network resources based on user, device, and resource attributes. Authenticate users and devices using multi-factor authentication and other identity verification techniques.
- (v) Authorize access to network resources based on the principle of least privilege.
- (vi) Implement network segmentation and micro-segmentation to limit lateral movement by attackers.
- (vii) Encrypt all network traffic to protect against interception and eavesdropping.
- (viii) Continuously monitor network activity, including user behavior, network traffic, and device behavior.
- (ix) Implement real-time threat intelligence and threat-hunting capabilities to proactively identify and mitigate security threats.
- (x) Continuously assess and update security controls, policies, and procedures based on new threats and changing organizational requirements.

### 7.3 Research Gaps and Questions

ZTN and SDP approaches have emerged as promising solutions to enhance the security of user and device authentication. By implementing stronger authentication mechanisms, monitoring user behavior, and enforcing access controls, ZTN and SDP can help organizations enhance their overall security posture and protect against a wide range of security threats. Some of the potential areas in user or device authentication that could be addressed using ZTN and SDP include:

1. *Biometric authentication*: Biometric authentication uses a physical characteristic of the user or devices, such as their fingerprint or facial recognition or Media Access Control (MAC) address, to verify their identity. Biometric authentication can be highly secure, but it can also be subject to spoofing attacks.
2. *Device and User trustworthiness*: Traditional authentication methods do not consider the trustworthiness of the user and device, which makes it vulnerable to attacks such as device spoofing and tampering. ZTN and SDP can address this gap by implementing device or user attestation, which verifies the integrity and security of the device before granting access to network resources.
3. *Integration with legacy systems*: Many organizations have legacy systems that may not be compatible with modern authentication mechanisms. ZTN and SDP can help address this by implementing secure access controls that can integrate with legacy systems.
4. *Insider threats*: Traditional authentication mechanisms are designed to prevent external attacks, but they may not be effective against insider threats. ZTN and SDP can help address this by monitoring user behavior and access patterns to detect potential insider threats.
5. *Security of cloud-based applications*: Cloud-based applications are becoming increasingly popular, but they are also vulnerable to security threats. ZTN and SDP can help address this by implementing secure access controls and monitoring user activity within cloud-based applications.
6. *Weak authentication mechanisms*: Traditional authentication mechanisms like passwords and PINs can be easily compromised, leading to unauthorized access to network resources. ZTN and SDP can address this by implementing stronger authentication mechanisms like multi-factor authentication (MFA) and biometric authentication.
7. *User or Device location*: Traditional authentication methods do not consider the user's location, which can lead to unauthorized access if the user is accessing the network from an untrusted location. ZTN and SDP can address this gap by imple-

menting geolocation-based authentication, which verifies the user's location before granting access to network resources.

### 7.3.1 Research on ZTN and SDP

From the beginning, the concept of ZTN and SDP has been proposed by Cloud Security Alliance (CSA) [175]. ZTN and SDP are emerging as potential areas of application for user and device IAM. Many researchers have explored possible areas of research for user and device authentication in a cloud environment. Virtualized network testbeds were used by Sallam et al. [176] to analyze the combination of SDP and smart-home infrastructure. The testing demonstrated that SDP has the ability to reason and counteract various types of attacks on smart-home systems, including flooding, spoofing, intrusion detection, and eavesdropping.

Abdallah et al. [177] proposed an SDP-based framework adopting a client-gateway architecture. The performance of the proposed approach was also evaluated using a virtualized network testbed for an internal enterprise scenario as a use case.

Singh et al. [178] suggest using a software-defined perimeter (SDP) as a solution for securing IaaS. SDP creates a logical boundary that limits access to services, using authentication and authorization to permit only authorized clients to connect to services that are hidden behind SDP gateways. The effectiveness of SDP in an AWS cloud environment was confirmed through implementation and verification, including the use of port scanning to test SDP behavior.

Sudakshina et al. [179] proposed an approach of an algorithm of *multiplying increases and adding decreases* and utilized to identify a sophisticated MAC spoofing attack prior to gaining access to the cloud resources based on SDN. Under this approach, a dynamic threshold is allocated to the incoming port number. The threshold marking's ability to self-learn aids in correcting genuine user traffic before classifying it as malicious. The mathematical and experimental outcomes demonstrated greater accuracy and detection rates.

To meet the lightweight requirements in a UAV swarm, authentication technologies, and

the special requirements of cyber security, Dongyu et al. [180] proposed an authentication scheme based on Zero Trust, which achieved rapid authentication of UAV swarm in data exchange and sharing and strengthened its ability to respond to cyber-attacks.

Vivin et al. [181] suggested creating a system that gathers a group of different characteristics, such as the actions of the user, the qualities of the application being accessed, and the device being utilized. This expanded collection of information makes it possible to generate a thorough understanding of the situation, which can then be used to calculate precise variations and assess the level of risk involved.

Details of the referred research are consolidated in Table 7.1 to highlight the observed shortcomings. A Lightweight Continuous Device-to-Device Authentication is proposed by Shah et al. [169]. This research has emphasized the requirements and advantages of continuous authentication. Hence there is a need to develop and implement Continuous and Lightweight Authentication protocol for devices compatible with ZTN and SDP infrastructure.

Singh et al. [178] proposed a Software-Defined Perimeter (SDP) Architecture for Infrastructure as a Service. Abdallah et al. [177] brought out that a single SDP controller is vulnerable to DoS attack. Hence there is a need to have multiple SDP controllers in the SDP network with load-balancing functionality.

Considering the shortcomings noted from Table 7.1 and the preceding discussion, the following research questions are considered to be answered to mitigate the current gaps noticed.

1. RQ7.1. How to apply the concept of MFA with a suitable biometric factor of authentication for users and devices using ZTN architecture on a multi-cloud setup?
2. RQ7.2. How to conduct intra and inter-segment device discovery for their mutual authentication in a ZTN-enabled segmented network with SDP having Continuous

**Table 7.1:** Prominent Observations on ZTN & SDP Research

Ref	Resource Typr	Model/Algo Used	Applica- bility	Shortfall
[176]	Set of virtual machines hosted by a physical machine running Linux Ubuntu 18.04 and Virtual-Box 5.2	MQTT Protocol, Waverly Labs? Open SDP project	VMs, desktop, laptops	Large computing resources, excessive setup.
[177]	The components were set up on VMs and an MYSQL database was also present.	Waverly Labs Open SDP project was used to implement the framework.	Virtual Machines	1. Possible Network Disruption. 2. SDP Controller becomes vulnerable and prone to many attacks.
[178]	AWS-EC2 VMs is used to deploy the gateway and controller on separate instances, both with Linux 16.04xenial images. A VPC was configured to block all traffic to the controller and services, only allowing traffic into the gateway.	Open SDP by Waverly Labs, Amazon Web Services	VMs, desktop, laptops	Heavyweight process, not suitable for lightweight such as IOT devices.
[179]	IP addresses, DDoS controller	AWS or Microsoft web directory cloud services. Multiplicative Increase and additive decrease algorithms were used.	Mobiles, tablets, desktops, laptops.	Identity manager is required for validation of individual network traffic.
[180]	Drone swarms as network system, Ground control station.	Static swarm, dynamic swarm, and hybrid swarm.	Drones	Large computing resources are costly, so group cooperation is needed.
[181]	Done on contextual data. Attribute set, which consists of User_Attribute, Application Attribute, Device Attribute, Auxiliary Information, and Storage schema.	SVM or Naive classifier depending on the attribute set taken.	Mobile devices, Web browsers, and Desktop applications.	1. Attributes may not be available sometimes. 2. It does not perform well when we have a large data set because the required training time is higher.

Lightweight Device to Device authentication ?

3. RQ7.3. How to prevent DoS attack on SDN controller?

### 7.4 Proposed Approach

The infrastructure for managing identities plays a crucial role in enabling a zero-trust architecture with identity at its core. Its main functions include unified authentication of network entities, issuing public key certificates, and managing the life cycle of identities. The zero-trust model can be applied in various scenarios, including remote offices, cloud computing platforms, big data centers, and the Internet of Things (IoT) [182].

#### 7.4.1 MFA in ZTN

Continuous authentication has been integrated with other factors like user biometric characteristics and also user device characteristics like MAC address. Similarly, entitled applications earmarked to be used by the authenticated user is also to be mapped with user authentication credentials. In this scenario, there is a theoretical amalgamation of MFA and ZTN. This needs to be explored in detail in light of continuous trust evaluation.

##### 7.4.1.1 Design and Development of ZTN Architecture

The broad functioning of ZTN includes user or device identification. At this level, the MFA functionality needs to be incorporated. For ZTN design, the pseudocode is presented in Algorithm 6.



---

### Algorithm 6 Zero Trust Network ( ZTN)

---

```

1: Switch (s)
2: Case 1. Define Function Authenticate (User, Device):
3: If (User,Device → (Identified and Registered) and authorized):
4: Return True. Else Return False.
5: EndIf
6: Case 2. Define Function Authorize (User, Device, Resource):
7: If (User, Device is authorized for resource):
8: Return True. Else Return False.
9: EndIf
10: Case 3. Define Function Connect-Gateway (User, Device):
11: If (User is authenticated):
12: If (Device is Authorized):
13: Create Secure Connection to Gateway.
14: Return True. Else Return False.
15: EndIf
16: EndIf
17: Case 4. Define Function Enforce-Policy (User, Device, Resource):
18: If (User, AND Device, AND Resource is authorized)→ Grant access.
19: Else: Deny access.
20: EndIf
21: Else: Deny access.
22: EndIf
23: Else: Deny access.
24: EndIf
25: Case 5. Function Monitor-Network():
26: For each User, Device and Resource:
27: Record Activity and Behavior.
28: For Network Segment:
29: Limit access to authorized Users and Devices.
30: Case 6. Function Detect-Threat(network):
31: For each recorded event:
32: Analyse anomalies and potential threats.
33: Trigger alert or response, as required.
34: Case 7. Function Encrypt-Traffic():
35: Encrypt all network traffic with encryption mechanism.
36: Case 8. Function Optimise-Policy(policy):
37: Continuously access and update security controls, policies and procedures based on
   new threats and changing organizational requirements
38: Case 9. Define_security_policy(resources, user_roles, authentication, authorization)
   segment_network()
39: implement multifactor authentication (MFA) and implement(access_based_on_roles)
40: monitor_and_log_traffic()
41: implement_continuous_security(policy_review_and_update, security_assessments)
42: implement_network_security_tools
43: while (true)
44: if (access_request)
45: authenticate_user(device, application)
46: if (authorized_user)
47: verify_authorization(access_based_on_roles)
48: if (access_granted) grant_access_to_resource()
49: else deny_access_to_resource()
50: else deny_access_to_resource() and Exit
51: EndCase
52: EndSwitch

```

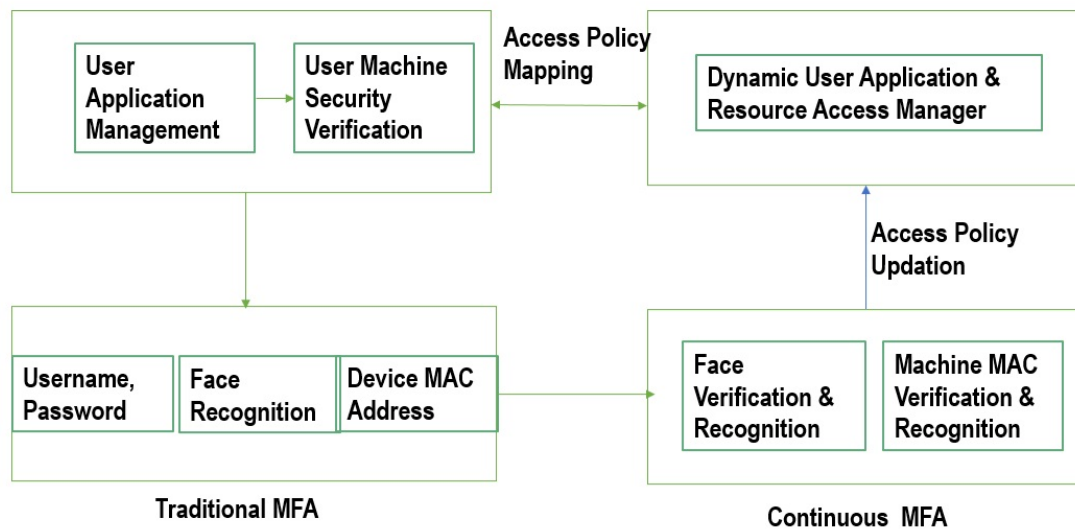
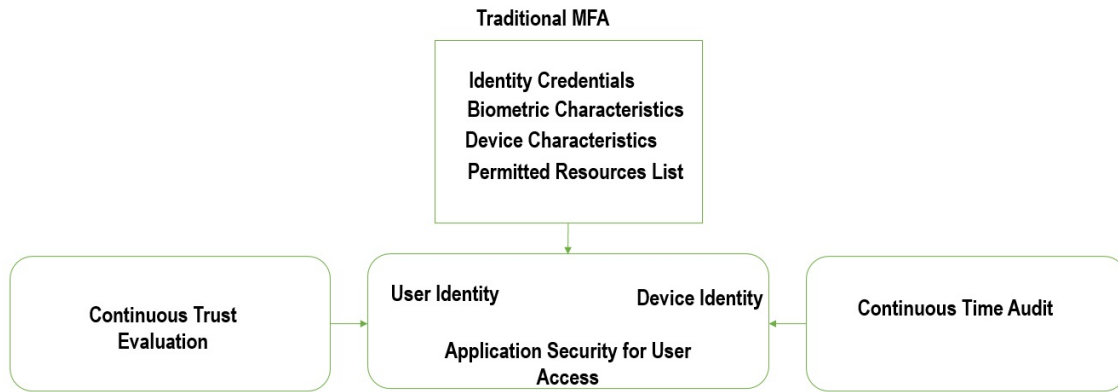


Figure 7.3: Design of MFA over ZTN

#### 7.4.1.2 MFA for User & Device in ZTN

To begin, the user and terminal identity are recorded. They are interconnected with user credentials together and establish a management system for application services based on security levels. Categorization applications into different security levels and customizing user application services based on trust evaluation values is made. Lastly, security checks for the management of users' terminals and their applications is conducted. The schematic design of the proposed approach is depicted in Figure 7.3. Next, traditional MFA is performed to verify the user's identity using existing technologies such as passwords and facial recognition. Traditional MFA requires selecting two or more authentication methods, but it only provides identity input, and it does not support dynamic authentication for security purposes. Therefore, after successful MFA authentication, dynamic authentication based on trust evaluation values is necessary. Once the user successfully completes MFA, dynamic authentication based on trust evaluation values is initiated. This serves two purposes: firstly, to dynamically authenticate the user's access for security and legality purposes, and secondly, to authenticate the user to the server. A continuous trust evaluation module calculates the trust evaluation value of both the user and their terminal. When the trust evaluation value and other attributes match the access policy, the authenticated user is granted the corresponding access authority. The authentication mechanism is depicted in Figure 7.4.



**Figure 7.4:** Design of Dynamic MFA over ZTN

This answers the Research Question 1.

### 7.4.1.3 Device to Device Lightweight Authentication

The proposed approach is covered in three phases namely (i) Initialization, (ii) Traditional MFA, and (iii) Augmenting Device to Device Lightweight Authentication, as discussed:

1. *Initialization:* During the Initialization phase, crucial parameters need to be configured for both the sensor nodes and gateways. Initially, the sensor node provides its identification information, which is known as  $ID_{Ser}$ . There are various methods to transmit the necessary confidential values to both the sensors and gateways. For instance, the sensor manufacturers can pre-embed secret values such as MAC address or globally unique identifier ( $GU_{ID}$ ) into the sensors during the production phase, and the gateway can obtain the necessary confidential values.
2. *Traditional MFA:* In the authentication phase, there is a mutual authentication process between the sensor node and the gateway through static authentication. Both parties also agree upon an initial token called  $TOK_{I_{Ser}}$ , which will be utilized for continuous authentication throughout the authentication period  $T$ .
3. *Augmenting Device to Device Lightweight Authentication:* This phase is designed to implement Lightweight Device to Device Authentication protocol discussed in Chapter 6 for a time duration  $T$  in a continuous manner. In this process, the *Ack*

generated between the mutual authentication between sensor and node is required to maintain a timestamp for audit and enforcement of access control policies.

---

**Algorithm 7** Software Defined Perimeter (SDP)

---

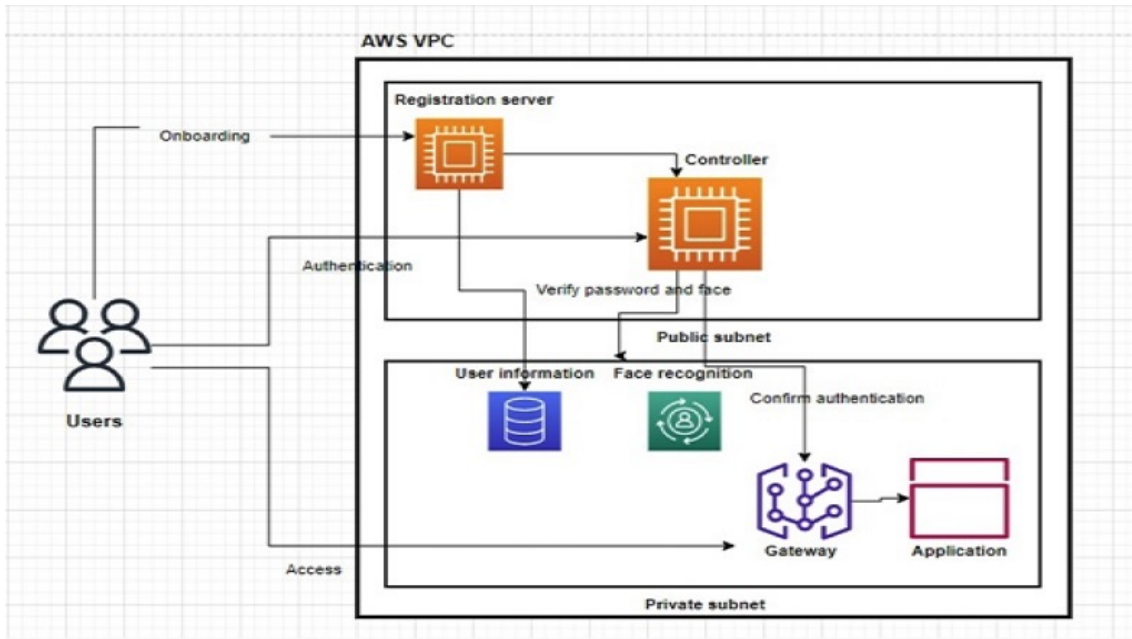
- 1: Function Authenticate (User, Device, Application)
- 2: If User AND Device AND Application Identified and Authenticated:
- 3: Return True. Else Return False.
- 4: EndIf
- 5: Function Authorize (User, Device, Application, Resource)
- 6: If User AND device AND application authorized for resources:
- 7: Return True. Else Return False.
- 8: EndIf
- 9: Function Monitor (network)
- 10: For each user, device and application, Record Activity and Behavior:
- 11: EndFor
- 12: For each resource, Record access attempts and events:
- 13: Function Detect-Threat (network)
- 14: EndFor
- 15: For each recorded event, analyze for anomalies and potential threats.
- 16: Trigger alert or response as desired by the organization.
- 17: Function Segment (network)
- 18: Separate Network to smaller segments based on resource sensitivity.
- 19: Limit access between segments based on least privileges
- 20: Function Encryption (Resources)
- 21: Encrypt data in rest and in transit
- 22: Function Optimize (Policy)
- 23: continuously access and update security controls, policies and procedures based on new threats and changing organizational requirements.
- 24: EndFor

---

### 7.4.1.4 Integration for A-MFA

Considering the advanced facial recognition approach for unique identification for users and MAC address for that of network-enabled devices, the MFA module is designed. Further, the MFA process is designed to be riding over a ZTN with cloud segmentation in a multi-cloud environment. A block diagram for integration of the components for Augmented MFA (A-MFA) is depicted in Figure 7.5.

**This answers the Research Question 2.**



**Figure 7.5:** Integration for A-MFA

### 7.4.2 Design and Development of SDP Architecture

The basic modules of SDP architecture are (i) SDP Controller, (ii) SDP Gateway or Accepting Host, and (iii) SDP client. In the SDP architecture, Users, Devices, and Applications are incorporated. For SDP design, the pseudocode is presented in Algorithm 7.

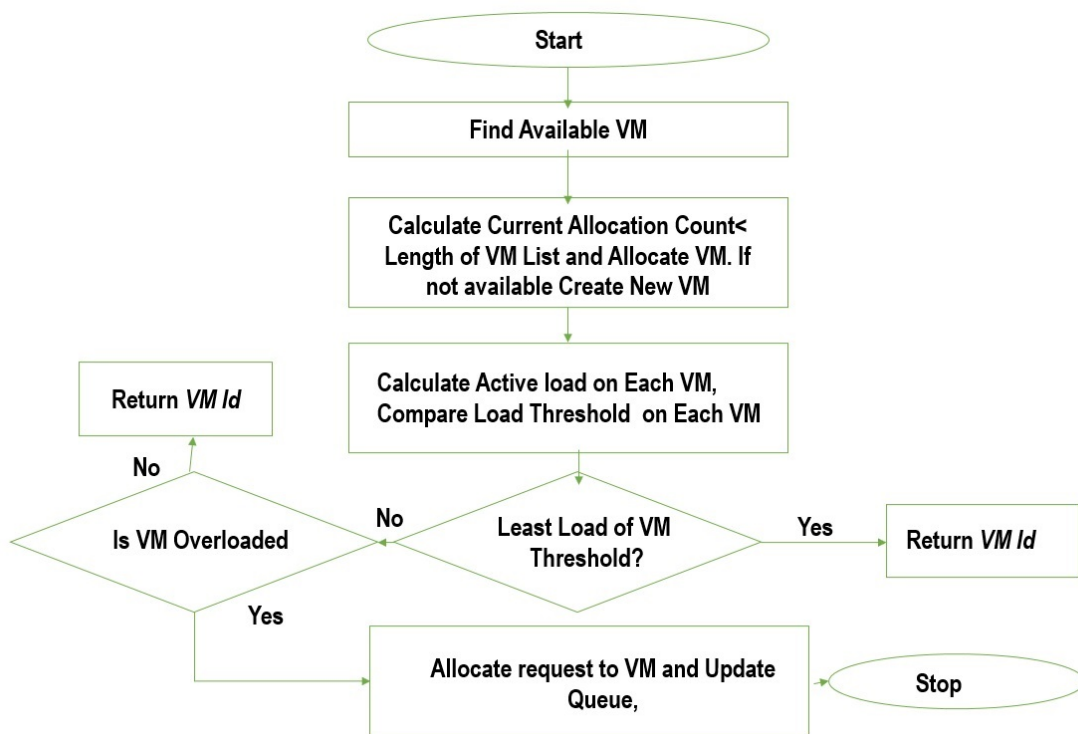
#### 7.4.2.1 Vulnerability for SDP Controller

The SDP controller performs a crucial role in the process of network segmentation. It dynamically updates the access control table. It is also responsible for ensuring *most restrictive access control policy in a ZTN Network*. At the same time, for its operation, the ZTN network mostly depends on a centralized SDP controller. Thus it is vulnerable to a Single Point of Failure (SPF), due to a Denial of Service (DoS) attack, in a common scenario. Hence this research proposes to have multiple SDN controllers for ZTN-enabled SDP setup. Such redundant SDP controllers are designed to be residing on the Cloud server for the inherent advantages of security provided by the Cloud Service Provider (CSP). In the case of a Multi-cloud scenario, such redundant SDP controllers be planned

to span across multiple CSPs, depending on the number of devices and users forming the tenant's overall load on the cloud. However, to streamline the whole process without much complexity. This research proposed to introduce a load balancer [183,183] for the multiple SDP controllers hosted on Multi-cloud.

### 7.4.2.2 SDP Load Balancer

The Load balancer algorithm is based on a Load Threshold calculation. The Virtual Machines (VMs) from the CSPs designated to be performing as a load balancer are combined to form a pool of VMs. The Load Balancer allocates the SDP controller load depending on the Load Threshold of the respective VM. The Flowchart for the operation of the Load balancer for the SDP controller on cloud setup is depicted in Figure 7.6.



**Figure 7.6:** Load Balancer for SDP Controller

This answers the Research Question 3.

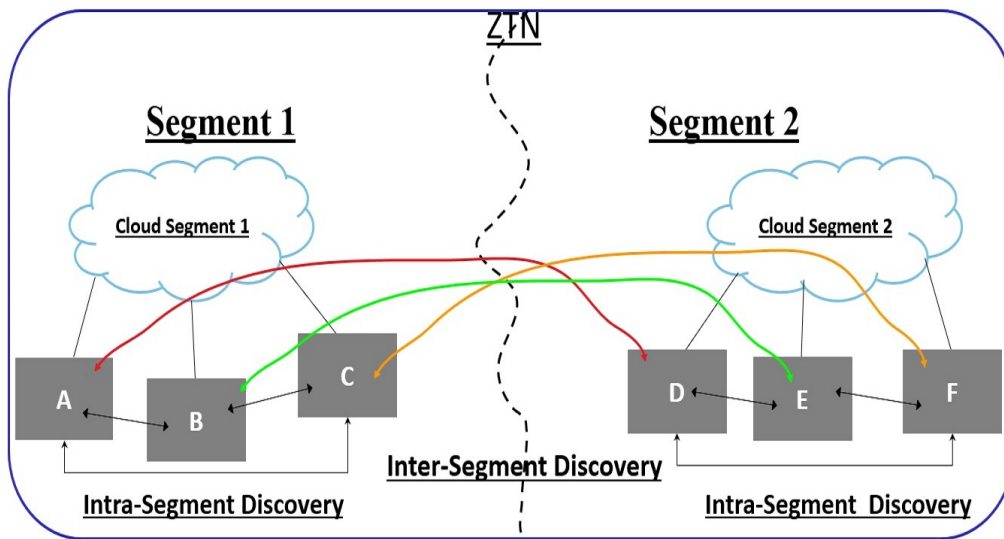


Figure 7.7: Testbed for A-MFA

## 7.5 Implementation and Evaluation

The ZTN is implemented with 16 virtual machines emulated as IoT devices with the webcam as the sensor for IoT applications.

### 7.5.1 Implementation

The implementation has been done in a phased manner. In phase I, the On-premise cloud has been segmented into two parts, configured with one CSP and within 2 Tenants each. Each tenant has been configured with eight users each. The same setup was augmented with a multi-cloud setup, having a database deployed in AWS virtual machine. The system deployment diagram for the testbed is depicted in Figure 7.7.

#### 7.5.1.1 Security Analysis

A comparative analysis of the proposed A-MFA protocol was done after its implementation with other similar functional protocols as depicted in Table 7.2

**Table 7.2:** Comparison of A-MFA With Similar Functioning Identity Authentication Methods

Identity Authentication Method	Continuous Trust Evaluation	Finegrained Access Control	Multi-level Authentication
A-MFA	Yes	Yes	Yes
Biometric Authentication	No	No	No
Password-based Authentication	No	No	No
PKI-based Identity Authentication	No	No	No
Smartcard-based Authentication	No	No	No

### 7.5.2 Performance Evaluation

Using the testbed mentioned at Figure 7.7, in phase I of implementation, 16 users were created on the In-Premise Cloud, which was segmented into two parts, namely CCloud Segment 1 and 2, respectively configured on ZTN architecture. In this phase, the number of users was made to log in using the C-MFA method discussed in Chapter4. Hence in this A-MFA implementation of the user, the associated VNs of the In-premise cloud were also authenticated, with MAC addressed as a factor of authentication for the VMs.

In Phase II, the A-MFA approach was applied to the test setup depicted at Figure 7.5 in a multi-cloud setup comprising AWS as well as the in-premise cloud. The testbed was subjected to varied test conditions where the number of users, as well as devices, were scaled up gradually to study the system performance.

Various gradually enhanced test conditions were subjected to in-premise as well as multi-cloud for several hours to study the processor utilization pattern. A comparison of processor utilization patterns between previously discussed C-MFA was made with the processor utilization pattern of A-MFA under similar test conditions as depicted in Figure 7.8. The A-MFA was found to be processor-heavy as compared to C-MFA. However, this extra processor utilization merits justifiability, considering the enhanced security and technical amalgamation of device and user authentication under one umbrella.

The Load balancer functionality was thoroughly checked under various conditions as above. Using the camera as a sensor of the connected device, a 4 MB video filer was uploaded as well as downloaded under varied scaled-up test conditions to check for system con-



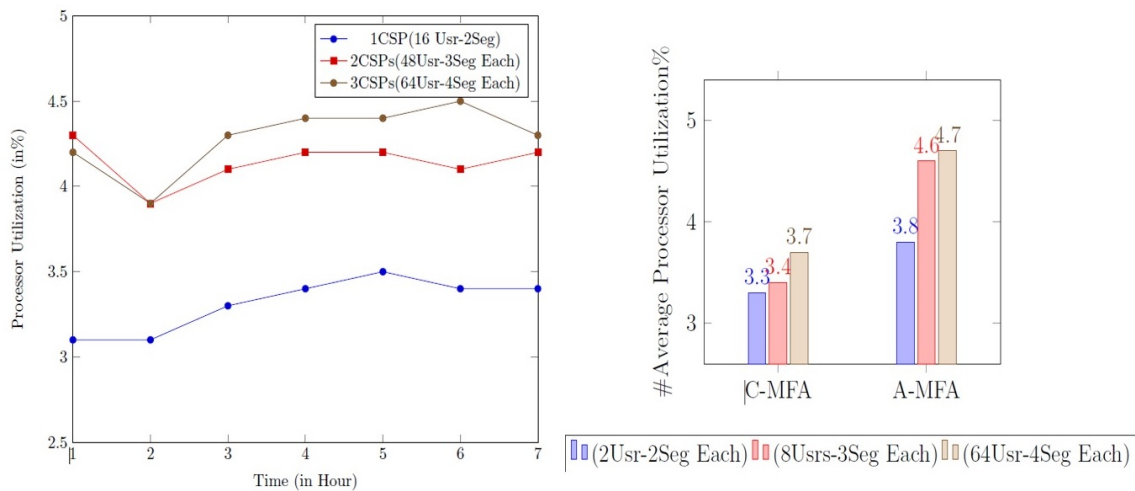


Figure 7.8: Processor Utilization for A-MFA & Comparison to C-MFA

sistency as depicted in Figure 7.9. It was observed that the system performance was consistent throughout the execution of various test conditions. This signified that apart from maintaining Confidentiality, Integrity, and availability, the A-MFA system is found to be consistent in terms of processor load, using various load balancers in an uninterrupted manner for seven hours.

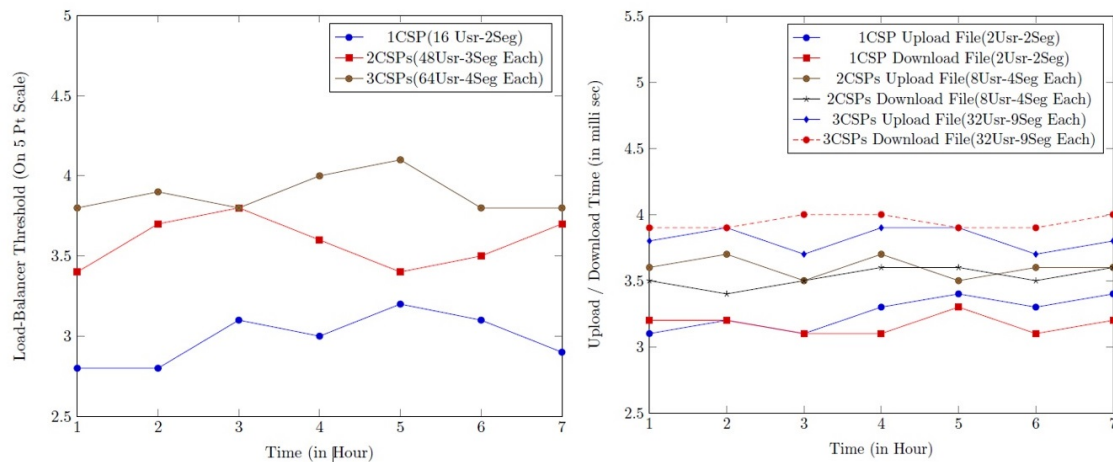


Figure 7.9: Performance of Load Balancer with Secured File Transfer

## 7.6 Chapter Summary

The concept of a Zero-Trust Architecture (ZTA) is becoming more popular and is now seen as the preferred security architecture. ZTA, as the name suggests, is based on the

principles of granting minimum privileges, fine-grained access control, and implementing strict and adaptable policies. ZTN approach for user or device identity has been found promising, with inherent advantages provided by the network features. This research has endeavored to refrain from using any additional or specialized hardware or software using the MFA process. Effective access control of users and devices has been achieved regarding face recognition applications and MAC-based device MFA. The technical amalgamation of user and device MFA in a ZTN platform with an SDP testbed of multi-cloud is a state-of-the-art approach for the future of MFA. The A-MFA approach for device and user authentication has promising results regarding processor utilization and secured data transfer on varied test conditions. Using lightweight authentication and load balancers has proved effective in their implementation. Thus, the A-MFA approach for authentication has been established to have a sound balance between the security of operation and system performance.

## 8 Conclusions

User credentials, including registered users' facial biometric data, are stored in the cloud. Hence, data storage security against any intrusion attempt is of paramount importance. Necessary data backup or redundancy is required to handle data loss. This research work assumes that the security of cloud storage is ensured. Modern spoofing techniques may find a loophole in the system's functioning even though the system is designed and verified against impersonation attacks.- Other attacks, such as spoofing attack prevention, are limited to the deep learning and CNN functionality of the particular face detection and recognition algorithm being utilized for facial recognition in the system. A deliberate attempt to alter crucial credentials and biometric data may render the system completely ineffective. Moreover, one cannot ignore the cloud administrators vital role in resource provisioning.

Our work aims to develop an efficient and reliable approach for secured authentication of users and devices in Multi-cloud and Multi-cloud of Things. In this chapter, we summarize our significant contributions and provide some insights for future work.

### 8.1 Contributions

In this research, we have covered the aspects of MFA in detail. In the process, we have implemented *ABCD of MFA*, namely *A-MFA (Augmented MFA)*, *B-MFA (Blockchain-Powered MFA)*, *C-MFA (Continuous MFA)* and *D-MFA (Dynamic MFA)* and studied their performance characteristics and applicability towards secured user authentication for stand-alone, Cloud as well as Multi-Cloud environments.

### 8.1.1 Findings With Respect to Users

MFA is an effective means of increasing the difficulty for intruders to gain unauthorized access to the system. MFA ensures secured access to resources for interactions between users and cloud infrastructure by facilitating efficient, user-friendly, and trustworthy authentication whenever accessing a service. This research presented an approach to dynamically verify a factor, particularly the biometric face recognition, without disturbing a logged-in user from ongoing work simultaneously without requiring additional hardware and software for the system implementation. Performance consistency of this approach has been experimentally established for various associated phases of MFA system functioning with an web enabled interface. ZAP test of the web-application for the MFA system have established its security as well as stability features. At an average of 350 milliseconds is taken by the MFA system for new user sign up including initial face registration average time of 80 milliseconds. Using an experimental setup where 100 number of users are created using a cloud environment, average time response of 900 milliseconds have been achieved for user sign in including facial verification of the MFA process for registered users. For uploading as well as downloading a 4 MB pdf document has shown promising results using a standard 2 MBPS Broadband internet connection with respect to their time of response in cloud environment. This research also made a notable effort to prevent impersonation attacks on the system. However, the trade-off between performance and security of user authentication has been found effective against impersonation attacks.

### 8.1.2 Findings With Respect to Devices

MFA has been proven effective in making it harder for unauthorized devices to access the system through device swapping. The use of DLT and DID has achieved the desired goals for device authentication and ease of access, while the implementation of hashing and PKI cryptographic functions has ensured system security against fraudulent devices. It is assumed that new device registration to the cloud is done manually or with the help of a cloud administrator. The initial registration of devices is based on the MAC address. In a research testbed, 16 virtual machines were created to emulate 16 devices with limited memory and processing power, using webcams as sensors for IoT devices connected over a multi-cloud. The study also demonstrated promising results for uploading a 4 MB MP4

video file using a standard 2 Mbps broadband internet connection from webcam sensors in a cloud environment. Additionally, the study focused on preventing impersonation attacks on the system using swapped devices with unknown MAC addresses. The use of SHA and PKI for DLT-based credential storage and liveness checks were efficient and effective in enhancing security. However, there is a trade-off between device authentication performance and security, and this trade-off was considered effective against impersonation attacks.

### 8.2 Lessons Learnt

This research focused on the objectives and scope of work within the declared restrictions of not using any additional or specialized hardware and software. Hence, the OEM-fitted webcam and the MAC address of the network interface were utilized as identifying agents for the users and IoT devices, respectively. To cater to the possible spoofing and impersonation attacks, desired threat modelling was done to detect and prevent unauthorized access to enhance security. To have optimum system performance, dynamic verification of users or devices was resorted to instead of continuously engaging computational capability for checking and verifying the authentication factor. However, an alternative means of Continuous MFA was proposed and implemented with CNN based face recognition system that used a pipeline concept starting from frame difference measurement till actual face recognition in a modular manner. The modules of the pipeline were designed to function as and when the need of a specific module was required. Similarly, the usage of a Digital Certificate and Hashing of the MAC address were used as anti-spoofing measures for IoT devices. Moreover, the inherent cryptographic advantages of Blockchain and Distributed Ledgers were also explored for system implementation. Such MFA scheme was tested for validation of functional performance in a phased manner, firstly on an in-premise cloud set-up. Subsequently, the same was ported to commercially available CSPs with a scaled-up number of tenants and users. Virtual machines with limited system configuration were prepared in the laboratory to simulate the IoT devices.

An augmented means for MFA was explored with the help of ZTN and SDP-based architecture for users and devices. However, its load balancer was found to be an unavoidable

computational overhead. Hence, its effect on secure data transfer was studied. For internet connectivity and to incorporate the virtual machines of test beds from the commercially available CSPs, an OFC-based 2 Mbps internet connection was used. Moreover, the testing and validation of the proposed system functionality and performance was done in an academic laboratory environment with an imaginary set of data in terms of user ID, a limited number of faces for testing associated with multiple user IDs and MAC addresses of virtual machines prepared to pose as IoT devices. For more realistic testing with real-time environment variables, the need for real-time dataset utilization is necessary.

### 8.3 Future Work

In this section, we provide a brief insight to the following possible extensions to our work.

1. Artificial Intelligence (AI) is making its presence felt in almost all spheres of life. Passwordless authentication using AI is an emerging approach that uses various AI techniques to authenticate users without requiring traditional passwords or other forms of authentication. Passwordless authentication can offer a higher level of security and convenience to users, as it eliminates the need for users to remember complex passwords or rely on other authentication factors such as tokens or biometrics.

We aim to explore passwordless authentication using AI due to its potential to offer a more secure and convenient authentication experience for users, while reducing the risk of password-related security breaches. However, it is important to ensure that the AI algorithms used for authentication are robust and secure, and that appropriate safeguards are in place to protect user privacy and prevent misuse of the technology.

2. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks can pose a significant threat to authentication systems, as they can overwhelm the system with traffic and disrupt the authentication process. To detect and prevent DoS and DDoS attacks on authentication systems, various techniques can be used.

We intend to conduct further research for detecting and preventing DoS and DDoS attacks on authentication systems. This would be requires a combination of tech-

niques that includes traffic analysis, rate limiting, IP blocking, load balancing, and cloud-based protection solutions. It is also important to regularly monitor the system for any signs of suspicious activity and update security measures as needed to stay ahead of evolving threats.

3. "Authentication as-you-go" model or "authentication on-the-fly" model, refers to the process of continuously verifying the identity of a user as they interact with a system, instead of relying on a single authentication event at the beginning of a session. It can also provide a more seamless user experience, as users are not required to repeatedly enter credentials or re-authenticate during a session.

This approach can provide a higher level of security and a more seamless user experience by continuously monitoring user behavior and detecting any signs of suspicious activity. As it is important to ensure that the techniques used for continuous authentication are reliable, accurate, and protect user privacy, we aim to further conduct research for enhancement of authentication reliability and accuracy.

4. The Internet of Things (IoT) and the concept of the Connected World have brought about the idea of a Connected Technology (CoT) ecosystem where multiple devices can connect and communicate with each other. Tamper-proof device authentication in CoT is critical to ensure secure communication and prevent unauthorized access. A combination of secure boot, hardware-based authentication, mutual authentication, certificate-based authentication, and physical security measures can be used to achieve tamper-proof device authentication in CoT.

We intend to explore details of these and conduct research to ensure that these techniques are reliable, secure, and protect the devices from being cloned.

5. A Physical Unclonable Function (PUF) is a technique used to generate unique and unpredictable cryptographic keys from a physical devices. PUFs can be used to prevent a bot from taking over a device in a cloud environment by providing a unique identifier that can be used for device authentication. PUFs are difficult to clone or replicate, making them an effective way to prevent unauthorized access to the cloud.

A DoS or DDoS attack on PUF key generation and decimation can make this means

as ineffective and an attacker bot can take over a genuine device of a CoT. We intend to conduct research to detect and prevent DoS and DDoS attacks on PUF in CoT.



## Bibliography

- [1] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, M. Badger, and D. Leaf, "Nist cloud computing reference architecture," 2011-09-08 00:09:00 2011. [Online]. Available: [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=909505](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=909505)
- [2] S. Iqbal, M. L. M. Kiah, N. B. Anuar, B. Daghighi, A. W. A. Wahab, and S. Khan, "Service delivery models of cloud computing: security issues and open challenges," *Security and Communication Networks*, vol. 9, no. 17, pp. 4726–4750, aug 2016. [Online]. Available: <https://doi.org/10.1002%2Fsec.1585>
- [3] F. K. Parast, C. Sindhav, S. Nikam, H. I. Yekta, K. B. Kent, and S. Hakak, "Cloud computing security: A survey of service-based models," *Computers & Security*, vol. 114, p. 102580, mar 2022. [Online]. Available: <https://doi.org/10.1016%2Fj.cose.2021.102580>
- [4] M. Jangjou and M. K. Sohrabi, "A comprehensive survey on security challenges in different network layers in cloud computing," *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 3587–3608, jan 2022. [Online]. Available: <https://doi.org/10.1007%2Fs11831-022-09708-9>
- [5] P. Sun, "Security and privacy protection in cloud computing: Discussions and challenges," *Journal of Network and Computer Applications*, vol. 160, p. 102642, jun 2020. [Online]. Available: <https://doi.org/10.1016%2Fj.jnca.2020.102642>
- [6] O. Tomarchio, D. Calcaterra, and G. D. Modica, "Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks," *Journal of Cloud Computing*, vol. 9, no. 1, sep 2020. [Online]. Available: <https://doi.org/10.1186%2Fs13677-020-00194-7>
- [7] "Annual data breach report," Jan 2023. [Online]. Available: <https://www.idtheftcenter.org/publication/2022-data-breach-report/>
- [8] V. C. Hu, "General access control guidance for cloud systems," apr 2020. [Online]. Available: <https://doi.org/10.6028%2Fnist.sp.800-210-draft>
- [9] M. Fagan, "IoT device cybersecurity guidance for the federal government:," oct 2020. [Online]. Available: <https://doi.org/10.6028%2Fnist.sp.800-213-draft>
- [10] M. Adam, M. Wessel, and A. Benlian, "AI-based chatbots in customer service and their effects on user compliance," *Electronic Markets*, vol. 31, no. 2, pp. 427–445, mar 2020. [Online]. Available: <https://doi.org/10.1007%2Fs12525-020-00414-7>

- 
- [11] Y. Lu and D. Zhao, "Providing impersonation resistance for biometric-based authentication scheme in mobile cloud computing service," *Computer Communications*, vol. 182, pp. 22–30, jan 2022. [Online]. Available: <https://doi.org/10.1016%2Fj.comcom.2021.10.029>
- [12] N. Kumar and S. C. Lee, "Human-machine interface in smart factory: A systematic literature review," *Technological Forecasting and Social Change*, vol. 174, p. 121284, jan 2022. [Online]. Available: <https://doi.org/10.1016%2Fj.techfore.2021.121284>
- [13] P. Osterrieder, L. Budde, and T. Friedli, "The smart factory as a key construct of industry 4.0: A systematic literature review," *International Journal of Production Economics*, vol. 221, p. 107476, mar 2020. [Online]. Available: <https://doi.org/10.1016%2Fj.ijpe.2019.08.011>
- [14] Z. A. Zukarnain, A. Muneer, and M. K. A. Aziz, "Authentication securing methods for mobile identity: Issues, solutions and challenges," *Symmetry*, vol. 14, no. 4, p. 821, apr 2022. [Online]. Available: <https://doi.org/10.3390%2Fsym14040821>
- [15] S. Tufail, I. Parvez, S. Batool, and A. Sarwat, "A survey on cybersecurity challenges, detection, and mitigation techniques for the smart grid," *Energies*, vol. 14, no. 18, p. 5894, sep 2021. [Online]. Available: <https://doi.org/10.3390%2Fen14185894>
- [16] A. H. Y. Mohammed, R. A. Dziauddin, and L. A. Latiff, "Current multi-factor of authentication: Approaches, requirements, attacks and challenges," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023. [Online]. Available: <https://doi.org/10.14569%2Fijacsa.2023.0140119>
- [17] K. S. S. Latha, H. V. Githiki, M. L. S. Morla, A. S. Vanama, and Y. Tripathi, "A systematic literature review on security in cloud technology," in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE, mar 2023. [Online]. Available: <https://doi.org/10.1109%2Ficears56392.2023.10085641>
- [18] A. Morton, "Managing security and access control," in *Mastering Snowflake Solutions*. Apress, 2022, pp. 67–98. [Online]. Available: [https://doi.org/10.1007%2F978-1-4842-8029-4\\_4](https://doi.org/10.1007%2F978-1-4842-8029-4_4)
- [19] E. P. Kechagias, G. Chatzistelios, G. A. Papadopoulos, and P. Apostolou, "Digital transformation of the maritime industry: A cybersecurity systemic approach," *International Journal of Critical Infrastructure Protection*, vol. 37, p. 100526, jul 2022. [Online]. Available: <https://doi.org/10.1016%2Fj.ijcip.2022.100526>
- [20] A. Manzoor, M. A. Shah, H. A. Khattak, I. U. Din, and M. K. Khan, "Multi-tier authentication schemes for fog computing: Architecture, security perspective, and challenges," *International Journal of Communication Systems*, vol. 35, no. 12, jun 2019. [Online]. Available: <https://doi.org/10.1002%2Fdac.4033>
- [21] R. AlHusain and A. Alkhalifah, "Evaluating fallback authentication research: A

- systematic literature review,” *Computers & Security*, vol. 111, p. 102487, dec 2021. [Online]. Available: <https://doi.org/10.1016%2Fj.cose.2021.102487>
- [22] K. Kritikos, K. Magoutis, M. Papoutsakis, and S. Ioannidis, “A survey on vulnerability assessment tools and databases for cloud-based web applications,” *Array*, vol. 3-4, p. 100011, sep 2019. [Online]. Available: <https://doi.org/10.1016%2Fj.array.2019.100011>
- [23] H. I. Kure, S. Islam, M. Ghazanfar, A. Raza, and M. Pasha, “Asset criticality and risk prediction for an effective cybersecurity risk management of cyber-physical system,” *Neural Computing and Applications*, vol. 34, no. 1, pp. 493–514, aug 2021. [Online]. Available: <https://doi.org/10.1007%2Fs00521-021-06400-0>
- [24] J. Surbiryala and C. Rong, “Cloud computing: History and overview,” in *2019 IEEE Cloud Summit*. IEEE, aug 2019. [Online]. Available: <https://doi.org/10.1109%2Fcloudsummit47114.2019.00007>
- [25] K. Hamid, M. W. Iqbal, Q. Abbas, M. Arif, A. Brezilianu, and O. Geman, “Cloud computing network empowered by modern topological invariants,” *Applied Sciences*, vol. 13, no. 3, p. 1399, jan 2023. [Online]. Available: <https://doi.org/10.3390%2Fapp13031399>
- [26] S. W. Shah and S. S. Kanhere, “Recent trends in user authentication – a survey,” *IEEE Access*, vol. 7, pp. 112 505–112 519, 2019. [Online]. Available: <https://doi.org/10.1109%2Faccess.2019.2932400>
- [27] A. R. Khan and L. K. Alnwihel, “A brief review on cloud computing authentication frameworks,” *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 9997–10 004, feb 2023. [Online]. Available: <https://doi.org/10.48084%2Fetasr.5479>
- [28] P. Voegel and A. Ouda, “An innovative multi-factor authentication approach,” in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, jul 2022. [Online]. Available: <https://doi.org/10.1109%2Fisncc55209.2022.9851710>
- [29] L. Megouache, A. Zitouni, and M. Djoudi, “Ensuring user authentication and data integrity in multi-cloud environment,” *Human-centric Computing and Information Sciences*, vol. 10, no. 1, apr 2020. [Online]. Available: <https://doi.org/10.1186%2Fs13673-020-00224-y>
- [30] S. Prithi, D. Sumathi, T. Poongodi, and P. Suresh, “Trust management framework for handling security issues in multi-cloud environment,” in *Operationalizing Multi-Cloud Environments*. Springer International Publishing, sep 2021, pp. 287–306. [Online]. Available: [https://doi.org/10.1007%2F978-3-030-74402-1\\_16](https://doi.org/10.1007%2F978-3-030-74402-1_16)
- [31] A. H. Shaikh and B. B. Meshram, “Security issues in cloud computing,” in *Intelligent Computing and Networking*. Springer Singapore, oct 2020, pp. 63–77. [Online]. Available: [https://doi.org/10.1007%2F978-981-15-7421-4\\_6](https://doi.org/10.1007%2F978-981-15-7421-4_6)

- [32] M. Humayun, M. Niazi, M. F. Almufareh, N. Z. Jhanjhi, S. Mahmood, and M. Alshayeb, "Software-as-a-service security challenges and best practices: A multivocal literature review," *Applied Sciences*, vol. 12, no. 8, p. 3953, apr 2022. [Online]. Available: <https://doi.org/10.3390%2Fapp12083953>
- [33] S. T. Zargar, H. Takabi, and J. Iyer, "Security-as-a-service (SECaaS) in the cloud," in *Security, Privacy, and Digital Forensics in the Cloud*. John Wiley & Sons Singapore Pte. Ltd, feb 2019, pp. 189–200. [Online]. Available: <https://doi.org/10.1002%2F9781119053385.ch9>
- [34] C. Alliance, "The treacherous twelve-cloud computing top threats in 2016," 2016.
- [35] "Security guidance for critical areas of focus in cloud computing v4.0."
- [36] I. Ali, S. Sabir, and Z. Ullah, "Internet of things security, device authentication and access control: a review," *arXiv preprint arXiv:1901.07309*, Apr 2022. [Online]. Available: <https://arxiv.org/abs/1901.07309>
- [37] L. Teng, M. Jianfeng, F. Pengbin, M. Yue, M. Xindi, Z. Jiawei, C. Gao, and L. Di, "Lightweight security authentication mechanism towards UAV networks," in *2019 International Conference on Networking and Network Applications (NaNA)*. IEEE, oct 2019. [Online]. Available: <https://doi.org/10.1109%2Fnana.2019.00072>
- [38] P. A. Grassi, M. E. Garcia, and J. L. Fenton, "Digital identity guidelines: revision 3," Tech. Rep., jun 2017. [Online]. Available: <https://doi.org/10.6028%2Fnist.sp.800-63-3>
- [39] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Computer Science Review*, vol. 33, pp. 1–48, aug 2019. [Online]. Available: <https://doi.org/10.1016%2Fj.cosrev.2019.05.002>
- [40] R. A. Nafea and M. A. Almaiah, "Cyber security threats in cloud: Literature review," in *2021 International Conference on Information Technology (ICIT)*. IEEE, jul 2021. [Online]. Available: <https://doi.org/10.1109%2Ficit52682.2021.9491638>
- [41] C. Tankard, "Credential stuffing – the new hack," *Network Security*, vol. 2021, no. 2, pp. 20–20, feb 2021. [Online]. Available: <https://doi.org/10.1016%2Fs1353-4858%2821%2900021-0>
- [42] Y. AlHumaidan, L. AlAjmi, M. Aljamea, and M. Mahmud, "ANALYSIS OF CLOUD COMPUTING SECURITY IN PERSPECTIVE OF SAUDI ARABIA," in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, sep 2018. [Online]. Available: <https://doi.org/10.1109%2Fhealthcom.2018.8531141>
- [43] X. Sun, "Critical security issues in cloud computing: A survey," in *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and*

- IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, may 2018. [Online]. Available: <https://doi.org/10.1109%2Fbds%2Fhpsc%2Fids18.2018.00053>
- [44] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9762–9773, dec 2019. [Online]. Available: <https://doi.org/10.1109%2Fjiot.2019.2931372>
- [45] G. S. Kumar, N. Kandavel, and K. Madhavan, "To discovery the cloud services authentication an expert based system using multi-factor authentication," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, mar 2020. [Online]. Available: <https://doi.org/10.1109%2Ficaccs48705.2020.9074195>
- [46] V. Singh and S. K. Pandey, "Revisiting cloud security threats: Replay attack," in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, dec 2018. [Online]. Available: <https://doi.org/10.1109%2Fccaa.2018.8777341>
- [47] Z. Ye, Y. Guo, A. Ju, F. Wei, R. Zhang, and J. Ma, "A risk analysis framework for social engineering attack based on user profiling," *Journal of Organizational and End User Computing*, vol. 32, no. 3, pp. 37–49, jul 2020. [Online]. Available: <https://doi.org/10.4018%2Fjoeuc.2020070104>
- [48] M. Sain, O. Normurodov, C. Hong, and K. L. Hui, "A survey on the security in cyber physical system with multi-factor authentication," in *2021 23rd International Conference on Advanced Communication Technology (ICACT)*. IEEE, feb 2021. [Online]. Available: <https://doi.org/10.23919%2Ficact51234.2021.9370515>
- [49] B. O. ALSaleem and A. I. Alshoshan, "Multi-factor authentication to systems login," in *2021 National Computing Colleges Conference (NCCC)*. IEEE, mar 2021. [Online]. Available: <https://doi.org/10.1109%2Fnccc49330.2021.9428806>
- [50] S. Subangan and V. Senthooan, "Secure authentication mechanism for resistance to password attacks," in *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*. IEEE, sep 2019. [Online]. Available: <https://doi.org/10.1109%2Ficter48817.2019.9023773>
- [51] H. A. Kholidy, "Detecting impersonation attacks in cloud computing environments using a centric user profiling approach," *Future Generation Computer Systems*, vol. 117, pp. 299–320, apr 2021. [Online]. Available: <https://doi.org/10.1016%2Fj.future.2020.12.009>
- [52] S. Sambangi, L. Gondi, and S. Aljawarneh, "A feature similarity machine learning model for DDoS attack detection in modern network environments for industry 4.0," *Computers and Electrical Engineering*, vol. 100, p. 107955, may 2022. [Online]. Available: <https://doi.org/10.1016%2Fj.compeleceng.2022.107955>

- 
- [53] S. Boonkrong, "Multi-factor authentication," in *Authentication and Access Control*. Apress, dec 2020, pp. 133–162. [Online]. Available: [https://doi.org/10.1007/978-1-4842-6570-3\\_6](https://doi.org/10.1007/978-1-4842-6570-3_6)
- [54] N. M. Kumar and P. K. Mallick, "Blockchain technology for security issues and challenges in IoT," *Procedia Computer Science*, vol. 132, pp. 1815–1823, 2018. [Online]. Available: <https://doi.org/10.1016/j.procs.2018.05.140>
- [55] H. S. K. Sheth, I. A. K, and A. K. Tyagi, "Deep learning, blockchain based multi-layered authentication and security architectures," in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. IEEE, may 2022. [Online]. Available: <https://doi.org/10.1109/ICAaic53929.2022.9793179>
- [56] P. Oza and V. M. Patel, "Active authentication using an autoencoder regularized CNN-based one-class classifier," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (2019)*. IEEE, may 2019. [Online]. Available: <https://doi.org/10.1109/Fg.2019.8756525>
- [57] M. Zulfiqar, F. Syed, M. J. Khan, and K. Khurshid, "Deep face recognition for biometric authentication," in *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, jul 2019. [Online]. Available: <https://doi.org/10.1109/Ficecce47252.2019.8940725>
- [58] Z. A. Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services BRAINS*. IEEE, sep 2020. [Online]. Available: <https://doi.org/10.1109/Fbrains49436.2020.9223292>
- [59] R. R. Omar and T. M. Abdelaziz, "A comparative study of network access control and software-defined perimeter," in *Proceedings of the 6th International Conference on Engineering & MIS 2020*. ACM, sep 2020. [Online]. Available: <https://doi.org/10.1145/F3410352.3410754>
- [60] K. Hatakeyama, D. Kotani, and Y. Okabe, "Zero trust federation: Sharing context under user control towards zero trust in identity federation," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, mar 2021. [Online]. Available: <https://doi.org/10.1109/Percomworkshops51409.2021.9431116>
- [61] J. Perhac, M. Ferencsik, V. Zhukovskyy, N. Zhukovska, and S. Shatnyi, "Visualization of syntax and semantics for simple functional language of natural numbers and boolean values," in *2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC)*. IEEE, jul 2022. [Online]. Available: <https://doi.org/10.1109/Ficcc202255925.2022.9922774>
- [62] S. Mujeye, Y. Levy, H. Mattord, and W. Li, "Empirical results of an experimental study on the role of password strength and cognitive load on employee productivity," *Online Journal of Applied Knowledge Management*, vol. 4, no. 1, pp.

- 99–116, may 2016. [Online]. Available: <https://doi.org/10.36965%2Fojakm.2016.4%281%2999-116>
- [63] S. C. Patel, S. Jaiswal, R. S. Singh, and J. Chauhan, “Access control framework using multi-factor authentication in cloud computing,” *International Journal of Green Computing*, vol. 9, no. 2, pp. 1–15, jul 2018. [Online]. Available: <https://doi.org/10.4018%2Fijgc.2018070101>
- [64] E. Erdem and M. T. Sandikkaya, “OTPaas—one time password as a service,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 743–756, mar 2019. [Online]. Available: <https://doi.org/10.1109%2Ftifs.2018.2866025>
- [65] H. Kim, J. Han, C. Park, and O. Yi, “Analysis of vulnerabilities that can occur when generating one-time password,” *Applied Sciences*, vol. 10, no. 8, p. 2961, apr 2020. [Online]. Available: <https://doi.org/10.3390%2Fapp10082961>
- [66] C. Ozkan and K. Bicakci, “Security analysis of mobile authenticator applications,” in *2020 International Conference on Information Security and Cryptology (ISCTURKEY)*. IEEE, dec 2020. [Online]. Available: <https://doi.org/10.1109%2Fiscturkey51113.2020.9308020>
- [67] I. Gordin, A. Graur, and A. Potorac, “Two-factor authentication framework for private cloud,” in *2019 23rd International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, oct 2019. [Online]. Available: <https://doi.org/10.1109%2Ficstcc.2019.8885460>
- [68] M. Bouchaala, C. Ghazel, and L. A. Saidane, “Enhancing security and efficiency in cloud computing authentication and key agreement scheme based on smart card,” *The Journal of Supercomputing*, vol. 78, no. 1, pp. 497–522, may 2021. [Online]. Available: <https://doi.org/10.1007%2Fs11227-021-03857-7>
- [69] S. Bobba and V. Paruchuri, “Single sign-on using contactless smart cards and fingerprint authentication,” in *Lecture Notes in Networks and Systems*. Springer International Publishing, oct 2021, pp. 158–166. [Online]. Available: [https://doi.org/10.1007%2F978-3-030-90072-4\\_16](https://doi.org/10.1007%2F978-3-030-90072-4_16)
- [70] T. Jaikla, S. Pichetjamroen, C. Vorakulpipat, and A. Pichetjamroen, “A secure four-factor attendance system for smartphone device,” in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*. IEEE, feb 2020. [Online]. Available: <https://doi.org/10.23919%2Ficact48636.2020.9061431>
- [71] Y.-T. Chang and M. J. Dupuis, “My voiceprint is my authenticator: A two-layer authentication approach using voiceprint for voice assistants,” in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*. IEEE, aug 2019. [Online]. Available: <https://doi.org/10.1109%2Fsmartworld-uic-atc-scalcom-iop-sci.2019.00243>
- [72] S. Debnath, K. Ramalakshmi, and M. Senbagavalli, “Multimodal authentication

- system based on audio-visual data: A review,” in *2022 International Conference for Advancement in Technology (ICONAT)*. IEEE, jan 2022. [Online]. Available: <https://doi.org/10.1109%2Ficonat53423.2022.9725889>
- [73] S. Liu, Y. Song, M. Zhang, J. Zhao, S. Yang, and K. Hou, “An identity authentication method combining liveness detection and face recognition,” *Sensors*, vol. 19, no. 21, p. 4733, oct 2019. [Online]. Available: <https://doi.org/10.3390%2Fs19214733>
- [74] A. Musa, K. Vishi, and B. Rexha, “Attack analysis of face recognition authentication systems using fast gradient sign method,” *Applied Artificial Intelligence*, vol. 35, no. 15, pp. 1346–1360, sep 2021. [Online]. Available: <https://doi.org/10.1080%2F08839514.2021.1978149>
- [75] A. Almadan and A. Rattani, “Compact CNN models for on-device ocular-based user recognition in mobile devices,” in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, dec 2021. [Online]. Available: <https://doi.org/10.1109%2Fssci50451.2021.9660033>
- [76] M. V. B. Reddy and V. Goutham, “IRIS TECHNOLOGY: A REVIEW ON IRIS BASED BIOMETRIC SYSTEMS FOR UNIQUE HUMAN IDENTIFICATION,” *International Journal of Research -GRANTHAALAYAH*, vol. 6, no. 1, pp. 80–90, jan 2018. [Online]. Available: <https://doi.org/10.29121%2Fgranthaalayah.v6.i1.2018.1596>
- [77] V. Kakkad, M. Patel, and M. Shah, “Biometric authentication and image encryption for image security in cloud framework,” *Multiscale and Multidisciplinary Modeling, Experiments and Design*, vol. 2, no. 4, pp. 233–248, may 2019. [Online]. Available: <https://doi.org/10.1007%2Fs41939-019-00049-y>
- [78] A. A. S. AlQahtani, Z. El-Awadi, and M. Min, “A survey on user authentication factors,” in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, oct 2021. [Online]. Available: <https://doi.org/10.1109%2Fiemcon53756.2021.9623159>
- [79] R. V. Adiraju, K. K. Masanipalli, T. D. Reddy, R. Pedapalli, S. Chundru, and A. K. Panigrahy, “An extensive survey on finger and palm vein recognition system,” *Materials Today: Proceedings*, vol. 45, pp. 1804–1808, 2021. [Online]. Available: <https://doi.org/10.1016%2Fj.matpr.2020.08.742>
- [80] “2020 index IEEE transactions on biometrics, behavior, and identity science vol. 2,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 2, no. 4, pp. 431–437, oct 2020. [Online]. Available: <https://doi.org/10.1109%2Ftbiom.2020.3028623>
- [81] D. Aiordachioaie, A. Culea-Florescu, and S. M. Pavel, “On human faces thermal image processing for classification purposes,” in *2019 6th International Symposium on Electrical and Electronics Engineering (ISEEE)*. IEEE, oct 2019. [Online]. Available: <https://doi.org/10.1109%2Fiseee48094.2019.9136103>



- [82] S. Kakarwal, K. Chaudhari, R. R. Deshmukh, and R. Patil, "Thermal face recognition using artificial neural network," in *2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*. IEEE, oct 2020. [Online]. Available: <https://doi.org/10.1109%2Ficsidempc49020.2020.9299648>
- [83] "2019 IEEE international conference on big data and smart computing (BigComp)," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, feb 2019. [Online]. Available: <https://doi.org/10.1109%2Fbigcomp.2019.8679460>
- [84] A. S. Anakath, S. Rajakumar, and S. Ambika, "Privacy preserving multi factor authentication using trust management," *Cluster Computing*, vol. 22, no. S5, pp. 10 817–10 823, sep 2017. [Online]. Available: <https://doi.org/10.1007%2Fs10586-017-1181-0>
- [85] A. Ometov, V. Petrov, S. Bezzateev, S. Andreev, Y. Koucheryavy, and M. Gerla, "Challenges of multi-factor authentication for securing advanced IoT applications," *IEEE Network*, vol. 33, no. 2, pp. 82–88, mar 2019. [Online]. Available: <https://doi.org/10.1109%2Fmnet.2019.1800240>
- [86] L. Dostálek and J. Šafařík, "MULTI-FACTOR AUTHENTICATION MODELLING," *Radio Electronics, Computer Science, Control*, vol. 0, no. 2, pp. 106–116, sep 2020. [Online]. Available: <https://doi.org/10.15588%2F1607-3274-2020-2-11>
- [87] S. Nozaki, A. Serizawa, M. Yoshihira, M. Fujita, Y. Shibata, T. Yamanaka, N. Matsuda, T. Ohki, and M. Nishigaki, "Multi-observed multi-factor authentication: A multi-factor authentication using single credential," in *Lecture Notes in Networks and Systems*. Springer International Publishing, 2022, pp. 201–211. [Online]. Available: [https://doi.org/10.1007%2F978-3-031-14314-4\\_20](https://doi.org/10.1007%2F978-3-031-14314-4_20)
- [88] P. Mishra, S. M. Busetty, and S. K. Gudla, "Enhanced activity recognition of the IoT smart home users through cluster analysis," in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*. IEEE, jun 2020. [Online]. Available: <https://doi.org/10.1109%2Fwf-iot48130.2020.9221177>
- [89] M. I. Hussain, J. He, N. Zhu, F. Sabah, Z. A. Zardari, S. Hussain, and F. Razque, "AAAA: SSO and MFA implementation in multi-cloud to mitigate rising threats and concerns related to user metadata," *Applied Sciences*, vol. 11, no. 7, p. 3012, mar 2021. [Online]. Available: <https://doi.org/10.3390%2Fapp11073012>
- [90] M. Rusdan, "Designing of user authentication based on multi-factor authentication on wireless networks," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 1, pp. 201–209, feb 2020. [Online]. Available: <https://doi.org/10.5373%2Fjardcs%2Fv12i1%2F20201030>
- [91] H. D. Silva, D. Wittebron, A. Lahiru, K. Madumadhavi, L. Rupasinghe, and K. Y. Abeywardena, "AuthDNA: An adaptive authentication service for any identity server," in *2019 International Conference on Advancements in Computing (ICAC)*.

- IEEE, dec 2019. [Online]. Available: <https://doi.org/10.1109%2Ficac49085.2019.9103382>
- [92] W. N. W. Muhamad, N. A. M. Razali, K. K. Ishak, N. A. Hasbullah, N. M. Zainudin, S. Ramli, M. Wook, Z. Ishak, and N. J. A. MSaad, "Enhance multi-factor authentication model for intelligence community access to critical surveillance data," in *Advances in Visual Informatics*. Springer International Publishing, 2019, pp. 560–569. [Online]. Available: [https://doi.org/10.1007%2F978-3-030-34032-2\\_49](https://doi.org/10.1007%2F978-3-030-34032-2_49)
- [93] K. A. Taher, T. Nahar, and S. A. Hossain, "Enhanced cryptocurrency security by time-based token multi-factor authentication algorithm," in *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*. IEEE, jan 2019. [Online]. Available: <https://doi.org/10.1109%2Ficrest.2019.8644084>
- [94] O. Mir, M. Roland, and R. Mayrhofer, "DAMFA: Decentralized anonymous multi-factor authentication," in *Proceedings of the 2nd ACM International Symposium on Blockchain and Secure Critical Infrastructure*. ACM, oct 2020. [Online]. Available: <https://doi.org/10.1145%2F3384943.3409417>
- [95] M. K. Rao, S. G. Santhi, and M. A. Hussain, "Multi factor user authentication mechanism using internet of things," in *Proceedings of the Third International Conference on Advanced Informatics for Computing Research*. ACM, jun 2019. [Online]. Available: <https://doi.org/10.1145%2F3339311.3339335>
- [96] V. Rao and P. K.V., "Light-weight hashing method for user authentication in internet-of-things," *Ad Hoc Networks*, vol. 89, pp. 97–106, jun 2019. [Online]. Available: <https://doi.org/10.1016%2Fj.adhoc.2019.03.003>
- [97] W. Ali, W. Tian, S. U. Din, D. Iradukunda, and A. A. Khan, "Classical and modern face recognition approaches: a complete review," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 4825–4880, oct 2020. [Online]. Available: <https://doi.org/10.1007%2Fs11042-020-09850-1>
- [98] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, may 2004. [Online]. Available: <https://doi.org/10.1023%2Fb%3Avisi.0000013087.49260.fb>
- [99] W. Zhou, S. Gao, L. Zhang, and X. Lou, "Histogram of oriented gradients feature extraction from raw bayer pattern images," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 946–950, may 2020. [Online]. Available: <https://doi.org/10.1109%2Ftcsii.2020.2980557>
- [100] B. Yang and S. Chen, "A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image," *Neurocomputing*, vol. 120, pp. 365–379, nov 2013. [Online]. Available: <https://doi.org/10.1016%2Fj.neucom.2012.10.032>
- [101] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and

- future of face recognition: A review,” *Electronics*, vol. 9, no. 8, p. 1188, jul 2020. [Online]. Available: <https://doi.org/10.3390%2Felectronics9081188>
- [102] D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, “A deep learning approach for face detection using YOLO,” in *2018 IEEE Punecon*. IEEE, nov 2018. [Online]. Available: <https://doi.org/10.1109%2Fpunecon.2018.8745376>
- [103] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016. [Online]. Available: <https://doi.org/10.1109%2Fcvpr.2016.91>
- [104] “A review on real time object detection using single shot multibox detector,” *Journal of Environmental Science, Computer Science and Engineering & Technology*, vol. 8, no. 3, aug 2019. [Online]. Available: <https://doi.org/10.24214%2Fjecet.b.8.3.23641>
- [105] Y. Jang, H. Gunes, and I. Patras, “Registration-free face-SSD: Single shot analysis of smiles, facial attributes, and affect in the wild,” *Computer Vision and Image Understanding*, vol. 182, pp. 17–29, may 2019. [Online]. Available: <https://doi.org/10.1016%2Fj.cviu.2019.01.006>
- [106] B. Jiang, Q. Ren, F. Dai, J. Xiong, J. Yang, and G. Gui, “Multi-task cascaded convolutional neural networks for real-time dynamic face recognition method,” in *Lecture Notes in Electrical Engineering*. Springer Singapore, jun 2019, pp. 59–66. [Online]. Available: [https://doi.org/10.1007%2F978-981-13-6508-9\\_8](https://doi.org/10.1007%2F978-981-13-6508-9_8)
- [107] S. V. Fernandes and M. S. Ullah, “A comprehensive review on features extraction and features matching techniques for deception detection,” *IEEE Access*, vol. 10, pp. 28 233–28 246, 2022. [Online]. Available: <https://doi.org/10.1109%2Faccess.2022.3157821>
- [108] V. Costa, A. Sousa, and A. Reis, “Image-based object spoofing detection,” in *Lecture Notes in Computer Science*. Springer International Publishing, 2018, pp. 189–201. [Online]. Available: [https://doi.org/10.1007%2F978-3-030-05288-1\\_15](https://doi.org/10.1007%2F978-3-030-05288-1_15)
- [109] S. CM and M. S. and, “An image quality assessment of multi-exposure image fusion by improving SSIM,” *International Journal of Trend in Scientific Research and Development*, vol. Volume-2, no. Issue-4, pp. 2780–2784, jun 2018. [Online]. Available: <https://doi.org/10.31142%2Fijtsrd15634>
- [110] Minivision-Ai, “Minivision-ai/silent-face-anti-spoofing: silent-face-anti-spoofin.” [Online]. Available: <https://github.com/minivision-ai/Silent-Face-Anti-Spoofing>
- [111] “Haar-cascade classifier.” [Online]. Available: [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html)
- [112] Minivision-Ai, “Minivision-ai/silent-face-anti-spoofing-apk.” [Online]. Available: <https://github.com/minivision-ai/Silent-Face-Anti-Spoofing-APK>

- 
- [113] A. Rosebrock, “Face detection with dlib (hog and cnn),” Apr 2021. [Online]. Available: <https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>
- [114] “Opencv dnn module and deep learning (a definitive guide),” Jan 2023. [Online]. Available: <https://learnopencv.com/deep-learning-with-opencvs-dnn-module-a-definitive-guide/>
- [115] C. Ning, H. Zhou, Y. Song, and J. Tang, “Inception single shot MultiBox detector for object detection,” in *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, jul 2017. [Online]. Available: <https://doi.org/10.1109%2Ficmew.2017.8026312>
- [116] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, oct 2016. [Online]. Available: <https://doi.org/10.1109%2Flsp.2016.2603342>
- [117] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. [Online]. Available: <https://doi.org/10.1109%2Fcvpr.2015.7298682>
- [118] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014. [Online]. Available: <https://doi.org/10.1109%2Fcvpr.2014.220>
- [119] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015. [Online]. Available: <https://doi.org/10.5244%2Fc.29.41>
- [120] A. S. Sanchez-Moreno, J. Olivares-Mercado, A. Hernandez-Suarez, K. Toscano-Medina, G. Sanchez-Perez, and G. Benitez-Garcia, “Efficient face recognition system for operating in unconstrained environments,” *Journal of Imaging*, vol. 7, no. 9, p. 161, aug 2021. [Online]. Available: <https://doi.org/10.3390%2Fjimaging7090161>
- [121] R. Padilha, F. A. Andaló, G. Bertocco, W. R. Almeida, W. Dias, T. Resek, R. da S. Torres, J. Wainer, and A. Rocha, “Two-tiered face verification with low-memory footprint for mobile devices,” *IET Biometrics*, vol. 9, no. 5, pp. 205–215, jul 2020. [Online]. Available: <https://doi.org/10.1049%2Fiet-bmt.2020.0031>
- [122] [Online]. Available: <https://www.acm.org/binaries/content/assets/public-policy/ustpc-statement-remote-test-admin-systems.pdf>
- [123] N. Mishra, R. K. Singh, and S. K. Yadav, “Design a new protocol for vulnerability detection in cloud computing security improvement,” *SSRN Electronic Journal*, 2021. [Online]. Available: <https://doi.org/10.2139%2Fssrn.3842825>

- 
- [124] W. Li, J. Wu, J. Cao, N. Chen, Q. Zhang, and R. Buyya, "Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions," *Journal of Cloud Computing*, vol. 10, no. 1, jun 2021. [Online]. Available: <https://doi.org/10.1186%2Fs13677-021-00247-5>
- [125] M. A. Ferrag, L. Maglaras, A. Derhab, and H. Janicke, "Authentication schemes for smart mobile devices: threat models, countermeasures, and open research issues," *Telecommunication Systems*, vol. 73, no. 2, pp. 317–348, sep 2019. [Online]. Available: <https://doi.org/10.1007%2Fs11235-019-00612-5>
- [126] S. Ayeswarya and J. Norman, "A survey on different continuous authentication systems," *International Journal of Biometrics*, vol. 11, no. 1, p. 67, 2019. [Online]. Available: <https://doi.org/10.1504%2Fijbm.2019.10016811>
- [127] P. Liu, S. H. Shirazi, W. Liu, and Y. Xie, "pKAS: A secure password-based key agreement scheme for the edge cloud," *Security and Communication Networks*, vol. 2021, pp. 1–10, oct 2021. [Online]. Available: <https://doi.org/10.1155%2F2021%2F6571700>
- [128] S. P. Otta and S. Panda, "Cloud identity and access management solution with blockchain," in *Intelligent Systems Reference Library*. Springer International Publishing, 2021, pp. 243–270. [Online]. Available: [https://doi.org/10.1007%2F978-3-030-69395-4\\_14](https://doi.org/10.1007%2F978-3-030-69395-4_14)
- [129] T. M. Alsultan, A. A. Salam, K. A. Alissa, and N. A. Saqib, "A comparative study of biometric authentication in cloud computing," in *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, jun 2019. [Online]. Available: <https://doi.org/10.1109%2Fisncc.2019.8909117>
- [130] P. J. Windley, "Sovrin: An identity metasystem for self-sovereign identity," *Frontiers in Blockchain*, vol. 4, jul 2021. [Online]. Available: <https://doi.org/10.3389%2Ffbloc.2021.626726>
- [131] N. Naik and P. Jenkins, "uPort open-source identity management system: An assessment of self-sovereign identity and user-centric data platform built on blockchain," in *2020 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, oct 2020. [Online]. Available: <https://doi.org/10.1109%2Fisse49799.2020.9272223>
- [132] A. Gruner, A. Muhle, and C. Meinel, "Analyzing interoperability and portability concepts for self-sovereign identity," in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, oct 2021. [Online]. Available: <https://doi.org/10.1109%2Ftrustcom53373.2021.00089>
- [133] M. Shuaib, S. Alam, M. S. Alam, and M. S. Nasir, "Self-sovereign identity for healthcare using blockchain," *Materials Today: Proceedings*, mar 2021. [Online]. Available: <https://doi.org/10.1016%2Fj.matpr.2021.03.083>
- [134] R. Mukta, H.-Y. Paik, Q. Lu, and S. S. Kanhere, "CredTrust: Credential based

- issuer management for trust in self-sovereign identity,” in *2022 IEEE International Conference on Blockchain (Blockchain)*. IEEE, aug 2022. [Online]. Available: <https://doi.org/10.1109%2Fblockchain55522.2022.00053>
- [135] J. D. Roberts, J. F. Defranco, and D. R. Kuhn, “Data block matrix and hyperledger implementation: Extending distributed ledger technology for privacy requirements,” *Distributed Ledger Technologies: Research and Practice*, feb 2023. [Online]. Available: <https://doi.org/10.1145%2F3585539>
- [136] T. Peng, M. Li, F. Chen, Y. Xu, and D. Zhang, “Learning efficient facial landmark model for human attractiveness analysis,” *Pattern Recognition*, vol. 138, p. 109370, jun 2023. [Online]. Available: <https://doi.org/10.1016%2Fj.patcog.2023.109370>
- [137] Z. Fan and Y. peng Guan, “A deep learning framework for face verification without alignment,” *Journal of Real-Time Image Processing*, vol. 18, no. 4, pp. 999–1009, oct 2020. [Online]. Available: <https://doi.org/10.1007%2Fs11554-020-01037-z>
- [138] F. Boutros, N. Damer, M. Fang, F. Kirchbuchner, and A. Kuijper, “MixFaceNets: Extremely efficient face recognition networks,” in *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, aug 2021. [Online]. Available: <https://doi.org/10.1109%2Fijcb52358.2021.9484374>
- [139] S. Nižetić, P. Šolić, D. L. de-Ipiña González-de Artaza, and L. Patrono, “Internet of things (IoT): Opportunities, issues and challenges towards a smart and sustainable future,” *Journal of Cleaner Production*, vol. 274, p. 122877, nov 2020. [Online]. Available: <https://doi.org/10.1016%2Fj.jclepro.2020.122877>
- [140] E. S. Babu, A. K. Dadi, K. K. Singh, S. R. Nayak, A. K. Bhoi, and A. Singh, “A distributed identity-based authentication scheme for internet of things devices using permissioned blockchain system,” vol. 39, no. 10, feb 2022. [Online]. Available: <https://doi.org/10.1111%2Fexsy.12941>
- [141] M. Liyanage, A. Braeken, P. Kumar, and M. Ylianttila, Eds., *IoT Security*. Wiley, dec 2019. [Online]. Available: <https://doi.org/10.1002%2F9781119527978>
- [142] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, “A survey of internet of things (IoT) authentication schemes,” *Sensors*, vol. 19, no. 5, p. 1141, mar 2019. [Online]. Available: <https://doi.org/10.3390%2Fs19051141>
- [143] M. I. G. Vasco, A. P. D. Pozo, and C. Soriente, “A key for john doe: modeling and designing anonymous password-authenticated key exchange protocols,” *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2019. [Online]. Available: <https://doi.org/10.1109%2Ftdsc.2019.2919013>
- [144] Z. Gong-Guo and Z. Wan, “Blockchain-based IoT security authentication system,” in *2021 International Conference on Computer, Blockchain and Financial Development (CBFD)*. IEEE, apr 2021. [Online]. Available: <https://doi.org/10.1109%2Fcbfd52659.2021.00090>

- [145] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4132–4156, mar 2021. [Online]. Available: [https://doi.org/10.1109%2Fjiot.2020.3026493](https://doi.org/10.1109/2Fjiot.2020.3026493)
- [146] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*. Springer Berlin Heidelberg, pp. 47–53. [Online]. Available: [https://doi.org/10.1007%2F3-540-39568-7\\_5](https://doi.org/10.1007%2F3-540-39568-7_5)
- [147] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in internet of things: Taxonomies and open challenges," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 796–809, jul 2018. [Online]. Available: <https://doi.org/10.1007%2Fs11036-018-1089-9>
- [148] D. Letz, "Blockquick: Super-light client protocol for blockchain validation on constrained devices," *Cryptology ePrint Archive*, 2019.
- [149] M. Y. Kubilay, M. S. Kiraz, and H. A. Mantar, "CertLedger: A new PKI model with certificate transparency based on blockchain," *Computers & Security*, vol. 85, pp. 333–352, aug 2019. [Online]. Available: <https://doi.org/10.1016%2Fj.cose.2019.05.013>
- [150] B. Bunz, L. Kiffer, L. Luu, and M. Zamani, "FlyClient: Super-light clients for cryptocurrencies," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, may 2020. [Online]. Available: <https://doi.org/10.1109%2Fsp40000.2020.00049>
- [151] A. Yakubov, W. Shbair, N. Khan, R. State, C. Medinger, and J. Hilger, "BlockPGP: A blockchain-based framework for PGP key servers," *International Journal of Networking and Computing*, vol. 10, no. 1, pp. 1–24, 2020. [Online]. Available: [https://doi.org/10.15803%2Fijnc.10.1\\_1](https://doi.org/10.15803%2Fijnc.10.1_1)
- [152] M. Toorani and C. Gehrman, "A decentralized dynamic PKI based on blockchain," in *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. ACM, mar 2021. [Online]. Available: <https://doi.org/10.1145%2F3412841.3442038>
- [153] M. Simic, M. Vucetic, T. Unkasevic, Z. Banjac, and M. Stankovic, "Entity identification and security solutions in IoT based on PKI and blockchain technology," in *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, mar 2020. [Online]. Available: <https://doi.org/10.1109%2Finfoteh48170.2020.9066282>
- [154] M. Mumtaz, J. Akram, and L. Ping, "An RSA based authentication system for smart IoT environment," in *2019 IEEE 21st International Conference on High Performance Computing and Communications: IEEE 17th International Conference on Smart City : IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE, aug 2019. [Online]. Available: <https://doi.org/10.1109%2Fhpcc%2Fsmartcity%2Fdss.2019.00112>
- [155] I. ul haq, J. Wang, and Y. Zhu, "Secure two-factor lightweight authentication

- protocol using self-certified public key cryptography for multi-server 5g networks,” *Journal of Network and Computer Applications*, vol. 161, p. 102660, jul 2020. [Online]. Available: <https://doi.org/10.1016%2Fj.jnca.2020.102660>
- [156] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. Doostari, “A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT,” *Computer Networks*, vol. 177, p. 107333, aug 2020. [Online]. Available: <https://doi.org/10.1016%2Fj.comnet.2020.107333>
- [157] N. Radhakrishnan and A. P. Muniyandi, “Dependable and provable secure two-factor mutual authentication scheme using ECC for IoT-based telecare medical information system,” *Journal of Healthcare Engineering*, vol. 2022, pp. 1–15, feb 2022. [Online]. Available: <https://doi.org/10.1155%2F2022%2F9273662>
- [158] X. Jia, N. Hu, S. Su, S. Yin, Y. Zhao, X. Cheng, and C. Zhang, “IRBA: An identity-based cross-domain authentication scheme for the internet of things,” *Electronics*, vol. 9, no. 4, p. 634, apr 2020. [Online]. Available: <https://doi.org/10.3390%2Felectronics9040634>
- [159] E. Lara, L. Aguilar, M. A. Sanchez, and J. A. García, “Lightweight authentication protocol for m2m communications of resource-constrained devices in industrial internet of things,” *Sensors*, vol. 20, no. 2, p. 501, jan 2020. [Online]. Available: <https://doi.org/10.3390%2Fs20020501>
- [160] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, “Quark: A lightweight hash,” *Journal of Cryptology*, vol. 26, no. 2, pp. 313–339, may 2012. [Online]. Available: <https://doi.org/10.1007%2Fs00145-012-9125-6>
- [161] T. Mangole, A. S. J. Helberg, and K. K. Nair, “Resource usage evaluation of the PHOTON hash function,” in *2022 Conference on Information Communications Technology and Society (ICTAS)*. IEEE, mar 2022. [Online]. Available: <https://doi.org/10.1109%2Fictas53252.2022.9744686>
- [162] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı, and I. Verbauwhede, “spongent: A lightweight hash function,” in *Cryptographic Hardware and Embedded Systems – CHES 2011*. Springer Berlin Heidelberg, 2011, pp. 312–325. [Online]. Available: [https://doi.org/10.1007%2F978-3-642-23951-9\\_21](https://doi.org/10.1007%2F978-3-642-23951-9_21)
- [163] “A lightweight hash function based on enhanced chaotic map algorithm(keccak),” *Iraqi Journal of Computer, Communication, Control and System Engineering*, pp. 53–62, jun 2022. [Online]. Available: <https://doi.org/10.33103%2Fuot.ijccce.22.2.5>
- [164] B. Li, J. Lin, Q. Wang, Z. Wang, and J. Jing, “Locally-centralized certificate validation and its application in desktop virtualization systems,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1380–1395, 2021. [Online]. Available: <https://doi.org/10.1109%2Ftifs.2020.3035265>
- [165] M. Anathi and K. Vijayakumar, “An intelligent approach for dynamic network traffic restriction using MAC address verification,” *Computer Communications*,



- vol. 154, pp. 559–564, mar 2020. [Online]. Available: <https://doi.org/10.1016%2Fj.comcom.2020.02.021>
- [166] P. Jiang, H. Wu, and C. Xin, “A channel state information based virtual MAC spoofing detector,” *High-Confidence Computing*, vol. 2, no. 3, p. 100067, sep 2022. [Online]. Available: <https://doi.org/10.1016%2Fj.hcc.2022.100067>
- [167] E. K. Subramanian and L. Tamilselvan, “Elliptic curve diffie–hellman cryptosystem in big data cloud security,” *Cluster Computing*, vol. 23, no. 4, pp. 3057–3067, feb 2020. [Online]. Available: <https://doi.org/10.1007%2Fs10586-020-03069-3>
- [168] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, “Zero trust architecture,” Tech. Rep., aug 2020. [Online]. Available: <https://doi.org/10.6028%2Fnist.sp.800-207>
- [169] S. W. Shah, N. F. Syed, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, “LCDA: Lightweight continuous device-to-device authentication for a zero trust architecture (ZTA),” *Computers & Security*, vol. 108, p. 102351, sep 2021. [Online]. Available: <https://doi.org/10.1016%2Fj.cose.2021.102351>
- [170] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, “Zero trust architecture (ZTA): A comprehensive survey,” *IEEE Access*, vol. 10, pp. 57 143–57 179, 2022. [Online]. Available: <https://doi.org/10.1109%2Faccess.2022.3174679>
- [171] K. Ragothaman, Y. Wang, B. Rimal, and M. Lawrence, “Access control for IoT: A survey of existing research, dynamic policies and future directions,” *Sensors*, vol. 23, no. 4, p. 1805, feb 2023. [Online]. Available: <https://doi.org/10.3390%2Fs23041805>
- [172] P. Phiayura and S. Teerakanok, “A comprehensive framework for migrating to zero trust architecture,” *IEEE Access*, vol. 11, pp. 19 487–19 511, 2023. [Online]. Available: <https://doi.org/10.1109%2Faccess.2023.3248622>
- [173] L. Dong, Z. Niu, Y. Zhu, and W. Zhang, “Specifying and verifying SDP protocol based zero trust architecture using TLA, booktitle = Proceedings of the 7th International Conference on Cyber Security and Information Engineering.” ACM, sep 2022. [Online]. Available: <https://doi.org/10.1145%2F3558819.3558826>
- [174] A. P. Gonzalez, J. M. R. López, V. G. Díaz, and A. G. Gómez, “Securing the software-defined perimeter framework with automated security configuration deployment systems,” *SSRN Electronic Journal*, 2023. [Online]. Available: <https://doi.org/10.2139%2Fssrn.4326496>
- [175] J. Koilpillai and N. A. Murray, “Software defined perimeter (sdp) and zero trust,” *Cloud Security Alliance (CSA)*, 2020.
- [176] A. Sallam, A. Refaey, and A. Shami, “Securing smart home networks with software-defined perimeter,” in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, jun 2019. [Online]. Available: <https://doi.org/10.1109%2Fiwcmc.2019.8766686>

- 
- [177] A. Moubayed, A. Refaey, and A. Shami, "Software-defined perimeter (SDP): State of the art secure solution for modern networks," *IEEE Network*, vol. 33, no. 5, pp. 226–233, sep 2019. [Online]. Available: <https://doi.org/10.1109%2Fmnet.2019.1800324>
- [178] J. Singh, A. Refaey, and J. Koilpillai, "Adoption of the software-defined perimeter (SDP) architecture for infrastructure as a service," *Canadian Journal of Electrical and Computer Engineering*, vol. 43, no. 4, pp. 357–363, 2020. [Online]. Available: <https://doi.org/10.1109%2Fcjece.2020.3005316>
- [179] S. Mandal, D. A. Khan, and S. Jain, "Cloud-based zero trust access control policy: An approach to support work-from-home driven by COVID-19 pandemic," *New Generation Computing*, vol. 39, no. 3-4, pp. 599–622, jun 2021. [Online]. Available: <https://doi.org/10.1007%2Fs00354-021-00130-6>
- [180] D. Yang, Y. Zhao, K. Wu, X. Guo, and H. Peng, "An efficient authentication scheme based on zero trust for UAV swarm," in *2021 International Conference on Networking and Network Applications (NaNA)*. IEEE, oct 2021. [Online]. Available: <https://doi.org/10.1109%2Fnana53684.2021.00068>
- [181] V. Krishnan and C. S. Sreeja, "Zero trust-based adaptive authentication using composite attribute set," in *2021 IEEE 3rd PhD Colloquium on Ethically Driven Innovation and Technology for Society (PhD EDITS)*. IEEE, nov 2021. [Online]. Available: <https://doi.org/10.1109%2Fphdedits53295.2021.9649474>
- [182] X. Zhou, W. Liang, J. She, Z. Yan, and K. Wang, "Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5308–5317, jun 2021. [Online]. Available: <https://doi.org/10.1109%2Ftvt.2021.3077893>
- [183] M. Hamdan, E. Hassan, A. Abdelaziz, A. Elhigazi, B. Mohammed, S. Khan, A. V. Vasilakos, and M. Marsono, "A comprehensive survey of load balancing techniques in software-defined network," *Journal of Network and Computer Applications*, vol. 174, p. 102856, jan 2021. [Online]. Available: <https://doi.org/10.1016%2Fj.jnca.2020.102856>

# List of Publications

## Peer-reviewed Journals

1. Soumya Prakash Otta, Subhrakanta Panda, Maanak Gupta, Chittaranjan Hota “A Systematic Survey of Multi-Factor Authentication for Cloud Infrastructure”. **Future Internet**. MDPI. <https://doi.org/10.3390/fi15040146>

## Peer-reviewed Conferences

1. Soumya Prakash Otta, Subhrakanta Panda “Decentralized Identity and Access Management of Cloud for Security as a Service”. In *14th International Conference on COMMunication Systems & NETWORKS (COMSNETS)* 2022 Jan 4-8 (pp. 299-303). **IEEE**. Bengaluru, India. <https://doi.org/10.3390/fi15040146>
2. Soumya Prakash Otta, Siddharth Kolipara, Subhrakanta Panda, Chittaranjan Hota “User Identification with Face Recognition : A Systematic Analysis”. In *3rd International Conference for Emerging Technology (INCET)* 2022 May 27-29 (pp. 01-06). **IEEE**. Belgaum, India. <https://doi.org/10.1109/incet54531.2022.9825108>
3. Soumya Prakash Otta, Siddharth Kolipara, Vijay Kumar Malhotra, Aman Raj Singh, Subhrakanta Panda, Chittaranjan Hota “Continuous Cloud User Authentication By Efficient Facial Recognition”. In *5th International Conference on Computational Intelligence and Networks (CINE)* 2022 Dec 1-3 (pp. 01-06). **IEEE**. Bhubaneswar, India. <https://doi.org/10.1109/cine56307.2022.10037567>

4. Soumya Prakash Otta, Subhrakanta Panda, Chittaranjan Hota “Identity Management with Blockchain: Indian Migrant Workers Prospective”. In *International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)* 2022 Dec 21-23 (pp. 01-06). **IEEE. Gwalior, India.** <https://doi.org/10.1109/IATMSI56455.2022.10119388>

## Peer-reviewed Book Chapters

1. Soumya Prakash Otta, Subhrakanta Panda “Cloud Identity and Access Management Solution with Blockchain”. pp. 243-270 **Blockchain Technology: Applications and Challenges. Springer 2021.** [https://doi.org/10.1007/978-3-030-69395-4\\_14](https://doi.org/10.1007/978-3-030-69395-4_14)
2. Soumya Prakash Otta, Subhrakanta Panda “Identity and Access Management for Internet of Things Cloud”. pp. 43-66 **The New Advanced Society: Artificial Intelligence and Industrial Internet of Things Paradigm. Wiley 2022.** <https://doi.org/10.1002/9781119884392.ch3>

# Biographies

## Brief Biography of the Candidate

Soumya Prakash Otta is a PhD Scholar at the Computer Science & Information Systems Department of Birla Institute of Technology and Science, Pilani - Hyderabad Campus. Prior to pursuing PhD he completed his MS in Software Systems from BITS Pilani in 2014. He is currently pursuing his Ph.D. degree in the Department of Computer Science and Information Systems, BITS-Pilani Hyderabad Campus under the supervision of Prof. Subhrakanta Panda. Since 24 years, he has been associated with management of data communication and cloud infrastructure of Indian Armed Forces. Presently he is serving as Associate Director of Operations in charge of Communication and Cyber Security in the Ministry of Defence, Govt of India. His research interests lie primarily in the areas of Cloud Computing Security, Security of Cyber Physical System, Zero-Trust Networks, Identity Management and Distributed Ledger Technologies.

## Brief Biography of the Supervisor

Prof. Subhrakanta Panda is currently working as an Associate Professor in the Department of Computer Science and Information Systems, BITS-PILANI, Hyderabad Campus. He received his Ph.D. in Computer Science and Engineering from NIT Rourkela, under the supervision of Dr. D. P. Mohapatra. His research interests include Software Testing, Social Network Analysis, Predictive Analytics, and Cloud Computing. He has published more than 30 scholarly peer reviewed research articles in reputed International Journals and International Conferences. He adapted the Indian edition of the book Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael Goldwasser,

Wiley, Print (ISBN: 9789354247866, eISBN: 9789354248351). In addition to research, he has guided 30 M.Tech. students and 50 B. Tech. students in various project related courses. He is a life member of CSI, ISTE, ISC, and OITS and member in ACM and IEEE. He has reviewed many papers for Journal of Applied Network Science, Springer: International Journal of Business Information Systems, Elsevier: Journal of King Saud University - Computer And Information Sciences, MoSICom - 2020, TENCON - 2019, ICACIE - 2018 and ICACIE - 2017.

### **Brief Biography of the Co-Supervisor**

Prof. Chittaranjan Hota, Senior Professor of Computer Sc. at BITS, Hyderabad completed his Ph.D. from BITS, Pilani and has more than thirty years of academic, research and administrative experience in Indian and universities abroad. His Ph.D. work was on QoS assurances in IP-Virtual Private Networks, a significant part of which was carried out at University of New South Wales, Sydney, Australia during his visit to Network Research Laboratory at UNSW under Indo-Australian joint research program. Prior to Ph.D., he had earned B.E in Computer Engineering, and M.E in Computer Sc. and Engineering. He had been in merit through his bachelors & master's degrees in engineering.

His research interests include developing algorithms and models for building systems and applications in areas like Big-data analytics, Cognitive IoT, and Cyber security in which he has more than 140 publications, in various journals and conferences. He has guided more than 15 PhD students. He is currently working on building secure Bio-CPS devices under National Mission on Interdisciplinary Cyber Physical Systems with funding support from DST (GoI). In the past, he has executed funded projects with support from MeitY, DeitY, UGC, NWO (NL), Intel, Progress software, and TCS in the areas of Network threat monitoring, Code tamper-proofing on IoT devices, and building secure Peer-to-Peer overlays using Artificial Intelligence, and Machine Learning techniques.