



REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

**INTEGRATE LIGHTWEIGHT CRYPTOGRAPHY
AND MONITORING FRAMEWORK FOR SECURE
AND EFFICIENT IOT-ENABLED CLOUD SYSTEM**

Zaid Abdulsalam Ibrahim ALMATWARI

Doctor of Philosophy

Supervisor

Asst. Prof. Dr. Muhammad ILYAS

İstanbul, 2025

**INTEGRATE LIGHTWEIGHT CRYPTOGRAPHY AND
MONITORING FRAMEWORK FOR SECURE AND EFFICIENT
IOT-ENABLED CLOUD SYSTEM**

Zaid Abdulsalam Ibrahim ALMATWARI

Electrical and Computer Engineering

Doctor of Philosophy

ALTINBAŞ UNIVERSITY

2025

The dissertation titled INTEGRATE LIGHTWEIGHT CRYPTOGRAPHY AND MONITORING FRAMEWORK FOR SECURE AND EFFICIENT IOT-ENABLED CLOUD SYSTEM prepared by ZAID ABDULSALAM IBRAHIM ALMATWARI and submitted on 16/06/2025 has been **accepted unanimously** for the degree of PhD in Electrical and Computer Engineering.

Asst. Prof. Dr. Muhammad ILYAS

Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Muhammad ILYAS	Department of Computer Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Oğuz ATA	Department of Software Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Hakan KOYUNCU	Department of Computer Engineering, Altınbaş University	_____
Assoc. Prof. Dr. Jawad RASHEED	Department of Computer Engineering, İstanbul Sabahattin Zaim University	_____
Assoc. Prof. Dr. Alaa HAMITOGLU	Department of Computer Engineering, İstinye University	_____

I hereby declare that this dissertation meets all format and submission requirements for PhD dissertation.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Zaid Abdulsalam Ibrahim ALMATWARI

Signature



DEDICATION

To my beloved father, whose wisdom, strength, and unwavering support have been my success. Your belief in me has been my greatest motivation, and your guidance has been my compass. This achievement is a testament to your enduring influence and encouragement.

To my dear mother, whose boundless love, patience, and sacrifices have nurtured my growth and dreams. Your constant support and comforting presence have been my refuge throughout this journey.

To my esteemed adviser, whose guidance, wisdom, and insight have been instrumental in shaping this work. Your mentorship has been an invaluable asset in my academic journey.

My brothers and sisters have always stood by me, providing comfort and camaraderie. Your love and support have been invaluable throughout this journey.

To my precious children, whose smiles and laughter have been the sunshine in my life, you have motivated me to strive for greatness and given me the joy to endure the challenges.

This PhD is dedicated to all of you with my deepest gratitude and love.

ABSTRACT

INTEGRATE LIGHTWEIGHT CRYPTOGRAPHY AND MONITORING FRAMEWORK FOR SECURE AND EFFICIENT IOT- ENABLED CLOUD SYSTEM

ALMATWARI, Zaid Abdulsalam Ibrahim

Ph.D., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Muhammad ILYAS

Date: 06/2025

Pages: 180

In the present generation, security issues are a major issue experienced by everyone due to the increase in digital technologies. With the introduction of digital technologies, data is stored and transmitted through platforms where security issues arise. Many cryptographic techniques are available to protect data in transmission and storage through digital technologies to overcome security issues. The methods are utilized to encrypt the data with passwords or PIN codes, which hackers can easily identify. In this case, for a safe process, a hybrid encryption method is proposed in this thesis for security data purposes, integrated with lightweight and hidden ciphertext policy attributes with the assistance of the cloud IoT environment. The lightweight encryption provides proxy services for data authentication. The analysis of the security has provided a positive impact on the encryption of the integrated lightweight. The system offers better positive results with hybrid encryption for data authentication for security purposes. The third part of the framework introduces a hybrid encryption system that combines a hidden hierarchical policy with a lightweight Ciphertext-Policy Attribute-Based Encryption (CP-ABE) enforcement model. This is to ensure fine-grained access control, minimal computational burden, and efficient encryption-decryption suited for IoT devices. It also incorporates QGE-HMAC for secure message authentication and ELP-DSA for signature validation, providing strong protection for encryption key integrity and end-to-end data authenticity. The final part presents a reliable cloud-enabled IoT network monitoring system powered by JPAO-GRU. This advanced gated recurrent unit model integrates Aranda-Ordaz activation and Jeffreys prior regularization for accurate anomaly detection. Additionally, the framework includes HTT-Fuzzy logic for outdated device detection, QASOA for intelligent load balancing, and LCA-BST for low-latency,

hierarchical device monitoring. Comprehensive experimental validation shows that the proposed dual approach enhances security, integrity, and network efficiency, achieving high accuracy (99.12%) in intrusion detection, 99.18% encryption security, and substantial reductions in encryption time, latency, and energy consumption. This research contributes a unified and scalable solution for addressing the growing security, authentication, and trust challenges in next-generation cloud-IoT systems.

Keywords: Authentication, Hybrid Encryption, Integrated lightweight, Hidden Hierarchy Cipher Text-policy Attribute, Lightweight Cryptography, Quantum Gibbs Entropy, Hash-Based Message Authentication Code.



TABLE OF CONTENTS

	<u>Pages</u>
ABSTRACT	vi
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
ABBREVIATIONS.....	xv
LIST OF SYMBOLS.....	xix
1. INTRODUCTION.....	1
1.1 BACKGROUND OF THE STUDY	1
1.2 INTRODUCTION TO CLOUD SYSTEM IN IOT ENVIRONMENT	4
1.2.1 Benefits and Functions of IoT Cloud Computing.....	5
1.2.2 Well-Designed Cloud Architecture.....	6
1.2.3 Types of Cloud Computing.....	7
1.2.3.1 Infrastructure as a service (IaaS).....	7
1.2.3.2 Platform as a service (PaaS).....	7
1.2.3.3 Software as a service (SaaS)	8
1.2.3.4 Function as a service (FaaS)	8
1.2.3.5 Database as a service (DBaaS).....	9
1.2.3.6 Containers as a service (CaaS).....	9
1.2.3.7 Integration platform as a service (iPaaS)	9
1.2.3.8 Network as a service (NaaS)	10
1.2.3.9 Storage as a service (STaaS).....	10
1.3 THE ROLE OF CLOUD COMPUTING IN EMPOWERING THE INTERNET OF THINGS	11
1.4 HYBRID ENCRYPTION METHOD FOR DATA AUTHENTICATION.....	12
1.4.1 The Key Components	13
1.4.2 The Encryption and Decryption Process.....	14
1.4.3 Decryption (Recipient's Perspective).....	14

1.4.4 Advantages of the Hybrid Encryption Method	15
1.5 CIPHERTEXT-POLICY ATTRIBUTE IN CLOUD SYSTEM USING IOT	16
1.6 MOTIVATION FOR THE RESEARCH.....	19
1.7 AIMS AND OBJECTIVES	20
1.8 SIGNIFICANCE OF THE STUDY	21
1.9 THESIS ORGANIZATION.....	22
2. LITERATURE REVIEW.....	24
2.1 CLUSTERING-BASED METHOD	25
2.2 ML-BASED METHOD	25
2.3 SCHEDULING METHOD.....	26
2.4 SUMMARY	27
3. RESEARCH METHODOLOGY	32
3.1 SYSTEM DESIGN AND ARCHITECTURE	32
3.2 PROPOSED SYSTEM MODEL	34
3.2.1 Lightweight Authenticated Encryption.....	40
3.2.2 Cipher Text-policy Attribute-Based Encryption (CP-ABE).....	45
3.3 LIGHTWEIGHT AUTHENTICATION ENCRYPTION FOR SECURING VALIDITY AND CONFIDENTIALITY OF SYSTEM COMMUNICATION	48
3.4 CIPHER TEXT-POLICY ATTRIBUTE-BASED ENCRYPTION (CP-ABE)	51
3.5 ENCRYPTION KEY SECURITY-AWARE RELIABLE DATA TRANSMISSION IN IOT-ENABLED CLOUD APPLICATION.....	56
3.5.1 Proposed Scheme	59
3.5.1.1 IoT network initialization and registration.....	60
3.5.1.2 Micro clustering	60
3.5.1.3 CH selection	62
3.5.1.4 Graph construction	63
3.5.1.5 MAC creation.....	66
3.5.1.6 Digital signature creation	67
3.5.1.7 Data Sensing	70

3.5.1.8 Data encryption	71
3.5.1.9 Digital signature verification.....	73
3.5.1.10 MAC verification	74
3.5.1.11 Data decryption	75
3.5.2 Summary	76
3.6 PROPOSED RELIABLE CLOUD-ASSISTED IOT NETWORK MONITORING	76
3.6.1 Proposed Methodology	79
3.6.1.1 IoT network initialization and device registration	81
3.6.1.2 Data sensing	81
3.6.1.3 Attribute extraction	82
3.6.1.4 Network monitoring tool.....	83
3.6.1.5 Outdated device identification	84
3.6.1.6 Data security.....	85
3.6.1.7 CPJTABE: Ciphertext-Policy Jacobi Theta Attribute-Based Encryption.....	85
3.6.1.8 Load balancing	88
3.6.1.9 Intrusion detection system	90
3.6.1.10 Intrusion detection using JPAO-GRU.....	92
3.6.1.11 Data authentication	95
3.6.2 Summary	96
4. RESULTS	97
4.1 RESULTS	97
4.1.1 Cipher Text-Policy Attribute-Based Encryption	97
4.1.2 Hybrid Cipher Text Policy Attribute-Based Encryption (H-CP-ABE)	98
4.1.3 Hybrid Cipher Text Policy Attribute-Based Encryption (ABE).....	101
4.1.4 Integrated lightweight and hidden Hybrid Cipher Text Policy Attribute-Based Encryption (IL-HH-CP-ABE).....	103
4.1.5 Implementation of Encryption Key Security-Aware Reliable Data Transmission in IoT-Enabled Cloud Applications	116
4.1.5.1 Performance assessment	117

4.1.5.2 Comparative validation	122
4.1.5.3 Key findings summary	123
4.1.6 Implementation of Reliable Cloud-assisted IoT Network Monitoring	123
4.1.6.1 Dataset description	124
4.1.6.2 Performance analysis	125
4.1.6.3 Comparative analysis	132
4.1.6.4 Key findings summary	133
4.2 PERFORMANCE EVALUATION	134
4.2.1 Communication Overhead Metrics	134
4.2.2 Computational Overhead Metrics	134
5. CONCLUSION AND FUTURE WORK	135
5.1 SUMMARY OF CONTRIBUTIONS	138
5.2 LIMITATIONS AND FUTURE SCOPE OF THE RESEARCH	141
REFERENCES	146

LIST OF TABLES

	<u>Pages</u>
Table 2.1: Comparative for Integrated Lightweight and Hidden Hierarchy Cipher Text-policy Attribute in Cloud-Assisted IoT	28
Table 4.1: Encryption Time Comparison	110
Table 4.2: Decryption Time Comparison	112
Table 4.3: Storage Analysis Comparison	113
Table 4.4: Key Generation Comparison	115
Table 4.5: Graph Construction Time Analysis	118
Table 4.6: Security Level and Attack Level Assessment	120
Table 4.7: Comparative Validation	122
Table 4.8: Dataset Characteristics	125
Table 4.9: Tree Creation Time	125
Table 4.10: Tree Searching Time	126
Table 4.11: Comparative Assessment	132

LIST OF FIGURES

	<u>Pages</u>
Figure 1.1: Data Lifecycle and Cloud Components Interaction Model	7
Figure 3.1: System Architecture for Cloud-assisted IoT Systems.....	32
Figure 3.2: Proposed System Framework (HH-CP-ABE)	33
Figure 3.3: Proposed LWC-ABE System Model	36
Figure 3.4: Hybrid Lightweight Signcryption	37
Figure 3.5: Registration Phase.....	41
Figure 3.6: Key Generation and Exchange Phase	44
Figure 3.7: Message Signing and Encryption Phase	44
Figure 3.8: Message Decryption and Verification Phase	45
Figure 3.9: Conventional Public Key	52
Figure 3.10: The Proposed Approach's Block Diagram	59
Figure 3.11: A Typical GCK-DAG with 7 Nodes.....	64
Figure 3.12: Pseudo Code for ELP-DSA's.....	70
Figure 3.13: Pseudocode of SArcCC.....	73
Figure 3.14: The Research Methodology's Design.....	80
Figure 3.15: The Pseudocode of The Proposed QASOA	89
Figure 3.16: The structural Representation of the Proposed JPAO-GRU.....	93
Figure 3.17: The Pseudocode of The Proposed JPAO-GRU.....	94
Figure 4.1: Encryption Time Comparison.....	110
Figure 4.2: Key Generations Time Comparison.....	112
Figure 4.3: Storage Analysis Comparison.....	114
Figure 4.4: Key Generations Time Comparison.....	116
Figure 4.5: Graphical Analysis Based on Performance Metrics	117

Figure 4.6: Comparative Validation	118
Figure 4.7: Performance Evaluation Regarding (a) Encryption Time and Decryption Time (b) Memory Usage on Encryption and Memory Usage on Decryption	119
Figure 4.8: Performance Assessment for IDS	121
Figure 4.9: Performance Assessment for IDS	127
Figure 4.10: Performance Validation for Data Security	127
Figure 4.11: Security Level	129
Figure 4.12: Performance Analysis for OD Identification	130
Figure 4.13: Performance Assessment for Load Balancing Regarding (a) Response Time (b) Latency (c) Load Balancing.....	131

ABBREVIATIONS

ABE	:	Attribute-Based Encryption
AES	:	Advanced Encryption Standard
AKS	:	Azure Kubernetes Service
AWS	:	Amazon Web Services
CaaS	:	Containers as a Service
CI/CD	:	Continuous Integration and Continuous Deployment
CP-ABE	:	Cipher Text-Policy Attribute-Based Encryption
CPJTABE- THKSCC	:	Ciphertext-Policy Jacobi Theta Attribute-Based Encryption- Twisted Hessian-Kummer Surface Curve Cryptography
CP-WABE	:	Cipher Text-policy Weighted Attribute-based Encryption
CSP	:	Cloud Storage Providers
CT	:	Cipher Text
DBaaS	:	Database as a Service
DSA	:	Data Encryption Standard
ECC	:	Elliptic Curve Cryptography
EKS	:	Elastic Kubernetes Service
ELP-DSA	:	Exponential Linear Probing Based Digital Signature Algorithm
FaaS	:	Function as a Service
FCSDDBSCAN	:	Fleiss Cohen's Based Silhouette Dunn Density-Based Spatial Clustering of Applications With Noise
G	:	Generate

GKE	:	Google Kubernetes Engine
GPk	:	Global Public key
H	:	Hash function
HABE	:	Hierarchical Attribute-based Encryption
HIDE	:	Hierarchical Identity-based Encryption
HTT-Fuzzy	:	Hyperbolic Tangent Trapezoidal-Fuzzy
IaaS	:	Infrastructure as a Service
IBE	:	Identity-based Encryption
IEC	:	The International Electro-Technical Commission
IL-HH-CP-ABE	:	Integrated Lightweight and Hidden Hierarchical Cipher Text Policy Attribute-Based Encryption
IOT	:	Internet of Things
IoT-NM	:	IoT Network Monitoring
IPaaS	:	Integration Platform as a Service
ISO	:	International Standards Organization
JPAO-GRU	:	Jeffreys Prior Aranda Ordaz Gated Recurrent Unit
K	:	Key
KDC	:	Key Distribution Centre
KP-ABE	:	Key-policy Attribute-based Encryption
LWC	:	Lightweight Cryptography
LWC-ABE	:	Lightweight Cryptography Cipher Text-Policy Attribute-Based

Encryption

M	:	Message
MAC	:	Message Authentication Codes
MFA	:	Multi-factor Authentication
MK	:	Master Secret Key
ML	:	Machine Learning
MPK	:	Master Public Key
MSK	:	Master Secret Key
NaaS	:	Network as a Service
PaaS	:	Platform as a Service
PK	:	Public key
PKI	:	Public Key Infrastructure
PRK	:	Privet Key
QASOA	:	Quokka Airy Swarm Optimization Algorithm
QGE-HMAC	:	Quantum Gibbs Entropy Hash-based Message Authentication Code
REQ	:	Registration Request Message
RSA	:	Rivest-Shamir-Adleman
SaaS	:	Software as a Service
SDT	:	Secure Data Transmission
SD-WAN	:	Software-defined Wide-area Networks
SK	:	Secret Key

StaaS	:	Storage as a Service
TA	:	Trusted Authority
Tas	:	Trusted Authority
VPNs	:	Virtual Private Networks



LIST OF SYMBOLS

$\oplus$	:	XOR
$*$	:	Multiply
σ	:	Cryptographic Signature
REQ_A	:	Registration Request Message
λ	:	Random Value
Δ	:	Combined Data
Pub_{A1}	:	Public Key Node A
Pri_{A1}	:	Private Key Node A
τ	:	MAC tag generated
τ'	:	Recalculated MAC tag
ω	:	Inverse Calculation
v	:	Signature Verification Value
S	:	Attribute Set
S^*	:	Specific Attribute Set
$\tilde{S}$	:	Sub-attribute Set
A	:	Access Structure
A^*	:	Target Access Structure
α	:	Key Coefficient
β	:	Parameter Coefficient
$\prod$	:	Multiplicative Product
$\forall$	:	Universal Quantifier

1. INTRODUCTION

1.1 BACKGROUND OF THE STUDY

A novel era marked by heterogeneous and distributed networks has been ushered in by the IoT, with IoT applications spanning several industries, such as building automation, Healthcare (HC), manufacturing, transportation, and environmental monitoring [1]–[3]. However, the collection and storage of huge data produced by IoT devices presents challenges. Cloud-based architectures have gained popularity as a solution to these problems, which ensures resource sharing and security. However, protecting the overall IoT implementation architecture from potential intrusions that can obstruct services and jeopardize data privacy, trustworthiness, or secrecy still needs to be a top priority.

Encryption is the cornerstone of preserving user data secrecy in the cloud context. Yet, IoT applications are still plagued by security issues, prompting a review of current security procedures and the investigation of cutting-edge methods. The main problem is caused by the resource limitations enacted by IoT devices, which frequently have constrained computing and power potentials. When used on these resource-constrained devices, conventional cryptographic primitives created for more robust computing environments sometimes prove ineffective [4]. In IoT ecosystems, authentication is the cornerstone of security, especially in cloud-assisted IoT use cases where data is regularly sent and processed remotely. Digital certificates or API keys are frequently used to implement device authentication, a common approach. The danger landscape must change to combat new threats effectively, necessitating more reliable authentication techniques. Multi-factor authentication (MFA) is a potent strategy to strengthen authentication security for cloud-assisted IoT. Before providing access, MFA requests various forms of identity from users and devices. Users might be required to enter a password and a fingerprint scan, for instance, or a one-time authentication code they received by SMS. MFA dramatically reduces the danger of unauthorized access by requiring attackers to compromise many factors to acquire access. Another essential component of IoT security is public critical infrastructure (PKI), which uses asymmetric encryption in which each device has a Public Key (PK) for encryption and a Private Key (PRK) for decryption [5]. Data in transit is protected by this method, which guarantees secure connectivity between devices and the cloud. Lightweight

Cryptography (LWC) has emerged as a potential field to address the issues caused by the limits of conventional cryptographic techniques on resource-constrained IoT devices [6]. LWC focuses on creating cryptographic primitives that are especially suited for devices with limited resources. LWC adopts established cryptographic elements and modifies them slightly rather than generating whole new cryptographic techniques. It includes the block and stream ciphers, hash function, and Message Authentication Codes (MAC) subsets of the four basic categories of cryptographic primitives. While stream ciphers encrypt data bit by bit, data is encrypted by the block ciphers in fixed-size blocks using a cryptographic key and algorithm. While MACs confirm the message's origin and integrity, hash functions convert data of any size into fixed-size numbers. These cryptographic building blocks protect IoT data in sparsely populated environments [7]. Modern LWC heavily emphasizes novel and unorthodox architectures to design compact, practical, secure ciphers suitable for resource-constrained environments. Despite the development of numerous cryptographic primitives, determining their cryptographic security is still a work in progress [8]–[9]. The creation of block ciphers has attracted much interest in LWC. Researchers have developed Attribute-Based Encryption (ABE) techniques to improve security standards. ABE reduces users' computing costs with outsourced decryption, attribute revocation provides prompt adjustments to access privileges, and policy updates consider evolving AC policies [10]. White box traceability is provided via LWC-ABE schemes, satisfying upcoming IoT requirements. Compact, low-power devices like RFID tags and contactless smart cards are often used in Internet of Things (IoT) applications, and they can benefit from LWC. In IoT applications, particularly those that use the CT authentication architecture, ensuring critical security is essential. Existing systems struggle with issues, including the discrete logarithm issue for Elliptic Curves (ECs), security hashing algorithms, and security breaches. In addition to improving data security, CP-ABE also makes access management easier, mitigating the data breach risk caused by improperly configured permissions or AC lists [11]–[12]. Given the significant amount of data created by IoT devices, efficiency is just as important as security in cloud-assisted IoT. This data may strain the cloud's resources, resulting in performance and latency problems. Alternative methods can minimize resource usage and improve overall efficiency in handling these problems [13].

Exploring different ABE techniques designed for applications with limited resources has been the focus of additional studies [14]. ABE's capacity to reduce overhead during crucial

generation is a noteworthy benefit. The acceptance of ABE solutions is also greatly influenced by design and implementation decisions. While cryptographic specialists have thoroughly studied the most popular and lightweight Elliptic Curve Cryptography (ECC) implementations, conventional ABE techniques have encountered difficulties because of several vulnerabilities.

IoT devices frequently produce huge data. Before the data is sent to the cloud, data compression technologies like Gzip or Brotli can drastically reduce the data size. This helps cloud servers run more efficiently while simultaneously conserving Bandwidth (BW). A concept known as "edge computing" has arisen that moves data storage and computation closer to IoT devices. This method decreases the requirement for constant data transfer to the cloud, diminishing latency and facilitating Real-Time (RT) Decision-Making (DM). It does this by processing data locally on edge devices. The tremendous influx of IoT data may take a lot of work for conventional relational databases to handle. Distributed database solutions can scale horizontally to handle the increasing data load, offering high availability and minimal latency. Examples include Apache Cassandra and Amazon DynamoDB. RT analysis of IoT data streams by Machine Learning (ML) algorithms enables the proactive detection of anomalies or security issues. This strategy allows for prompt action, increasing security and effectiveness.

Three trustworthy encryption approaches, such as SPECK, AES, and SIMON, have also been the subject of extensive research into evaluating their performance [15]. HC signal processing expenses for short-term monitoring and emergency notifications have decreased owing to the expanding adoption of the IoT. Encryption is required as data secrecy becomes essential. However, deploying reliable encryption techniques in IoT contexts is difficult due to the strict restrictions on memory, processing speed, power consumption, and device size [16]. Due to their limited resources, IoT platforms frequently find using conventional data encryption techniques challenging. Compared to the original SIMON block-encryption method and AES, developed a method. In this case, an encryption technique requires a two-dimensional geometry for specific vertices. Mathematical enclosures play a crucial role, indicated as $K_p=K$, where K is a definite mathematical enclosure with a perfect feature p . Tate pairing—mathematical operations or abelian morphisms that frequently span limited or precise locations—have been utilized to create novel processes and processes for a variety

of activities. Pairing-based encryption has many uses in CP-ABE, such as signature techniques, ABE-based essential establishment methods, and privacy-enhancing systems utilizing credentials that are anonymous. Additionally, pairing-centric Identity-Based Encryption (IBE) systems have been developed due to security concerns. These systems enable senders to encrypt messages without requiring prior identification or transmitting the recipient's PK. [17] proposed an IoT-specific CP-WABE scheme. Cryptographers have suggested employing a projective homogeneous coordinate system to decrease the arithmetic operations' inversion cost on elliptical and elliptic category curves, such as point addition and multiplication [18]. Lightweight web-based apps interfacing with data storage and transfer through wireless connectivity vulnerable to hacking typically use Searchable ABE. Within the field of finite order, these functions yield EC points' pairs. [19] introduced a comprehensive and flexible ABE that is based on data access policies. However, this method is susceptible to several attacks, encompassing alteration attacks, attacks on perfect forward secrecy, and attacks on known critical security [20]. These research efforts have led to more effective computations and memory, power, and BW savings, all beneficial in contexts with limited resources. As security requirements continue to change, LWC-ABE surpasses its rivals in terms of competitiveness [20].

1.2 INTRODUCTION TO CLOUD SYSTEM IN IOT ENVIRONMENT

Cloud Computing (CC) is a critical element that significantly boosts the success of the IoT. It provides the capacity to utilize resources provided over the Internet to conduct computing activities. IoT and cloud technology integration have produced a positive feedback loop that advances both industries. This potent combination can alter several sectors and cause substantial gains. An in-depth discussion of the considerable effects of IoT CC is provided in this section, with particular emphasis on its benefits, functionalities, and the critical function that well-designed cloud architecture plays in assuring dependability, security, economy, and performance optimization.

IoT and CC's convergence is a technology catalyst that benefits organizations and consumers. The requirement for data generation and processing has augmented dramatically owing to the extensive usage of IoT devices. CC provides the optimal infrastructure to handle these difficulties effectively. Our interactions with data and devices are changing due to the interaction between IoT and CC, opening up new opportunities [21]–[25].

1.2.1 Benefits and Functions of IoT Cloud Computing

The many advantages and features of IoT CC are revolutionizing how people interact with technology and data [26],[30].

The extensive connectivity possibilities provided by IoT CC allow users to access CC services from various devices, including mobile phones, tablets, and laptops. Although this accessibility is beneficial, it calls for developing strong network access points to handle the increased number of linked devices.

IoT CC is accessible to users and developers as a web service without further authorization or support. The only prerequisite is having internet access. Utilizing cloud resources is made more accessible by their on-demand nature.

The scalability of CC is one of its distinguishing characteristics. Cloud services can be scaled by the users for meeting their unique necessities. This scalability includes increasing storage capacity, changing software options, and compensating for changes in user numbers.

CC involves pooling resources, enabling increased collaboration, and building closer connections between users, empowering enterprises to harness ample computer power and storage resources as needed. This resource pooling makes it easier to allocate and use resources effectively.

Security becomes increasingly important as the IoT environment grows with more devices and automation. Cloud systems use strong authentication and encryption techniques to protect data and communications. This protects IoT data's privacy and security while reducing any potential weaknesses.

IoT CC has a pay-as-you-go business model. Users are charged according to usage, and service providers regularly watch usage statistics. Cost efficiency results because businesses pay only for the resources they utilize. This strategy ties charges to actual consumption, which makes it a financially appealing choice for companies.

1.2.2 Well-Designed Cloud Architecture

A well-planned cloud architecture is pivotal for IoT CC's success. Several important factors must be taken into account to guarantee dependability, security, economy, and performance optimization [31]–[32]:

The creation and deployment of cloud-based IoT applications rely severely on Continuous Integration and Continuous Deployment (CI/CD) pipelines. They allow software updates to be tested, integrated, and deployed automatically, providing a smooth and reliable rollout. APIs and service boundaries should be clearly defined in the cloud architecture. This supports maintainability, scalability, and modularity. Structured services make complex IoT application management more effortless. To ensure security and agility, sandboxed development and testing environments are crucial. These secure environments let programmers test and improve IoT applications without endangering the live system's integrity. In IoT cloud architecture, effective data management is essential. Organizations may gain insights from the vast IoT data created by implementing data pipelines, storage solutions, and data processing frameworks. The cloud architecture must have strong security procedures, including encryption, AC, and threat detection techniques to protect IoT data from cyber-attacks and illegal access. IoT cloud solutions must be built to grow horizontally and handle dynamic workloads. Resource allocation is optimized by load balancing to meet varying demands. Redundancy mechanisms and disaster recovery strategies should be employed to guarantee higher availability and data resilience. Data must be simulated across several data centers to reduce the risk of data loss. In the (Figure 2.1) shows lifecycle and cloud interact model.

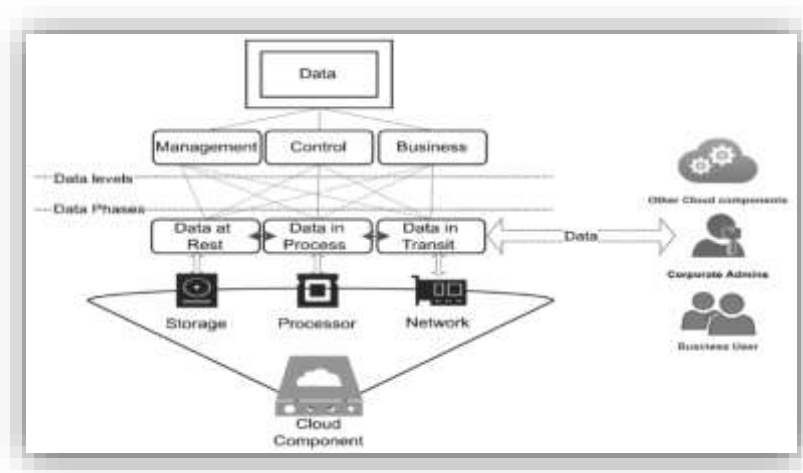


Figure 1.1: Data Lifecycle and Cloud Components Interaction Model.

1.2.3 Types of Cloud Computing

Various forms of CC exist, each suited to particular corporate requirements and applications [33],[34].

1.2.3.1 Infrastructure as a service (IaaS)

IaaS is a fundamental CC concept that provides consumers online access to virtualized computing resources. Users can access critical infrastructure elements like virtual computers, storage, and networking in an IaaS environment. Scalability, which enables users to adjust resource levels up or down in response to demand, is one of the core features of IaaS. Businesses with variable workloads will significantly benefit from this flexibility. Operating systems, customizations, and programs within users' virtualized environments remain under their control. IaaS is economical since it does not require initial capital expenditures for physical hardware. Prominent IaaS providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) render various services and resources, which makes it appropriate for multiple use cases, including web hosting, big data processing, and disaster recovery.

1.2.3.2 Platform as a service (PaaS)

PaaS is a CC model that simplifies application growth and execution. Developers can build, execute, and maintain applications on a platform rendered by PaaS without dealing with the difficulties of managing underlying infrastructure components. Platforms as a service (PaaS)

provide various services, such as hosting for applications, databases, and development frameworks. This paradigm facilitates the coding process using tools and services, which expedites the development of applications. Developers don't need to worry about infrastructure upkeep because PaaS providers oversee patching and security updates. Instead, they can concentrate on building applications. PaaS is especially beneficial for businesses aiming to increase developer productivity, achieve rapid deployment, and cut maintenance costs. Heroku, Google App Engine, and IBM Cloud Foundry are notable PaaS providers.

1.2.3.3 Software as a service (SaaS)

A CC technique termed SaaS permits subscription-centric delivery of software applications over the Internet. Users access SaaS apps through web browsers; hence, local installations or updates are not needed. SaaS is the best option for remote work situations because it can be accessed anywhere with an internet connection. SaaS providers manage application updates to guarantee that customers have access to the most recent features and security updates. Users can pay for SaaS apps monthly or yearly thanks to subscription-based pricing, which makes expense management more effortless. Microsoft 365, Salesforce, Dropbox, and Google Workspace are well-known SaaS examples. Applications for file sharing and storage, CRM, productivity suites, and email and collaboration tools are commonly used SaaS services.

1.2.3.4 Function as a service (FaaS)

The CC approach, FaaS, sometimes serverless computing, entirely abstracts infrastructure administration. Users of FaaS deploy individual programs or pieces of code that run concerning specific occasions or triggers. Event-driven execution, where events like HTTP requests or changes in data trigger functions, is the defining feature of FaaS. To ensure optimum performance and resource utilization, FaaS platforms automatically scale resources in response to incoming demands. FaaS is a cost-effective option for occasional workloads because users are only charged for execution time and resource usage. FaaS is commonly used for web applications, data processing, IoT applications, and microservices architectures. Prominent FaaS platforms include AWS Lambda, Azure Functions, Google Cloud Functions, and IBM Cloud Functions.

These four basic types of CC models and customized variations provide businesses and developers with an extensive range of possibilities for utilizing cloud resources successfully. The sort of CC selected relies on particular needs, goals, and the makeup of the applications or workloads hosted there.

1.2.3.5 Database as a service (DBaaS)

Users can get managed database services through the CC concept called database as a service (DBaaS). Organizations can delegate administrative duties related to database maintenance to cloud service providers using DBaaS. Numerous benefits, such as automatic backups, scalability, high availability, and security measures, are provided by this paradigm. While the cloud provider performs duties like software patching, database maintenance, and data replication, users may concentrate on database building, querying, and data analysis. DBaaS is especially beneficial for businesses wishing to ease the operational load of database management and guarantee data reliability. Amazon RDS, Azure SQL Database, and Google Cloud SQL are notable DBaaS providers.

1.2.3.6 Containers as a service (CaaS)

The CC approach, containers as a service (CaaS), focuses on managing and orchestrating containers. Applications and their dependencies are contained within containers, which are compact, portable, and uniform environments. CaaS platforms provide tools for deploying, scaling, and maintaining containerized applications. A typical element of CaaS products is the open-source container orchestration engine Kubernetes. CaaS makes scaling and application deployment easier, which makes it an excellent fit for microservices architectures. Applications may be deployed in containers with ease and flexibility, and operations teams can effectively manage workloads that have been containerized. Google Kubernetes Engine (GKE), AWS Elastic Kubernetes Service (EKS), and Azure Kubernetes Service (AKS) are three popular CaaS systems.

1.2.3.7 Integration platform as a service (iPaaS)

Platform for Integration as a Service (iPaaS) is a CC platform that focuses on making it easier to integrate applications and data. To enable smooth data flows and process automation, iPaaS platforms render tools and services to connect multiple applications, systems, and data sources. This paradigm is essential for businesses that need effective

integration across various application environments. For well-known applications and services, iPaaS solutions provide pre-built connections and templates, facilitating integration more quickly and widely. Less custom coding is required because users can create, install, and manage integration workflows visually. Data synchronization, transformation, and transfer between on-premises and cloud systems are all streamlined by iPaaS. Dell Boomi, MuleSoft Anypoint Platform, and Microsoft Azure Logic Apps are well-known iPaaS solutions.

1.2.3.8 Network as a service (NaaS)

The CC concept termed NaaS renders network-related services. Virtual Private Networks (VPNs), Software-Defined Wide Area Networks (SD-WAN), load balancing, and security services are just a few of the services included in NaaS offerings. NaaS improves network flexibility, lowers hardware dependence, and simplifies network management. Users may provision and manage network resources through a centralized cloud-based interface, enabling dynamic adjustments to network configurations and scaling in response to demand. Organizations looking to improve network performance, Secure Data Transmission (SDT), and simplify network infrastructure management will find NaaS especially helpful. Leading NaaS providers offer a range of network services specifically designed to satisfy different connectivity needs.

1.2.3.9 Storage as a service (STaaS)

A CC concept termed STaaS provides businesses with scalable and controlled storage options. Block, object, and file storage are just a few of the many storage possibilities included in STaaS solutions. Users can scale up or down capacity to satisfy changing data storage requirements by allocating storage resources as needed. Cloud providers manage the underlying storage infrastructure to ensure data security, availability, and longevity. Organizations with various data storage requirements, from archiving and backup to content delivery and analytics, can benefit from STaaS. Organizations can lower capital expenditures on storage infrastructure and streamline data management by implementing STaaS. Leading STaaS providers include Google Cloud Storage, Azure Blob Storage, and Amazon S3.

To handle certain use cases and requirements, these specialized CC models (IaaS, PaaS, SaaS, and FaaS) expand the capabilities of the core models. These specialist services can be

used by businesses to improve productivity, simplify operations, and increase resource usage in their cloud settings.

1.3 THE ROLE OF CLOUD COMPUTING IN EMPOWERING THE INTERNET OF THINGS

The development of CC has dramatically enhanced the IoT capabilities and potential as a foundational technology. They combine to provide a potent synergy, transforming how data is gathered, processed, and used in our networked world. The CC industry's scalable and reliable infrastructure can assist the massive network of IoT devices. The IoT sensors, devices, and endpoints produce enormous amounts of data. The flexibility of the cloud enables it to scale up smoothly and down to manage the constant inflow of data. This scalability guarantees IoT systems can increase functionality without facing performance or resource problems. The value of IoT is derived from its capacity to produce worthwhile insights from data. CC provides robust data analytics and processing technologies that turn unprocessed IoT data into useful information. Organizations may find trends, patterns, and anomalies in IoT data by utilizing cloud-centric analytics tools and ML algorithms. This intelligence makes RT DM, preventative maintenance, and process optimization possible, which boosts operational effectiveness. A logistical difficulty might arise when managing and monitoring IoT devices spread across large geographic areas. This process is made more accessible by CC's centralized remote management and control capabilities. Organizations may provision, monitor, and control IoT devices from a single interface through cloud platforms. The stability and security of IoT ecosystems are guaranteed by this centralized control, which facilitates device updates, security enforcement, and configuration changes. IoT data is stored in scalable and secure cloud storage options. Even if a device malfunctions, cloud data storage ensures accessibility and longevity. Thanks to cloud-based storage, IoT-generated data may be easily retrieved and accessed from any location with a connection to the internet. RT monitoring, investigation, and information exchange depend on this accessibility. IoT implementations are highly concerned with security. The security methods utilized by cloud providers, namely authentication, encryption, and AC, are powerful. IoT ecosystems may efficiently reduce cybercrime threats and preserve sensitive data by utilizing these cloud security capabilities. The ability to continuously monitor security and respond to incidents through cloud platforms improves threat detection and resilience. IoT projects

can use the cost-effective concept of CC. By paying for cloud services on a pay-as-you-go basis, businesses can avoid making substantial upfront capital expenditures on infrastructure. Organizations can spend resources wisely and lower the financial risks involved with scaling IoT initiatives because of this cost predictability, which fits well with the changing needs of IoT projects. IoT devices frequently need regular firmware and software updates to improve operation and fix security flaws. CC makes disseminating updates to an extensive range of devices more accessible. Organizations can use the cloud to remotely install patches and updates, ensuring that IoT devices are protected against new risks and always up to date. Cloud service providers run data centers worldwide, extending the reach and accessibility of IoT solutions. Because of their proximity, IoT devices and cloud services may interact quickly and in RT. This is especially important for applications, such as driverless cars and industrial automation, in which split-second choices can significantly impact.

1.4 HYBRID ENCRYPTION METHOD FOR DATA AUTHENTICATION

To increase security, a hybrid approach combines encryption algorithms, each with specific strengths and limitations. Hybrid approaches strive to render higher levels of security by combining the advantages of many encryption technologies and minimizing each one's particular shortcomings. Other hybrid encryption methods have been created, including AES-ECC, Fibonacci series-XOR bitwise-PN sequence, IDEA-RSA, and DES-RSA [35]–[40]. In a hybrid encryption scenario, plaintext data is initially encrypted with one encryption method and a Secret Key (SK). Another encryption method is then used to encode this SK. The original plaintext is then once more encrypted using the SK that was previously encrypted. This multi-layered strategy improves security and can be used with multimedia data, comprising text documents, photos, audio files, and videos. For instance, the base64 encoding of the data is the 1st step in the AES-ECC hybrid technique. The initial SK is produced randomly using the AES. ECC PK encryption is then utilized for encrypting this key. The plaintext data encrypts the encrypted AES key, creating CT. The AES-encrypted content is then subjected to an additional layer of encryption utilizing ECC PK encryption to increase security further. The input message is split into equal-sized blocks for the Fibonacci-XOR logic-PN Sequence hybrid approach. Each block contains the same number of characters. However, more information on this strategy is still pending. Hybrid encryption algorithms provide a reliable and adaptable way to protect data across numerous applications

and data types. They offer improved security and defense against potential weaknesses by integrating the capabilities of various encryption methods. These hybrid strategies help ensure that data is private and impenetrable while dealing with sensitive information and multimedia material.

Data integrity and security are crucial in a society that is becoming more linked. Keeping data private, unaltered, and authentic while sending sensitive financial information, securing personal communications, or protecting vital company data is essential. The Hybrid Encryption Method for Data Authentication is a potent method that overcomes these issues. This approach contains data authentication procedures and integrates the advantages of symmetric and asymmetric encryption techniques to offer complete data protection.

At its core, hybrid encryption is a two-step process that ensures SDT:

- i. Symmetric Encryption

In the first stage, the data is encrypted utilizing a symmetric encryption approach. This approach uses the symmetric key for decryption and encryption. Effective symmetric encryption techniques like AES can benefit bulk data encryption.

- ii. Asymmetric Encryption

Asymmetric encryption safely exchanges symmetric keys between the sender and the recipient. PK and PRKs are utilized in asymmetric encryption. While the PRK is kept private, the PK is publicly shared. The sender encrypts the symmetric key by utilizing the recipient's PK, which ensures that only the recipient with access to the linked PRK may decrypt it.

1.4.1 The Key Components

To implement the Hybrid Encryption Method for Data Authentication effectively, several key components and processes must be in place:

The choice of a secure and efficient symmetric encryption algorithm is critical. AES, a widely adopted symmetric encryption standard, offers robust security and performance.

Both the sender and the recipient generate the pairs of asymmetric keys: PKs and PRKs. These keys must be securely managed, with the PRKs kept secret.

A random symmetric encryption key, often called a session key, is generated for each data transmission. This key is utilized exclusively to encrypt and decrypt the actual data.

Data authentication mechanisms, including cryptographic hashing and digital signatures, are integrated into the method for the verification of data authenticity and integrity.

1.4.2 The Encryption and Decryption Process

The Hybrid Encryption Method involves several crucial steps in encrypting and decrypting data:

a. Encryption (Sender's Perspective)

Create a random session key for encrypting the data itself.

Encrypt the data with the session key and the symmetric encryption algorithm of your choice (like AES).

Use the PK of the receiver for encrypting the session key, ensuring that only the recipient can decrypt it.

To generate a digital signature and perform data authentication, create a cryptographic hash of the encrypted data utilizing the PRK of the sender.

b.Data Transmission

Transmit the encrypted data along with the encrypted session key and, optionally, the digital signature to the recipient through a secure communication channel.

1.4.3 Decryption (Recipient's Perspective)

- i. Using the PRK of the recipient, which is kept private, the session key is decrypted.
- ii. To decode the encrypted data and expose the original plaintext message, use the session key that has been decrypted.

- iii. If a digital signature is present, use the PK of the sender for decrypting the attached digital signature and disclosing the original Hash Value (HV).
- iv. Recalculated the HV from the received data content.
- v. To check for data integrity and authenticity, compare the received HV with the decrypted HV from the digital signature.

1.4.4 Advantages of the Hybrid Encryption Method

The Hybrid Encryption Method offers a range of advantages:

- i. Data confidentiality and secure key exchange are ensured by combining symmetric and asymmetric encryption. Without the PRK, an attacker cannot decrypt the data even if they can access the encrypted PKs.
- ii. Asymmetric encryption is only used for the secure exchange of the session key, whereas symmetric encryption is more effective for bulk data encryption. Performance is optimized by this method.
- iii. Data authentication techniques ensure data integrity and origin. • The approach can be enhanced to include access policies and attribute-based AC, permitting Data Owners (DOs) to regulate who can access their encrypted data. The recipient confirms that the data is not altered during the transfer.

The Hybrid Encryption Method for Data Authentication has many practical applications. Especially in secure messaging apps, it plays a crucial part in secure communication by guaranteeing that messages maintain confidentiality, integrity, and authenticity during transmission. Online shopping platforms use hybrid encryption to protect clients' financial data in the world of e-commerce, ensuring the security of credit card numbers and personal information during transactions. Hybrid encryption protects files that users upload to cloud storage services, ensuring that even if the cloud provider keeps the data, only authorized users with decryption keys can access it. Organizations also use hybrid encryption to exchange sensitive data securely across offices or with outside partners, where data integrity and authenticity are crucial. Finally, hybrid encryption frequently works with digital

signatures to verify the validity of signed documents and communications while maintaining the integrity of the content.

1.5 CIPHERTEXT-POLICY ATTRIBUTE IN CLOUD SYSTEM USING IOT

A cryptographic system called CP-ABE, which bases its encryption algorithm on a CT-policy attribute, enables precise AC to encrypt data. Only users whose attributes fit the policy could decrypt and access the data, allowing DOs to set ACs based on attributes [45]–[51].

A CP-ABE system includes four fundamental algorithms: setup, encryption, key generation (keygen), and decryption [52]–[60].

Let's explore each of these algorithms [61]–[64] comprehensively:

a. Setup Algorithm:

The setup algorithm is accountable for initializing the CP-ABE system. It generates the necessary parameters and keys for encryption and decryption.

b. Key components include:

- i. A Master Secret Key (MSK) is an SK held by the authority or DO who manages AC policies. It is used to generate user-specific PRKs.
- ii. A Master Public Key (MPK): This PK can be freely distributed and utilized for encryption. It contains system parameters and policies.
- iii. The setup algorithm ensures that the system parameters are securely generated and that the MPK and MSK are created without exposing sensitive information.

c. Encryption Algorithm

- i. A DO uses the encryption algorithm for encrypting data so that users with particular attributes can decrypt it.
- ii. The technique uses plaintext messages and a set of ACs defined in terms of attributes to encrypt data.

- iii. It creates a ciphertext that contains the message that has been encrypted as well as an access policy that outlines the characteristics needed for decryption. Thanks to the encryption procedure, only users whose attributes fit the access policy could successfully decrypt the ciphertext.

d. Key Generation (Keygen) Algorithm

- i. Users utilize the key generation algorithm to create their PRKs centered on the traits and access policies established by the DO.
- ii. Using their PRK, users can decrypt ciphertexts that fulfill specific criteria.
- iii. A user must provide the key creation algorithm with their properties for generating a PRK. The algorithm generates the PRK using the MSK and the user's properties.
- iv. To decrypt ciphertexts that comply with the attributes of the user, the produced PRK must be particular to those qualities.

e. Decryption Algorithm

Authorized users could access the original plain text data by decrypting ciphertexts using the decryption method.

- i. A user must give both the ciphertext and their PRK, which was produced based on their attributes, to decipher a ciphertext.
- ii. The decryption algorithm verifies that the properties of the user comply with the access policy stated in the ciphertext.
- iii. The PRK is utilized by the algorithm for decrypting the ciphertext and exposing the plaintext message if the characteristics of the user meet the policy.

Decryption fails, and the user cannot access the data if the attributes do not match. Creating a Cyber text Policy Attribute in a Cloud System employing IoT is essential to provide data security, AC, and privacy in a linked world. This section briefly describes the application of Cybertext Policy Attributes in a Cloud System that uses IoT devices.

Within an IoT ecosystem, Cyber Text-Policy Attributes are connected to individuals, objects, or data components. These attributes specify particular AC policies according to various factors, such as user roles, device properties, location, and more. They are crucial in defining who has access to information and under what circumstances. IoT is a network of interlinked gadgets and sensors that produce enormous volumes of data. Cyber Text-Policy Attributes are included to guarantee that access to this material is limited and managed by specified policies. This is essential to ensure data privacy, security, and compliance.

f. Key Steps in the Implementation

- i. Establish the characteristics and rules that will control data access in the IoT ecosystem. For example, attributes might comprise device types, data sensitivity levels, user roles (admin, operator, viewer), and location-based attributes.
- ii. Assign users, gadgets, or data components the appropriate properties. For instance, attributes identifying the level of confidentiality and the particular medical facility to which the data belongs may be applied to a sensor collecting sensitive patient health data.
- iii. Include policy enforcement methods in the Cloud System. When an access request comes in, these mechanisms analyze it to see if the user or device making the request satisfies the requirements outlined in the Cyber Text-Policy Attributes.
- iv. The examination of Cyber Text-Policy Attributes determines whether access to data is permitted.
- v. Implement reliable logging and auditing mechanisms to track data access and usage. For instance, a medical IoT device transmitting patient data might only permit access to authorized HC professionals inside a particular facility. Accountability is ensured, and a record of who accessed what data and when is provided. These are essential for compliance and investigation of security incidents.
- vi. Cybertext-policy attributes can be dynamically changed to account for shifting conditions and scenarios. For example, if a user's role changes, the user's attribute and related access rights can be modified without disrupting the system as a whole.

- vii. Ensure that the Cloud System and IoT devices are seamlessly integrated. Devices must be able to identify and enforce access regulations based on the given Cyber text Policy Attributes.
- viii. To prevent data interception or eavesdropping, create secure communication routes betwixt IoT devices and the cloud system. Data can be encrypted during transmission using security protocols like SSL/TLS.

1.6 MOTIVATION FOR THE RESEARCH

The dynamic and quickly changing IoT landscape is an intrinsic motivation for our research. One of the highly substantial technological revolutions of time is the IoT, which is distinguished by its exponential growth and constantly expanding network of linked gadgets. However, this deep upheaval comes with several urgent difficulties and opportunities, which drive the motivation for this research. By 2020, there are expected to be tens of billions of IoT devices, which highlights the urgent need for adequate security of data and privacy securities. Owing to the vast amounts of data these devices collect and communicate, including everything from private information to industrial telemetry, there is a significant threat of data breaches as well as privacy violations. By creating encryption and authentication procedures that protect data while it is in transit between IoT devices and cloud-based services, this research aims to solve this essential worry. The resource limitations that many IoT devices have are a significant motivator. These devices' processing power, memory, and energy efficiency, which are frequently found embedded in industrial machines or at the edge of networks, are constrained. Conventional encryption techniques created for more competent computer environments are impractical for these resource-constrained IoT endpoints. For these devices to securely engage in the IoT ecosystem, lightweight yet robust security solutions must be developed expressly. This research is also motivated by the desire to render fine-grained AC in IoT contexts. Managing access rights gets incredibly complex when many linked individuals and devices exist. A possible approach to dealing with this complexity is ABE, particularly CP-ABE. However, the computational burden associated with CP-ABE presents difficulties, especially for portable IoT devices. The goal is to enable secure and effective communication while protecting data privacy and integrity by bridging the gap between the IoT's security needs and these devices' resource limitations. In essence, a convergence of variables, comprising the enormous

expansion of IoT, the resource limitations of IoT devices, the necessity to safeguard sensitive data, and the requirement for accurate AC, is what drives this research. The project intends to enable the smooth integration of IoT into various sectors, reaching its full potential while assuring data security in an increasingly linked society by tackling these difficulties.

1.7 AIMS AND OBJECTIVES

This research aims to enhance data security and accessibility control in the cloud-assisted IoT context by developing a Hybrid Encryption method based on an integrated lightweight and hidden hierarchy CP-ABE. The primary focus is ensuring data confidentiality, authentication, and integrity in transit between IoT edge devices and back-end systems while minimizing computational costs and communication overhead.

The research aims to accomplish the following objectives.

- i. To establish a secure communication framework that ensures non-repudiation, confidentiality, and data integrity during transmission between IoT edge devices and cloud platforms utilizing QGE-HMAC-enabled encryption key protection and ELP-DSA-based signature validation.
- ii. To employ a hybrid encryption approach that incorporates asymmetric and symmetric cryptographic methods, particularly incorporating CPJTABE and THKSCC, to reduce computational overhead and increase encryption-decryption efficiency.
- iii. To introduce a lightweight version of CP-ABE eliminating the need for bilinear pairing and modular exponentiation, enhancing it for low-power IoT devices while supporting expressive AC policies and hierarchical encryption.
- iv. To propose a Gini Coffman Kahn Directed Acyclic Graph (GCK-DAG) structure for adaptive graph updates and network routing, which reflects the dynamic nature of IoT communication flows.
- v. To design a JPAO-GRU-based Intrusion Detection System (IDS) able to detect multiple types of anomalies in cloud-IoT communication using temporal learning with reduced overfitting and enhanced training efficiency.

- vi. To apply a novel clustering mechanism (FCSDBSCAN) for extending network lifetime, reducing energy consumption, and supporting low-latency clustering under varying network densities.
- vii. To evaluate the proposed approaches' performance through metrics, encompassing encryption/decryption time, signature generation and verification time, MAC creation latency, memory usage, energy efficiency, throughput, and overall security level.
- viii. To demonstrate the effectiveness of HTT-Fuzzy for outdated device identification and QASOA for intelligent load balancing across distributed IoT nodes, which ensures timely updates and system responsiveness.
- ix. By attaining these goals, the research intends to contribute to the development of secure and efficient data authentication security mechanisms in cloud-assisted IoT, addressing the unique difficulties posed by resource-constrained IoT devices.

1.8 SIGNIFICANCE OF THE STUDY

The study is crucial in the developing landscape of the IoT in which the propagation of linked devices across diverse industries has led to a critical need for secure, efficient, and scalable data transmission and monitoring solutions. Ensuring data confidentiality, integrity, authentication, and RT trust becomes paramount with billions of edge devices continuously generating and transmitting sensitive information in sectors, including smart cities, HC, industrial automation, and energy. This research responds to that pressing need by proposing a hybrid encryption and monitoring framework that addresses the security of data in transit and the dynamic behavior of IoT devices in the cloud environment.

This study's first significant contribution lies in evolving a hybrid encryption system—built on CP-ABE and improved through lightweight optimization and hidden hierarchical policy enforcement. This novel adaptation supports fine-grained, attribute-driven AC when maintaining lower computational overhead, which makes it ideal for resource-constrained IoT devices. The proposed model makes encryption feasible even on low-power sensors and actuators operating at the network's edge by eliminating resource-intensive cryptographic operations such as bilinear pairings. This enables secure IoT deployment in remote, harsh, and resource-limited environments, extending the practical scope of intelligent applications.

Secondly, the research incorporates reliable cloud-assisted IoT network monitoring by integrating intelligent anomaly detection and lightweight data authentication mechanisms. Using JPAO-GRU for intrusion detection, HTT-Fuzzy for outdated device recognition, and QASOA for load balancing ensures that security is maintained at the data level and the behavioral and operational levels of IoT ecosystems. These modules collectively ensure early threat detection, efficient load distribution, and adaptive network health monitoring, reinforcing system trustworthiness without overwhelming device potential.

Moreover, the security chain is further strengthened by the emphasis on encryption key integrity and authentication through mechanisms like QGE-HMAC and ELP-DSA, ensuring that encryption keys are not compromised during transmission and that the data's integrity and origin can always be confirmed. These features are especially critical in multi-tenant cloud environments in which key spoofing and forged data packets can lead to severe breaches.

Lastly, this study's integrated approach to lightweight security, key management, and anomaly detection is a foundation for future IoT systems that demand scalable, robust, and low-power security models. It assists regulatory compliance, enhances user trust, and paves the way for secure, efficient, and intelligent IoT infrastructures capable of RT autonomous DM.

In essence, IoT ecosystems are empowered by this research to operate at scale securely, adapt to changing network conditions, and ensure data privacy and reliability, thereby contributing to interconnected systems' sustainable and resilient future.

1.9 THESIS ORGANIZATION

The thesis is organized into several chapters to comprehensively address the research on authentication security and CT-policy attributes in cloud-assisted IoT. Chapter 1 introduces the research, providing the background and motivation for the study. It also outlines the research's aim, objectives, and importance, setting the stage for further chapters. Chapter 2 delves into a comprehensive literature review, exploring existing work and various algorithms employed in cloud-assisted IoT, such as clustering-based, ML-based, and scheduling methods. This chapter summarizes prior research in preparation for the proposed methodology. Chapter 3 outlines the research methodology, detailing the proposed system

model, which integrates lightweight authenticated encryption and CP-ABE for data authentication and security. This chapter serves as the core of the thesis, describing the techniques used and their implementation. Chapter 4 renders the results of the proposed methodology, including a detailed evaluation using performance metrics like Communication Overhead and Computational Overhead. This chapter quantifies the effectiveness of the methodology. Chapter 5 concludes the thesis, summarizing the key contributions and highlighting limitations. It also provides insights into future research directions, ensuring a well-rounded conclusion and a potential roadmap for further studies in this field.



2. LITERATURE REVIEW

The article by [65] states the ideology of implementing CP-ABE for cloud storage to secure data privacy and authentication in AI-facilitated IoT systems [65]. There is a requirement to initiate security methods in cloud storage for IoT systems and applications. Individuals and organizations have a feature for storing the data in cloud servers called data centers. However, the drawback in the cloud storage systems is that to render proper security to the access policy of the data; even this is required to be stored in the cloud storage. Different encryption methods were developed in the market to offer security to the data access policy. Among various encryption methods, CP-ABE was selected in the paper to evaluate its performance. The paper focuses on the CP-ABE method to analyze and assess the performance of data authentication in cloud storage, which is assisted by the IoT environment. The CP-ABE method is an advanced technique with high-quality features that protect data and provide authentication [65]. The proposed systems are examined and contrasted with the CP-ABE for the expressive policies. The AC functions were tested and studied in the research paper, and work was proposed to implement the hybrid encryption method on the IoT for better results.

The thoughts of [66]. They expressed their approach to implementing data encryption techniques in CC and IoT environments. The IoT is an advanced version of a computer system that automatically works without any human support and intervention [66]. It is a network interlinked with different applications and the computer system. It consists of unique IP addresses and codes for the devices to share information from one device to another and from one system to another. With this unique code or IP address, the IoT can automatically share or receive information from other devices without human help. CC is the latest version to save the data and access the information anytime, anywhere. All the data generated in vast amounts by IoT applications is stored in the cloud with the help of a data processing technique in CC. A hybrid encryption method is required to secure the vast amount stored in the cloud. As various encryption methods are available in the market for securing data and providing authentication, they are not supported in IoT devices as they are small and cannot handle complex algorithms. The paper focuses on implementing lightweight encryption techniques to resolve security issues by providing a secure network to communicate and transmit data in IoT devices [66]. The paper aims to analyze the

performance of the lightweight encryption method in resolving security issues and providing data authentication. A survey has been conducted to investigate the techniques, features, variants, and issues used in CC and the IoT.

2.1 CLUSTERING-BASED METHOD

In the IoT, huge amounts of heterogeneous data are generated from various constrained applications and sensors based on the resources. In CC, the data generated from the IoT is stored in servers. With the capabilities of both techniques, IoT-Cloud convergence is developed with a mutual relation and capabilities to share the benefits mutually [67]. In a framework, cluster-based methods can be utilized like all the IoT devices as a group, which can convert them into several clusters. Each cluster is used to share the information by converting them into a head. Each IoT device utilized as a head in the cluster could send the aggregated information to the cloud systems through the gateway. The information sent to the cloud systems is used to continue the process. Various clustering methods have been developed and are available for conventional Wireless Sensor Networks (WSNs) [68]. Now, all the clustering methods are unsuitable for implementation in WSNs.

Specifically, some characteristics of the IoT must be selected before implementing any cluster methods for WSNs. One of the best features of the IoT that can be widely capitalized on with the adoption of the clustering method is it can collaborate with heterogeneous IoT devices among them devices [69]. Knowledge of the network is widely used to provide information at cloud centers. The information related to the relations of the devices is beneficial to the IoT gateway from the cloud centers. The paper evaluates and proposes assistance with the cloud system and an interacting clustering scheme for heterogeneous IoT networks. In the proposed approach, the details of residual energy and the closeness factor of the device are jointly utilized to help select the cluster heads effectively [70]. The proposed clustering method has a positive impact on prolonging the lifetime of the IoT network.

2.2 ML-BASED METHOD

ML is an emerging technique that is a perfect solution for attacks identified on the IoT. The problem of different attacks in the IoT applications provides a solution with IDSs in the ML-

based method [71]. This IDS's features are widely helpful for maintaining security in IoT networks. The features of IDSs are testing and running the system in the IoT networks to analyze any threats and attacks. A process must be followed to collect data from the IoT gateway and store the data in the cloud systems. The collected data requires scanning and cleaning initially, and the model must be built in the same way. All the activities must be done within the time, or the undetected risks will slip from the scanning and evaluation process [72]. The ML-centric IDS generally detects the anomalies in the traffic that are known or newly added attacks.

ML will be adopted by analyzing the requirements and computing resources in the large sector for an operational platform [73]. In this case, the ML technique is integrated into the embedded systems for the operation in the enlarged sector of the IoT networks with cloud assistance. The ML models are adopted in the cloud-based architecture as they easily fit in IoT devices for security purposes in the configurations for operation. The two layers are combined to interact with this framework's machine-learning technique [74]. The cloud layer is utilized to carry out heavy-weight tasks with the help of IoT devices, which helps to reduce the maintenance of workload in IDSs

2.3 SCHEDULING METHOD

The IoT is a developing technology that has revolutionized organizations and individuals to work consciously and harmoniously. With the introduction of the IoT, various sectors adopted this technology to manage their work quickly and automatically without human interaction [75]. The data produced from the IoT applications must be handled by CC. With the loads of data generated from the IoT applications, it is difficult for CC services to manage the data. Being a traditional CC server, the demands of the IoT could not be met, and difficulties were faced. To meet the expectations of the IoT, the CC servers came up with a scheduling method and timely allocation of the IoT requests. Many scheduling algorithms were proposed for CC. Still, these vary based on the heterogeneous servers, and the dynamic approach for scheduling depends on the priority requests from the IoT devices to the CC [76]. CC can accept requests from the IoT as a priority and provide quality service.

Various scheduling algorithms are available for CC. Among them, static scheduling and dynamic scheduling algorithms are mainly utilized. Static scheduling algorithms are adopted

by small and medium organizations in CC [77]. In the same way, dynamic scheduling algorithms are adopted by following three factors when accepting data requests from IoT devices. First come, first service is the main factor that aids in maintaining the data quality and CC service. In the same way, large-scale organizations in CC adopt dynamic scheduling algorithms. The requests from the IoT are analyzed, and every user request is accepted by CC based on the request time and data. The static scheduling algorithm follows three factors in receiving the data requests from the IoT devices [78]. First come, first service is the main factor that helps maintain the quality of the data and CC service.

2.4 SUMMARY

This research is conducted by selecting a topic: hybrid encryption methods. Based on the hybrid encryption method adopted in the cloud systems with the assistance of the IoT devices and applications [79]. Two hybrid encryption methods were selected with in-depth analysis: hidden CP-ABE and integrated lightweight-centric authentication encryption. These two hybrid encryption methods are utilized to protect the data and provide authentication stored in the cloud servers and generated in IoT applications. For securing the data from various attacks, the two hybrid encryption methods were selected for implementation in the cloud systems with the assistance of the IoT environment. In the background of the study, detailed information related to the security issues faced by cloud systems and IoT applications is mentioned. Cloud systems' difficulties are maintaining the heterogeneous data of IoT applications securely. The benefit of implementing hybrid encryption methods in cloud systems is that they can be done with the assistance of the IoT environment [80]. This research discusses detailed information on cloud systems in the IoT environment and hybrid encryption for data authentication with various examples. The requirements for maintaining the data authentication are mentioned by analyzing various literature papers see Table 2.1.

Ciphertext-policy attributes in CC with the IoT environment are also discussed in the paper and compared with other hybrid encryption methods. The motivation of the study initiation and interest in selecting the topic is discussed [81]. The factors and criteria that are calculated for choosing the topic are mentioned in detail to mitigate the security issues in the cloud systems' data with the assistance of the IoT environment. The aim and objectives of the topic are explained in the paper. The literature review discusses the existing work by analyzing some research papers on hybrid encryption methods. Some other techniques were discussed,

including implementing cloud systems with the assistance of the IoT environment. The techniques discussed are scheduling, clustering, and ML-based methods [82]. The different algorithm implementations in cloud systems, with the assistance of IoT applications, are evaluated and analyzed for performance.

Table 2.1: Comparative Table for Integrated Lightweight and Hidden Hierarchy Cipher Text-policy Attribute in Cloud-Assisted IoT.

Ref	Title	method	Finding	Research Gap
[83]	An Effectual PK Secure Scheme for Cloud and IoT Security	Cloud and IoT Security (Knapsack method to encrypt our ENPKES)	It demonstrates a robust assessment of efficiency by altering the size of the N-bit modulation up to 10K bits for the key generation, data encryption, and decryption on regular RSA, Enhanced and Secured RSA Key Generation Scheme (ESRKGS), and ENPKES. Overall, 89% of conventional RSA and 27% of ESRKGS are consumed by ENPKES.	Public essential secure techniques suffer a high computational cost; thus, these are unsuitable for the cloud or the IoT as they are too energy-intensive and expensive. This necessitated the proposition of elastic, preemptive cloud protection against IoT infect-replace strategies and fully monolithic similar-dynamics security methodologies PKC scheme that is expected to encrypt essentials for secure key exchange externally.
[84]	Design and Implementation of Hybrid Encryption for Security of IOT Data	AES	It utilized the designing tool in the Xilinx ISE-Design 14.5 software to confirm the outcome after implementing the recommended layout on the Xilinx SPARTAN-6—the transmitter and the recipient process the RSA decryption and encryption algorithms on MATLAB.	A different sector of active study is improving the efficiency of AES, particularly in contexts with limited resources like IoT devices. Bitwise Reverse Transposition is one technique that has shown promise in increasing the speed at which encryption and decryption occur.

Table 2.1: Comparative Table for Integrated Lightweight and Hidden Hierarchy Cipher Text-policy Attribute in Cloud-Assisted IoT “Table Continued”.

[85]	An enhanced attribute-centric encryption approach for data security in CC	ABE, Distributed Hash Table (DHT) network, and Identity-Centric Timed-Release Encryption (IDTRE).	Finally, the enveloped ciphertext is kept on cloud servers, and the ciphertext shares are dispersed over the DHT network. Performance and security analyses demonstrate that the recommended approach outperforms the current systems in a cloud setting.	It is still being done to safeguard user privacy and defend DHTs against attacks like Sybil.
[86]	Security and detection mechanism in IoT-centric CC utilizing a hybrid technique	The critical processing method of RC4 and RSA algorithm	The entire parametric comparison indicates that our framework performs better than conventional methods.	Many apps deem RC4 insecure due to its numerous known flaws. Unstrung MACs could be utilized to reverse cryptography, and critical scheduling is one of its weaknesses.
[87]	A Privacy-Enhanced Retrieval Technology for the Cloud-Assisted IoT	PERT, a privacy-improved retrieval technology for cloud-assisted IoT	Comprehensive experiments and performance analysis have demonstrated the technology's efficacy for data privacy. The architecture can securely and effectively retrieve saved data while lowering computing costs by shrinking operations.	PERT emphasizes time a lot, which might make it restrictive for projects that need to be flexible.

Table 2.1: Comparative Table for Integrated Lightweight and Hidden Hierarchy Cipher Text-policy Attribute in Cloud-Assisted IoT “Table Continued”.

[88]	Secure and Lightweight User Authentication Scheme for Cloud-Assisted IoT	It provides a fresh approach to user authentication that is safe and requires little processing power on gateways.	Regarding security and performance, the recommended approach outperforms ten modern methods by meeting all twelve security requirements while using the most negligible computation and storage on gateways.	Reliable internet access is required for IoT devices.
[89]	Facilitating secure time-series data sharing through homomorphic encryption in cloud-assisted IoT	SmartCrypt, a method for exchanging and storing data, supports efficient analytics over protected data with time series.	The findings demonstrate that, compared to the best-performing scheme in the modern and advanced, SmartCrypt increases performance by 9%, scalability by 20%, and query time by 17%, 32%, and ranged time per query.	Usability problems may arise during PKI (Public et al.) integration, particularly for businesses that do not currently use PKI.
[90]	CP-ABE for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System	CP-ABE	It may also check the operation of any AC added to the IoT. This recommended study also examines security against indistinguishable adaptive selected ciphertext attacks.	This strategy has some disadvantages even though it somewhat preserves the policy's privacy: In certain instances, the access policy continues to divulge sensitive and vital information included in the attribute name Since the end user must determine which attribute value is precisely contained in the ciphertext, the decryption time may be super-polynomial if each attribute has several values.

Table 2.1: Comparative Table for Integrated Lightweight and Hidden Hierarchy Cipher Text-policy Attribute in Cloud-Assisted IoT “Table Continued”.

[91]	SHARE-ABE: an effectual and secure data-sharing framework centered on CP-ABE and Fog computing	SHARE-ABE, a novel collaborative approach	Tests and evaluations of the security characteristics show that the recommended strategy is effective and safe, particularly for IoT devices with limited resources.	Devices with low processing power, including the IoT, may encounter difficulties with the resource-intensive encryption and decryption processes in CP-ABE.
[92]	Authority CP-ABE for security improvement in a cloud storage unit	The recommended approach offers many traceable, dynamic policy updating procedures and high-security multi-authority.	To verify its efficacy, the recommended approach can be compared to traditional approaches, such as revocable signature (CP-ABSC) with CP-ABSC with Accountable and Verifiable Outsourced Designcrypton (CP-ABSc-AVODs) and Ciphertext-Policy Attribute-Based Signcrypton with Outsourced Designcrypton (CP-OABSC).	It can take much time to update the policy regularly. Every update must be carefully examined, approved, and shared with all pertinent parties.

3. RESEARCH METHODOLOGY

Our compact approach is centered on the fundamental paradigm of CP-ABE. Consequently, it encompasses the 4 key algorithms listed below, implemented by the various system components.

Config: This procedure is conducted by the Trusted Authority (TA) to determine the system variables, such as the originator and the EC. It generates a PK and an MSK.

Au, MK KeyGen: The user's set of attributes (Au) and the MK are input into the KeyGen (Au, MK) algorithm, which the TA runs when the user submits a request to generate the user's SK.

Encrypt (M, Ac, and PK): The information owner obtains the encryption key (C), generated by the TA during setup, and employs the Encrypt algorithm for encrypting the data M with an access policy Ac.

Decrypt (C, SK, and PK): This algorithm uses the user's supplied data. It only accepts the Au by the AC set by the DO. It outputs the data M after receiving the user's SK, PK, and CT (C) as inputs.

3.1 SYSTEM DESIGN AND ARCHITECTURE

As shown in (Figure 3.1) a high-level system architecture is designed considering IoT data, DOs, cloud infrastructure, TA, and ABE.

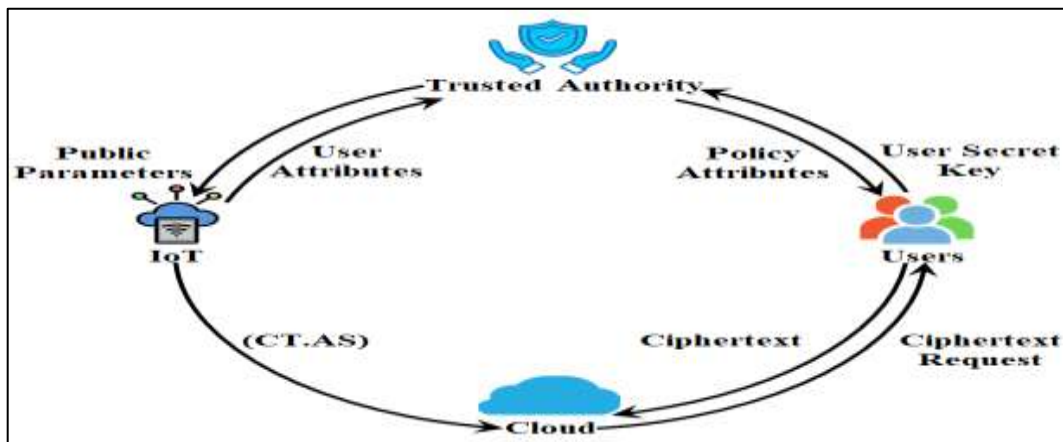


Figure 3.1: System Architecture for Cloud-assisted IoT Systems.

The application layer in the Cloud-IoT architecture serves as an interface between cloud platforms and IoT devices. Verified IoT devices and authenticated users can access the network system using this layer, including AC, database, and other system components. The physical layer is crucial in gathering data from IoT devices stored in cloud servers. Ensuring data security and privacy in IoT systems is a paramount concern, especially concerning the sensitive information stored within the network. Data privacy is a critical consideration for safeguarding such sensitive data. In the realm of IoT, the physical layer serves as the cornerstone, facilitating smooth communication between devices, guaranteeing the reliability of data transfer, and laying the groundwork for efficient IoT network functionality. The cloud's centralized nature increases the vulnerability of the data to attackers and thereby poses a significant threat to the security and privacy of the data collection process. This issue can be alleviated using the proposed HH-CP-ABE approach, wherein only authorized users can access the data, and the data is encrypted utilizing a lightweight encryption algorithm, as seen in (Figure 3.2).

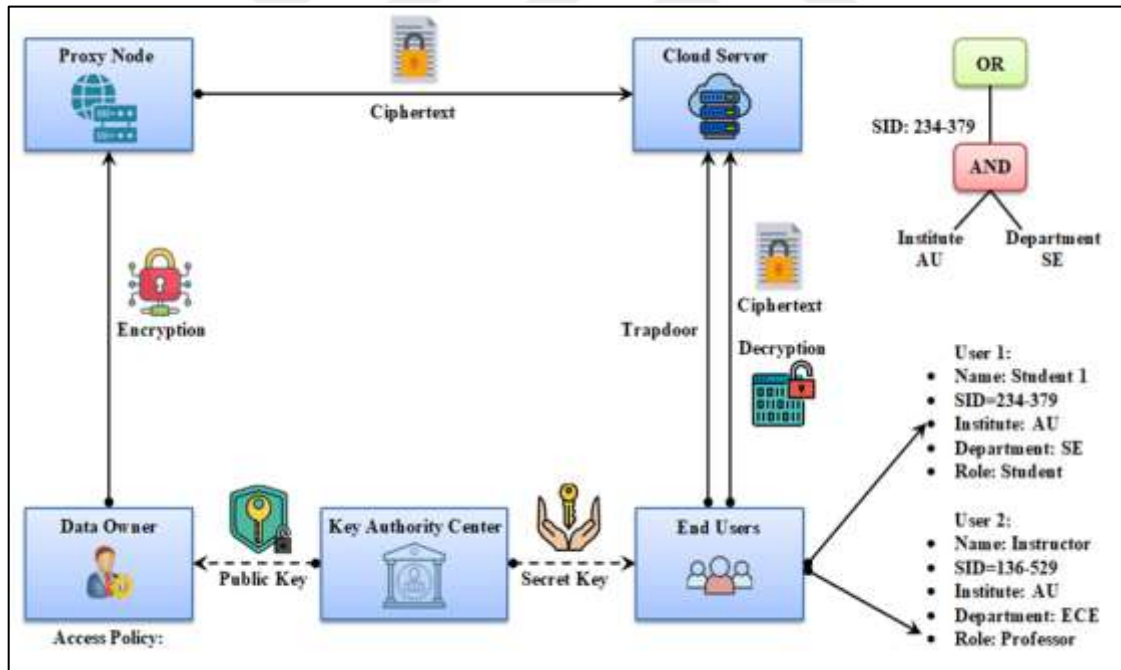


Figure 3.2: Proposed System Framework (HH-CP-ABE).

The encrypted data will be shared between multiple entities while maintaining privacy when stored on the cloud servers. In this work, synthetic data is used. They are tagged with specific attributes that determine their access policies. Individuals or entities who possess IoT data and want to share it securely. They specify the access policies for the data. The cloud

provider stores the encrypted IoT data and performs encryption and decryption operations. It also hosts the TA. The TA generates and manages the MSK and PK. It ensures proper key distribution to users.

As shown in (Figure 3.1) the DOs interact with the TA to set up encryption policies and generate keys. In the next stage, TA generates the MK and PK and distributes the PK to authorized users. Authorized users request access to encrypted IoT data, and the TA generates their SKs based on their attributes. DOs encrypt the IoT data using the access policy, generating ciphertext (C) and storing it in the cloud. Authorized users with valid attributes can request ciphertext and use their SKs to decrypt the IoT data. The information owner obtains the encryption key (C), generated by the TA during setup, and employs the Encrypt algorithm for encrypting the data M with an access policy Ac. Further, the decryption algorithm uses the data the user has supplied. It only accepts the Au by the AC set by the DO. It outputs the data M after receiving the user's SK, PK, and C as inputs. Furthermore, the DO initiates the creation of the policy update key, which is subsequently transmitted to the CSP in the event of an alteration in the access policy. The Policy AC Property stipulates that DOs can employ either Playfair or ciphertext encryption or even both simultaneously. The chosen encryption method is then utilized to update the ciphertext stored in the cloud. Then, if the attributes of the data user align with the ciphertext access policy, they transfer the SK to a trusted entity, which generates the accurate SK. The TA produces the final decrypted message upon completion of the decryption process. Should any inconsistencies or suspicions arise during this procedure, the relevant authorities receive the suspicious user's ID from the trusted party through the tracing algorithm.

3.2 PROPOSED SYSTEM MODEL

An LWC is a method that allows any message or information to be safely sent over a communication network between two or more distant parties while being watched over by an eavesdropper. Encryption and decryption are the two steps in every cryptographic scheme. Encryption converts a text message into a CT message, and it is again decrypted (decoded) into the original text message during the decryption process. Furthermore, each cryptographic scheme relies on a single shared key or a pair of PK and PRKs that translate text messages into C and vice versa. Higher expressiveness, access policy updates, broad attribute fields, and white box traceability are all supported by the recommended LWCABE

approach. The procedure by which the system can recognize the malicious user if a user purposefully gives their decryption key to an unauthorized user by utilizing the information contained in the key is known as "white-box traceability." The following examples highlight the characteristics of the indicated LWC-ABE method:

- i. Large attribute domain: public parameters do not increase linearly concerning attributes; instead, the number of authorized institutions affects their size. If the system is formed, then updating the system attributes is not required.
- ii. Policy modification: To comply with increasingly stringent security requirements, DOs regularly alter the policy access specifications and produce various CTs.
- iii. Additionally, the DOs flexibly adjust the data access properties by the modified policy. The system may monitor malicious users who distribute PRKs without authorization. White box traceability helps identify unauthorized users with minimal computational overhead by compiling various users and their access permissions.
- iv. Numerous legitimate authorities by presenting the multiple authority system, the issues of data integrity and the inadequacy of credibility of the single authority are resolved. Intentional or unintentional human error has the potential to compromise data integrity. Transfer errors occur when information is moved between devices, including inadvertent changes or data compromise. The key reasons for data integrity are malware/viruses, vulnerabilities, hacking, and other online threats.
- v. It supports any customized AC access method and monotonous access structure.

(Figure 3.3) represents the foundational operational blocks of the proposed LWC-ABE framework: the trusted party, system party, data users, DOs, attribute authorities, and CSP. Using network infrastructure as a rich, functional platform, a communications service provider offers telecommunication services or a combination of information and media services, content, entertainment, and application services across networks. First, the system setup is called to generate the public parameters.

Furthermore, in the first instance, attribution authorities, data users, DOs, and trusted parties receive access to some of the public parameters. Furthermore, constants and counter PPs generated for the CT encryption include values, nonces, and seed keys. Subsequently, the

attribute authorities generate PKs through the authority establishment process and distribute them to DOs, users, and trusted parties. Additionally, the attribute If the data user has the necessary authorization, authorities-centric characteristics are assigned centered on their request. The DO creates CT encryption for encrypted communication and sends it to the CSP. Additionally, the Playfair algorithm encrypts the data sent from the CSP.

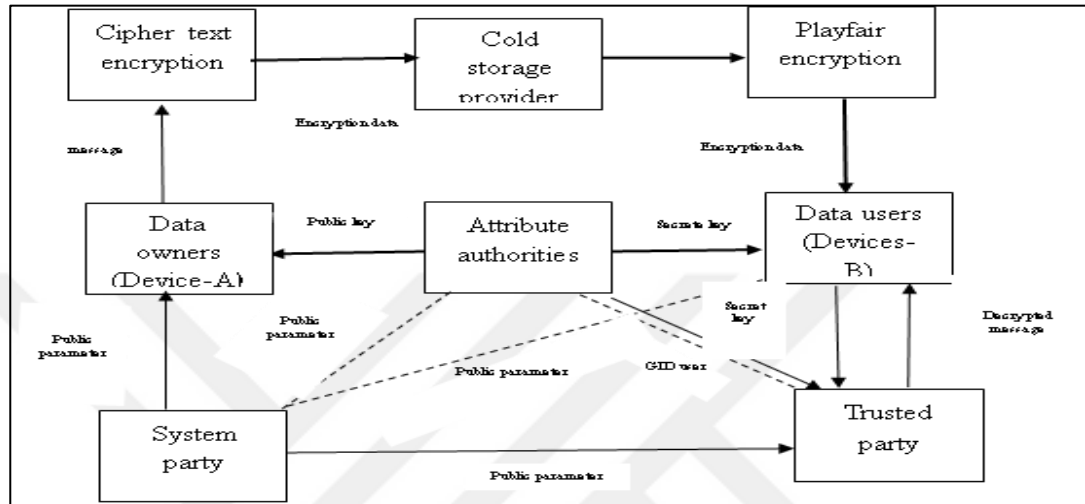


Figure 3.3: Proposed LWC-ABE System Model.

In addition, the DO generates the policy update key, which is then sent to the CSP in case of an access policy change. Policy Access Change Property states that DOs can simultaneously use Play Fair or CT encryption. The selected encryption technique will then update the CT in the cloud storage. Subsequently, if the data user's attributes align with the CT access policy, they forward the SK to a reliable entity, which generates the precise SK. The trusted party produces the final decrypted message after they have completed the decryption process. If a discrepancy or suspicion arises during this process, the attributed authorities receive GID from the trusted party via the tracing algorithm.

a. Hybrid Signcryption System

The lightweight block cipher with 31 rounds PRESENT is SPN. An 80/128-bit key is applied to transform the input of a 64-bit message into a 64-bit CT. Given the key size, there are two variations of PRESENT: There are two present values: 80-bit and 128-bit. The Present's S box has four bytes in it. Round keys, S box functions, and P box functions are thus added with each round. The issue is the weak session key in the current Signcryption scheme or the weak SK utilized for encrypting the message.

We recommend a hybrid system (Figure 3.4) that uses asymmetric encryption for securing the session key and circumvent these problems. The suggested system has three stages: setup, signcryption, and unsigncryption.

Setup phase: The central 3rd party that gives the registered user their key is called the key distribution center (KDC). Each registered sender (S) and recipient (R) in KDC has a unique set of PK and PRKs. Assume that prior to communication, sender S and recipient R authenticate themselves and obtain each other's PKs from the KDC.

Let p be the generator of G , and let G be the additive group of order q . Next, render a security parameter K , and let E be the EC over finite field Z .

P is the curve's point; N is the largest prime number; choose $x = [0, N-1]$.

Public key $PK = x * p$,

private key $PRK = x$,

where PK_R , PRK_R , and PK_S , PRK_S , Respectively, it represents the PK and PRK of the sender and recipient.

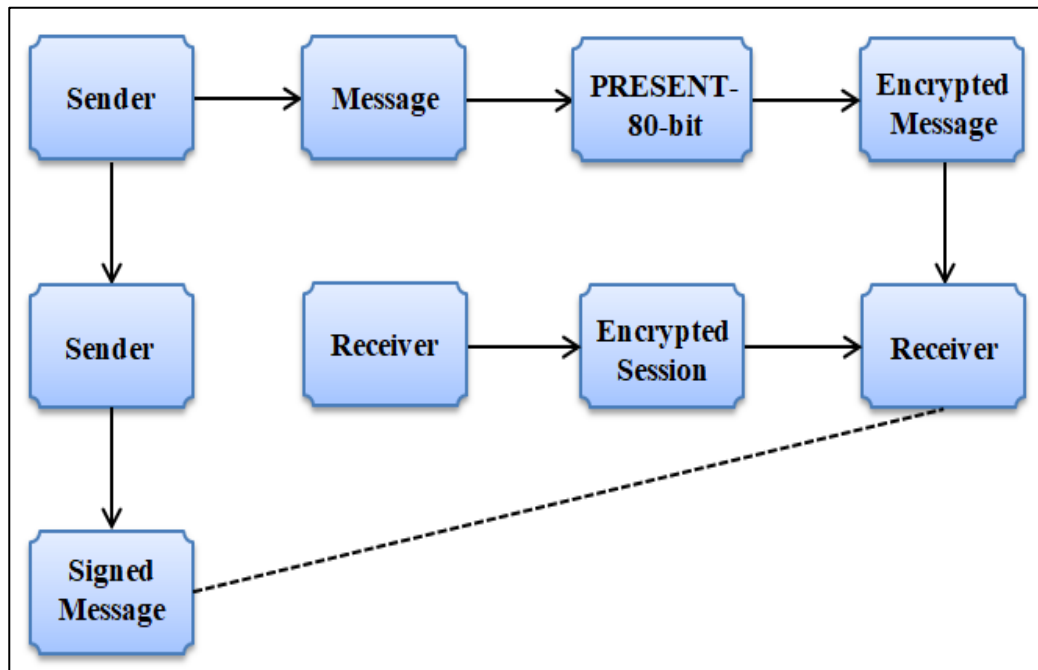


Figure 3.4: Hybrid Lightweight Signcryption.

In the signcryption phase, the message's MAC code must be calculated to create a digital signature. Following MAC code computation, the PRK of the sender is utilized for signing the code digitally. As a result, the non-repudiation is guaranteed, and the sender cannot dispute the message authorship in the future.

In the encryption phase, the sender's secret session key, consisting of 80 bits, encrypts the 64-bit message M utilizing the ultra-lightweight symmetric block cipher PRESENT. Consequently, a 64-bit CT is produced.

$$M \rightarrow CT(M, S_K)$$

The recipient must receive the session SK securely. To protect it, a hybrid strategy is proposed that encrypts the session key—which is then utilized for encrypting the message—using the ECC algorithm. Thus, the recipient's PK encrypts the 80-bit secret session key using the ECC algorithm.

$$S_K \rightarrow CK(R_{PK}, S_K)$$

The recipient receives the CT and cipher key with the signature.

$$CT + CK \rightarrow R.$$

The sender receives an encrypted copy of the CT along with the cipher key. To reduce the time required for serial processing, the processes of encryption and signature generation are carried out in parallel.

Determine the values of $k \in Z_q$ and $t \in Z_q$.

Then, compute $b = t \bmod q$,

$$k_2 = k \text{ PRKR, and}$$

$$M = (m_1, m_2, m_3 \dots m_i).$$

where SK is the symmetric key of the 80-bit PRESENT algorithm, and m_i is the message block of size 64 bits.

$$C_i = (SK, m_i, \{E(SK) \rightarrow SK, PK_r\})$$

Find the value of $c = m \oplus H_2(PRKS, K_2)$

Compute $h = H_3(m, PRKS, K_2, b, c)$

The formula is $w = (h + k) - 1 \cdot t - 1 \bmod q$.

Therefore, the receiver $PRKR \rightarrow C(SK) \rightarrow m$ can retrieve both the encrypted message $\sigma = (c, PRKS, b, v, w)$ and the m message.

Proposed lightweight hybrid Signcryption scheme.

Step 1: $D(d_1, d_2, d_3) \rightarrow M$

Where d_1, d_2 , and d_3 are the various data from the device D , and M is the message to be encrypted.

Step 2: $M \rightarrow CT(M, S_K)$

The message M is encrypted to a cipher text CT by applying the session SK of the PRESENT algorithm. S_K on the message M .

Step 3: $SK \rightarrow CK(R_{PK}, S_K)$

The secret key SK is encrypted into a cipher key CK using the receiver R PK on the secret key. S_K .

Step 4: $S \rightarrow CT + CK \rightarrow R$

The sender will encapsulate the cipher text CT and cipher key CK to the receiver R .

Step 5: $S_{PRK} \rightarrow SC(M) \rightarrow R$

The sender S private key PRK is used to sign the message and send it to the receiver R . Steps 4 and 5 are computed in parallel. The sender ensures the non-repudiation by signing the message M using the PRK .

Step 6: $R \rightarrow CT + CK$

The receiver R will receive the cipher text CT and the cipher key CK .

Step 7: $R \rightarrow S_K (CK, R_{PRK})$

The receiver R computes the secret key SK by applying the private key PK on the cipher key CK.

Step 8: $R \rightarrow US (S_{PK}, M)SPK$

The receiver verifies the authenticity of the message using the sender's S PK. P_K On the message M.

Step 9: $R \rightarrow M(CT, S_K)$

Steps 8 and 9 are computed in parallel. The receiver, R, decrypts the message M by applying the secret session key. S_K On the cipher text CT.

b. Unsigncryptpion phase

The CT, the cipher key, and a signature are all verified at the receiving end. The receiver's PRK is utilized for decrypting the cipher key, which is further applied to the CT to decrypt the message. The digital signature of the message is confirmed using the sender's PK.

To unsign the message, the process of decrypting it and verifying its signature happens simultaneously.

The recommended approach offers integrity, confidentiality, and non-repudiation in a single step while also being more time-efficient.

Given a signcrypted $\sigma = (c, PRKS, b, v, w)$, m's signature can be confirmed using v and w. If it is legitimate, the message is ignored, and the signature is valid for message m.

Do the message calculation use $m = c \oplus H2 (PRK_s, K2)$

Subtract m from h to $H3 (m, PRK_s, K2, b, c)$

3.2.1 Lightweight Authenticated Encryption

The proposed system operates in three phases: registration, message signing and encryption, and message decryption and signature verification. (Figure 3.4) details the data integrity and

authentication process stage. The public and private values are generated at each phase using a distinct set of EC values.

a. Registration Phase

Let's look at client node A, which has the identity ID_A . As indicated by equation 3.1, node A sends a registration request message (REQ_A) to the server.

$$A \rightarrow S : REQ_S = H1(ID_S) \parallel ID_A XOR key_M \quad (3.1)$$

When the server receives a REQ_S message, it computes the HV $H1(ID_S)$. After comparing the HVs $H1(ID_S)$ and $H1'(ID_S)$, communication is cut off if the comparison is unsuccessful. If the comparison is successful, the server notifies the client that the response has been accepted and XORs the client's identity to get the ID_A . (Figure 3.5) depicts the registration phase's whole operation.

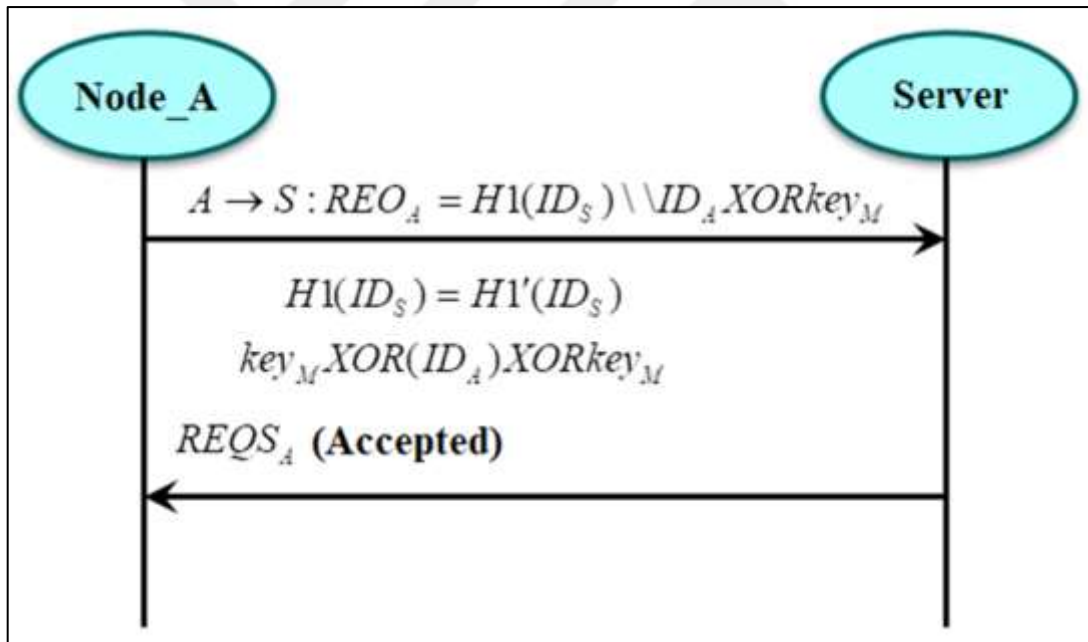


Figure 3.5: Registration Phase.

b. Key Generation and Exchange Phase

During this phase, two distinct ECs—EC-L1 and EC-L2- are utilized for generating two pairs of PK and PRK. Using ID_A and λ_{A_j} values, the PRKs are Pri_{A1} and Pri_{A2} , with Pri_{A1}

$= \text{XOR}(ID_A, \lambda_{A1})$ and $Pri_{A2} = \text{XOR}(ID_A, \lambda_{A2})$. The random values betwixt n and DT_A , (deployment time of node A) are represented by the λ_{Ai} .

Then, like [83], the inverse of the PRKs is utilized to generate the PKs as $Pub_{A1} = (Pri_{A1}^{-1} \text{ mod } n) * G_1$ and $Pub_{A2} = (Pri_{A2}^{-1} \text{ mod } n) * G_2$ represents the PKs and G_1 and G_2 represent the generator points of the EC-L1 and EC-L2 curves, respectively. Following key generation, A sends the EC reference values and PKs to the server as

$$\Delta = X_1 || X_2, X_2 = \text{XOR}(EC - L1_r || EC - L2_r, key_M), \text{ and finally } X1 = Pub_{A1} || Pub_{A2}$$

After receiving Δ , the server node divides X_1 and X_2 and uses key_M To perform an XOR operation to get the values of the EC. After the values of the EC are known, the server node also uses n and DT_s , to produce random values λ_{S1} and λ_{S2} that are comparable to λ_{A1} and λ_{A2} . Subsequently, the PRK and PK are created as follows equation 3.2.:

$$Pub_{s1} = Pri_{s1}^{-1} \text{ mod } n * G_1; Pub_{s2} = (Pri_{s2}^{-1} \text{ mod } n) * G_2. \quad (3.2)$$

Ultimately, the server sends the node A its PKs as $H_2 (Pub_{s1} || Pub_{s2}) || Pub_{s1} || Pub_{s2}$. Conversely, node A obtains the PKs, calculates the hash $H_2' (Pub_{s1} || Pub_{s2})$, and compares H_2 and H_2' to verify the message's legitimacy, as shown in (Figure 3.6).

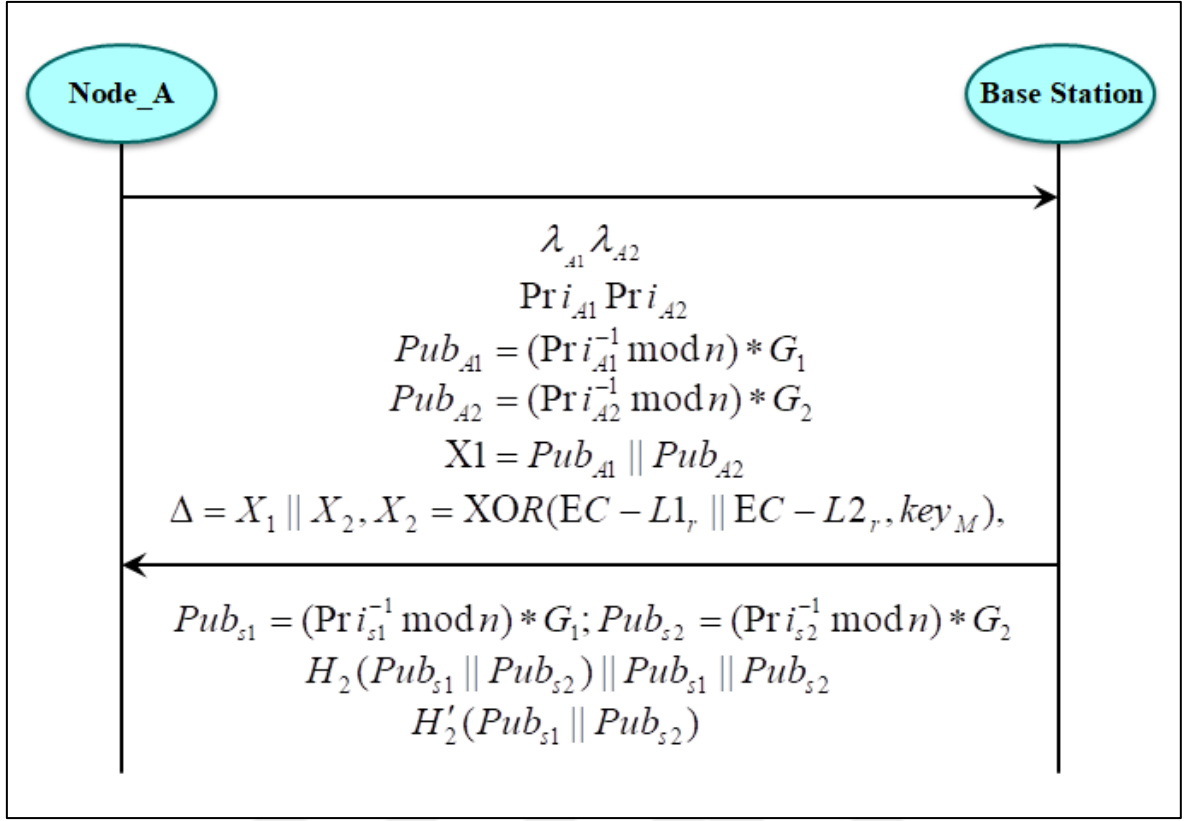


Figure 3.6: Key Generation and Exchange Phase.

c. Message Signing and Encryption Phase

Consider a message as Msg that needs to go to the server. See (Figure 3.7) Initially, cBLAKE2b generates the message digest, and the output is $H3(Msg)$. Next, for a single communication, the client chooses at random an integer. λ_{pm} between n and λ_{A1} . The client acquires a session key for each message by performing scalar point multiplication, which is defined by $\epsilon = \lambda_{pm} * G_1$.

The signature pair of ECDSA is derived as follows by taking into account the x coordinate of ϵ : $\sigma_1 = \epsilon_x \bmod n$ and $\sigma_2 = [(\sigma_2 * \lambda_{pm}) - H3(Msg)] * Pri_{A1} \bmod n$. Ultimately, the client concatenates the message and HV as T_{msig} before encrypting the data. T_{msig} appears to be $T_{msig} = Msg || H3(Msg)$.

Over the T_{msig} , The client will carry out the ECIES-based encryption procedure. By using KAF based on ECIES, a shared key known as A MAC key is known as K_{MAC} and K_{ENC} regenerated. To attain the CT, $tmsig$ is encrypted utilizing. K_{ENC} . Then, a MAC tag τ

is generated using the ECIES MAC function. Ultimately, a cryptogram (CPG_A) is produced by joining CT, τ , and XOR values of the PK and signature pair on the server and sent to the server see equation 3.3.

$$CPG_A = CT \parallel \tau \parallel \text{XOR}(Pub_{s2}, (\sigma_1, \sigma_2)) \quad (3.3)$$

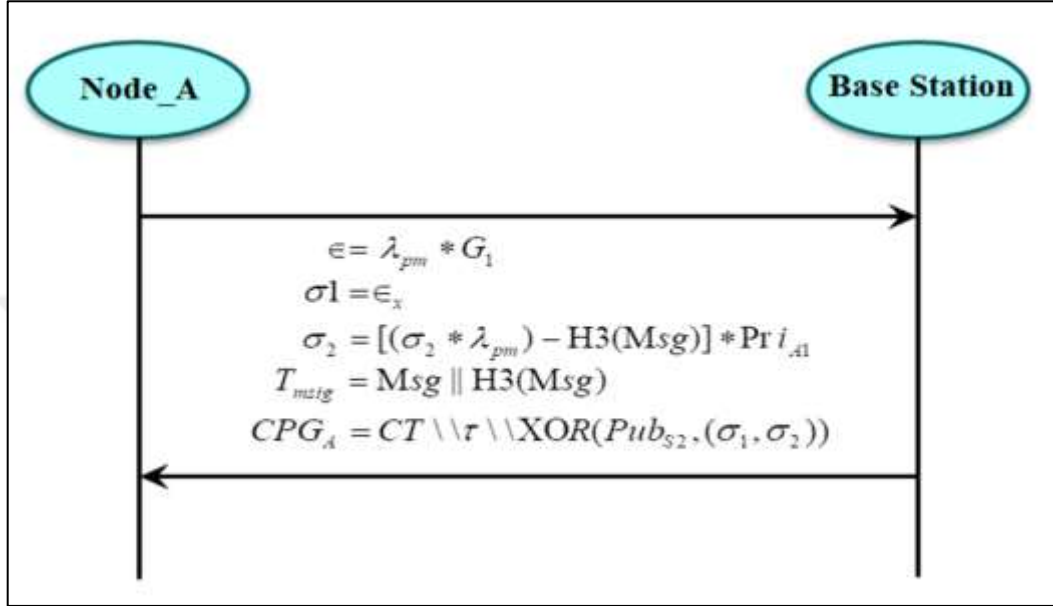


Figure 3.7: Message Signing and Encryption Phase.

d. Message Decryption and Verification Phase

The CPG_A server creates the K_{ENC} and K_{MAC} Keys and splits the cryptogram upon receiving it. After that, the server creates the MAC tag τ' and contrasts it to τ . When τ' does not equal to τ , the connection breaks. If not, the server uses K_{ENC} to decrypt the CT and separate the T_{msg} values, as shown in (Figure 3.8.).

The server verifies that. σ_1 and σ_2 Fall within the range of $[1, n - 1]$. Next, use the cBLAKE2b hashing algorithm to generate the message's HV, or $H3(Msg)$. $H3'(Msg)$ and $H3(Msg)$ are compared afterward. The conversation ends if the comparison is unsuccessful. If not, $\omega = \sigma_1^{-1} \bmod n$, $v_1 = H3(Msg) * \omega \bmod n$, and $v_2 = \sigma_2 * \omega \bmod n$ are the signatures that are verified. With these values, the server finds the point on the curve where $(x, y) = v_1 \times G + v_2 \times Pub_{A1}$ and rejects the signature if $(x, y) = 0$. If not, the server calculates $t = x \bmod n$ by converting the x-coordinate of (x, y) to an integer x . Accept the signature if $t = \sigma_1$, Or else, reject it.

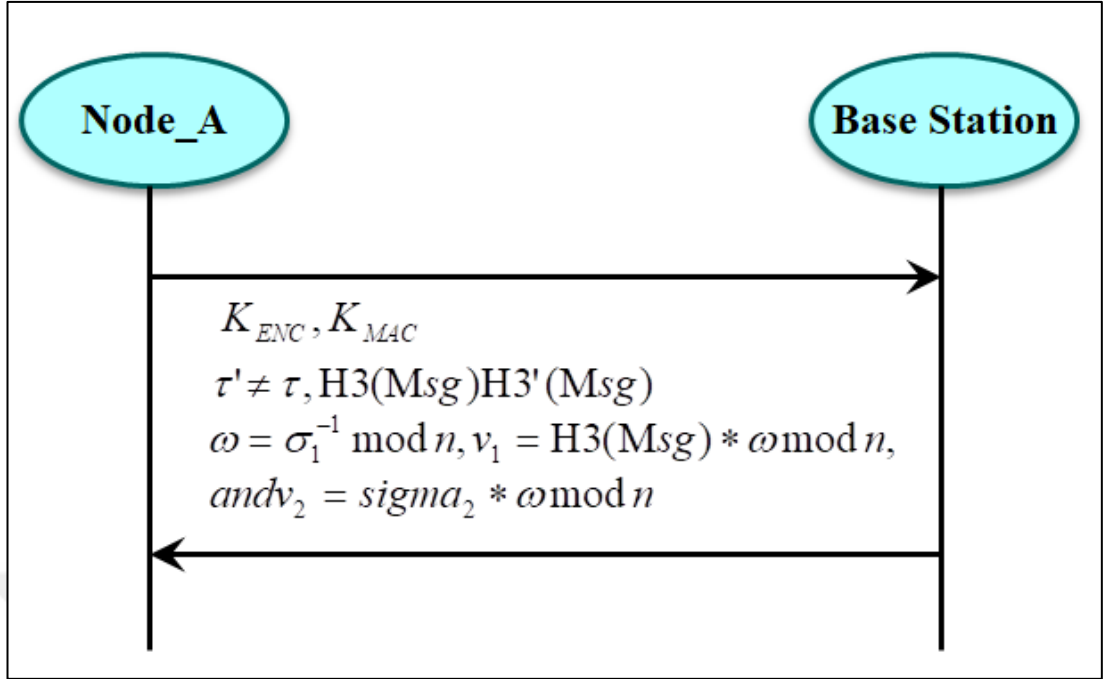


Figure 3.8: Message Decryption and Verification Phase.

3.2.2 Cipher Text-policy Attribute-Based Encryption (CP-ABE)

Formal definitions of CP-ABE security are first provided. The history of bilinear maps is then provided. An access structure is defined and incorporated into security definitions, imitating the work of [93]. However, in these definitions, the labels for various encrypted data sets will come from the access structures, while the attributes will characterize the users.

a. Definitions

Definition 1 (Access Structure [1])

Let $\{P_1, P_2, \dots, P_n\}$ Represent a group of parties. If $A \subseteq 2^{\{P_1, P_2, \dots, P_n\}}$ is a monotone collection, then $\forall B, C$: if $B \in A$ and $B \subseteq C$, then $C \in A$. A collection of non-empty subsets of $\{P_1, P_2, \dots, P_n\}$ that is, $A \subseteq 2^{\{P_1, P_2, \dots, P_n\}} \setminus \{\emptyset\}$, is called an access structure (or monotone access structure). The sets that are not in A are referred to as unauthorized sets, and the sets that are in A are referred to as authorized sets.

In this instance, the attributes assume the role of the parties. Therefore, the authorized sets of attributes will be present in access structure A . We limit our focus to monotone access structures. Nonetheless, our techniques can also be used to (inefficiently) realize general

access structures by treating the not of an attribute as a stand-alone attribute. Consequently, there will be twice as many attributes in the system.

Henceforth, unless indicated, an access structure will be understood to be monotone.

The four main algorithms of a CP-ABE scheme are Setup, Encrypt, KeyGen, and Decrypt. We also provide a 5th algorithm, Delegate, as an option.

i. Setup

Only the implicit security parameter is required as input by the setup algorithm. Together with MSK, it outputs the public parameters PK.

ii. Encrypt (PK, M, A)

The message M, the access structure A across the universe of attributes, and the public parameters PK are inputs to the encryption algorithm. Through the use of encryption, the algorithm will create a CT that can only be decrypted by a specific user who meets the access structure's requirements. We're going to assume that the CT implicitly includes A.

iii. Key Generation (MK, S).

The Master Key (MK) and a list of attributes S that define the key are inputted into the key generation algorithm. It generates a PRK, SK.

iv. Decrypt (PK, CT, SK).

The CT, which includes an access policy A, the public parameters PK, and the PRK SK, which is a PRK for a set S of attributes, are fed into the decryption algorithm. The algorithm will return a message M upon decrypting the CT if the set S of attributes meets the requirements of the access structure A.

v. Delegate (SK, S[~]).

An SK for a set of attributes (S) and a set of attributes ($S^{\sim} \subseteq S$) are inputted into the delegate algorithm. It outputs the set of $\sim$ attributes $S^{\sim}$ along with an SK.

A security model for CT policy ABE schemes is now being described. The security model permits the adversary to query for any ineffective PRKs for decrypting the challenge CT,

just like IBE schemes [94],[95],[96]. Attributes identify the PRKs in CP-ABE; access structures determine the CTs. Accordingly, the adversary in our security definition will select to challenge encryption to an access structure A^* and may request any PRK S if S doesn't satisfy A^* .

b. Security Model for CP-ABE

Setup: After executing the Setup algorithm, the challenger provides the adversary with the public parameters, PK.

Phase 1 The adversary repeatedly creates PRKs that match sets of characteristics. $S_1, S_2, \dots, S_{q_1}$.

Challenge: The opposing team sends out 2 messages, M_0 and M_1 , of the same length. Furthermore, the adversary provides a challenge access structure A^* , which is unsatisfactory for any of the sets $\{S_1, S_2, \dots, S_{q_1}\}$ From Phase 1. The challenger encrypts M_b , Under A^* , and flips a random coin, b . The adversary receives the cipher text CTTM.

Phase 2 Phase 1 is executed again, stipulating that the access structure corresponding to the challenge is not satisfied by any of the attributes. $S_{q_1 + 1}, \dots, S_q$.

Guess: The opponent provides a guess, b' , for b .

In this game, the benefit of an adversary A is given by $\Pr[b' = b] - 1/2$. It is worth mentioning that the model can be effortlessly expanded to accommodate chosen CT attacks by permitting decryption queries during both Phases 1 and 2.

Definition 2 A secure encryption scheme that relies on attributes from the ciphertext and policy is one in which all polynomial-time adversaries have a minimal benefit in the abovementioned game.

Bilinear Maps: We give some information about groups whose bilinear maps can be computed efficiently.

Assume that G_0 and G_1 are two prime order p multiplicative cyclic groups. Let $e: G_0 \times G_0 \rightarrow G_1$ be a bilinear map, and let g be a generator of G_0 . The following characteristics of the bilinear map e are:

a. Bilinearity: $e(u^a, u^b) = e(u, v)^{ab}$ holds for all $u, v \in G_0$ and $a, b \in Z_p$.

b. Non-degeneracy: $e(g, g) \neq 1$.

If both the group operation in G_0 and the bilinear map $e: G_0 \times G_0 \rightarrow G_1$ can be computed efficiently, then G_0 is considered a bilinear group. From the fact that $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$, you can see that the map e is symmetric

3.3 LIGHTWEIGHT AUTHENTICATION ENCRYPTION FOR SECURING VALIDITY AND CONFIDENTIALITY OF SYSTEM COMMUNICATION

IoT security issues include system configuration, information management and storage, AC, privacy preservation, and authentication [97]. IoT devices provide responsive care and sensitive services that allow them to adapt to human needs and facilitate communication and convenience. However, security can be jeopardized, especially if user data is leaked or privacy is threatened [98]. Users must feel secure about their data and privacy. As a result, security is the main issue with the IoT, especially since its conventional communication channel is the Internet, which is vulnerable to attacks from both public and private cybercriminals. Hardware and physical security may be required in addition to system and infrastructure security measures [99].

Security refers to any action made in a computer system to guard against unauthorized access and misuse of data by an outsider. Daily malicious attacks include things like credit card theft, virus attacks on machines, financial data leaks, and the disclosure of private data. To determine computer system protection, the requirements that follow should be met:

- i. **Confidentiality:** It ensures that data on the system is shielded from unauthorized access from outside sources. There are various approaches to maintaining data confidentiality. Nonetheless, AC and cryptography techniques are the most popular ones. The process by which the intended data is encoded and rendered unusable by any entity absent the decryption key is termed cryptography. Practical encryption algorithms comprise DSA, AES, and RSA. Anyone whose identity cannot be verified is denied unauthorized access to sensitive information via the AC process. Yet, IoT never provides adequate protection since sophisticated encryption and verification algorithms cannot be used [99].

- ii. Integrity: Data not altered by an incomprehensible entity are considered integrated [100].
- iii. Authentication: This is the procedure used to confirm a user's identity when they access system resources [100].
- iv. Non-repudiation: The original node can't retract an action it has taken, like sending a message.
- v. Availability: Authenticated users can access system resources due to their availability.
- vi. Privacy: The privacy feature ensures that the user's identity can't be deduced from their actions within the system [99].
- vii. Anonymity: The RFID authentication scheme keeps the tag's identity anonymous for privacy preservation.
- viii. Authorization: Authorization is a security mechanism that determines user rights or degrees of access to system resources, such as software, files, services, data, and application features.

Rapid advancements in communication technologies, which provide constant access to information resources at any location and at any time, are driving an upsurge in the usage of IoT. Although the idea of a future Internet raises some questions about network scalability and system security, it also renders users a superior experience. The main security issue with IoT is delivering integrated, private, and authentic information [100].

Traditional cryptography models like the AES, RSA, and ECC function well on systems with large amounts of memory and processing power. Still, they struggle to function well on embedded systems and sensor networks. The computing capacity of sensors, smart cards, and other tiny devices in the IoT is frequently constrained. Therefore, when applied to systems with limitations relating to computational requirements, physical size, energy consumption, and memory limitations, LWC models are proposed to tackle numerous issues of standard cryptography. Secure encryption can be used on devices with limited resources thanks to LWC's restricted resources.

Lightweight encryption is a good technique to guarantee data security. Encryption is generally used as a tool for system security. The International Electro-Technical Commission (IEC) and the International Standards Organization (ISO) uphold information and communication technology standards. The following criteria should be met by algorithms for them to be classified as " LWC ":

- a. **Security Strength for Lightweight Cryptography:** While 80-bit security is the minimum, it is suggested that systems that will render security for longer periods use at least 112-bit security. **Smaller key size:** A lightweight key, which is smaller in size, is needed to attain low power consumption and a limited battery life.
- b. **Software Implementation Attributes:** Fewer resources should be needed for the code size and necessary RAM size than in the current standards for the same platform.
- c. **Hardware Implementation Attributes:** When compared with current ISO standards, the chip area covered by the cryptographic model and the energy consumption should be lower. Stream ciphers are symmetric key ciphers that create an output cipher stream by synthesizing plaintext digits with a pseudorandom key stream. They are also known as state ciphers because they have a secret internal state that is utilized for generating the pseudorandom key stream. During the encryption process, a bit-wise XOR operator is typically used to combine the plaintext and key stream. The same key stream is generated at the receiving end and is utilized for decrypting the cipher and returning to the original plaintext. Cryptography algorithms known as block ciphers process data in units known as blocks. To create ciphertext blocks, plaintext blocks are integrated with a key. It is utilized for hashing messages, generating random bits, data encryption, message authentication, etc. AES and DES, which were created as distinct international standards, are presumably the two most significant block ciphers in use today. A message of any kind can be sent over any network while maintaining its security, thanks to cryptography. This goal makes use of numerous encryption algorithms. The 2 primary approaches are asymmetric (public-key) encryption and symmetric (secret-key encryption) encryption.
- d. **Security Threats for IoT, Countermeasures Based on Encryption :** The greatest security risk posed by IoT systems compared to traditional IT systems is the possibility of cyberattacks even when collecting data from real-world devices. By gathering data from

numerous sensors placed in production equipment, analyzing it, and executing autonomous control in RT, for instance, the application of IoT to a plant can greatly increase its productivity and maintainability. Significant harm could arise from falsifying sensor data during this process, leading to inaccurate analysis results and erroneous control. Furthermore, keeping measurement data and control commands private is crucial from a competitiveness perspective because they are linked to production and management expertise and are considered trade secrets. Even when there isn't a problem right now, it's important to consider potential threats' consequences.

By implementing data protection for confidentiality and integrity on sensor devices, encryption can be used as an effective countermeasure against potential threats. One purpose of LWC is to make it possible to use secure encryption on devices with constrained resources. Cellphones and other communication systems already use encryption by default on the data link layer. Even in this scenario, end-to-end data protection from the device to the server and independent security from the communication system can be achieved through application layer encryption. Then, encryption must be used on unused resources and at the application's processor handling, so it should ideally be as light as possible.

3.4 CIPHER TEXT-POLICY ATTRIBUTE-BASED ENCRYPTION (CP-ABE)

a. Attribute-based Encryption (ABE)

A sort of PK cryptography known as attribute-centric encryption allows any user to be uniquely recognized by a collection of attributes, namely their name, department, title, or ID. 2 keys are utilized in a traditional PK cryptography system: a private or SK only known to the message's recipient and a PK known to all. To send User2 a secure message, User1 encrypts the message utilizing User2's PK. After that, User2 decrypts it with her PRK. (Figure 3.9) depicts the traditional PK cryptography scheme. Assume a DO desire to put File 1 in central data storage in encrypted form, making it accessible only to Users 1, 2, and User 3. Using the PKs of Users 1, 2, and 3, PRU1, PRU2, and PRU3 are used to encrypt File1. This results in the creation of three distinct CT files: CT1: $Encrypt_{pRus1}(File1)$, CT2: $Encrypt_{pRus2}(File1)$, and CT3: $Encrypt_{pRus3}(File1)$,. Only User1's PRK PRU1 can decrypt the ciphertext file CT1, User2's PRK PRU1 can decrypt CT2, and User3's PRK PRU1 can decrypt CT3.

Every user in attribute-centric encryption uses the same global PK (GPK). Every user has a unique PRK provided by a centralized, reliable authority. Each user uses the GPK to encrypt the file, and they each utilize their PRK linked to a specific set of properties to decrypt it. Figure 3.8 depicts the ABE scheme. Assume that the DO wishes to store File1 in encrypted form on the central data storage, with access restricted to User1, User2, and User 3. Using the GPK, it encrypts File 1 and creates the CT file $CT: \overset{a}{\text{Encrypt}}_{GPK}(\text{File1})$. Decrypting the ciphertext file CT with a different user's PRK linked to a set is possible.

Therefore, if a system has 'n' users, then using conventional PK cryptography to render fine-grained AC will result in 'n' different ciphertext files and '2n' keys. In contrast, the attribute-centric encryption approach will produce a single CT file and 'n+1' keys. As a consequence, both encryption time and ciphertext storage are mitigated.

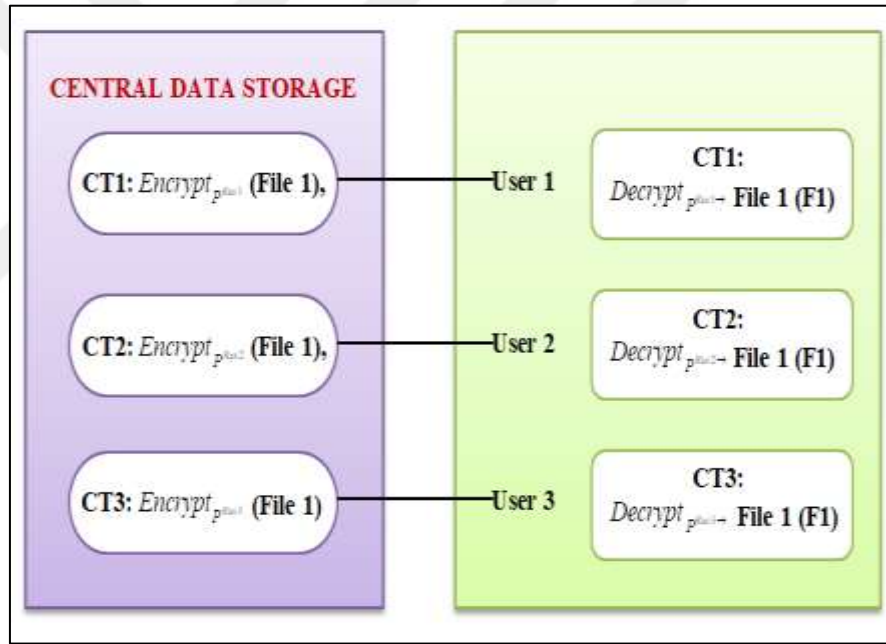


Figure 3.9: Conventional Public Key.

b. Cipher text Policy Attribute-Based Encryption (CP-ABE)

All users of CP-ABE share the same GPK. Every user's PRK is connected to a collection of attributes, and access policies are explained using logical operators over attributes during the encryption process [101]. A tree access structure is defined for access conditions, with external nodes representing distinct attributes and internal nodes acting as logical operators. Only users whose properties specified during key generation satisfy the access structure can

access a ciphertext file. Access conditions are logical formulas constructed from the values listed in the attribute list and logic operators. The logic operators AND, OR, and t of n threshold gate (in which t out of n attributes should be matched in the access conditions) can be utilized for defining the access conditions. Therefore, 1 of n threshold gates is represented by the OR operator and n by the AND operator. A user should possess a key to match attributes to decrypt a ciphertext. At the moment of encryption, the file's access conditions are established and embedded.

The DO uses the access policy T for encrypting a message M . Users who meet the T requirement in their attribute set for decrypt the ciphertext CT . The CP-ABE technique is applied in the following four steps:

- i. Setup ($1k$): GPK, MSK
- ii. This process creates a GPK and MSK based on a security parameter k . The administrator uses MSK and GPK for generating every single user's key. Every user is also familiar with GPK, which is utilized in encryption as well as decryption.
- iii. Key-generation (GPK, MSK, S): PRK
- iv. This process generates an SK for a particular user based on inputs such as the GPK, MSK, and set of user attributes S . For each user in the system, the administrator generates the PRK.
- v. Encryption (GPK, M , T): CT
- vi. Using GPK, this process encrypts a file M under the designated decryption policy T and produces ciphertext, CT . By defining access conditions, any user in the system can encrypt a file, limiting its decryption to only those whose attributes match the access policy. Relational operators such as " $\setminus$," " $>$," " $\setminus=$," " $>=$," and " $=$ " can be used to define policies, particularly when it comes to numerical attributes.
- vii. Decryption (PRK, CT , GPK): M

PRK, CT , and GPK are input by this process. When the user's attributes, S , meet the requirements of the access policy T , PRK decrypts file CT , allowing the user to access the generated file M .

c. Construction of Proposed CP-ABE

A set P of descriptive attributes is linked to PRKs in the CP-ABE scheme being presented. A user uses an access tree structure, T , to specify the access policy when they wish to encrypt a message. The internal nodes stand in for threshold gates or logic operators. The threshold value of an internal node will be $1 \leq t_x \leq n_x$ if it has $t_x = n_x$ children numbered as $1 \dots n_x$. The threshold gate functions as an AND operator if $t_x = n_x$ and as an OR operator if $t_x = 1$. The leaf node, or external node, z , denotes the attribute value, $\text{att_val}(z)$ (connected to that node).

To construct the proposed CP-ABE scheme, bilinear pairing $e: G_0 \times G_0 \rightarrow G_1$ is used for pairing operations in which G_0 and G_1 are 2 multiplicative cyclic groups of prime order p . The two properties listed below are satisfied by the binomial map e :

First, bilinearity According to this property, the mapping function needs to meet the requirements of equation 3.4.

$$e(u_1^{a_1}, u_2^{b_1}) = e(u_1, u_2)^{a_1 b_1} \forall u_1, u_2 \in G_0 \text{ and } a_1, b_1 \in Z_p \quad (3.4)$$

Second, non-degeneracy equation 3.5: $e(g, g) \neq 1$, where g is the generator of the group G_0

$$3.5: e(g, g) \neq 1 \quad (3.5)$$

Lagrange coefficient $\Delta_{t,s}$, where $t \in Z_p$ and s is the set of elements in Z_p , is also used in the construction. The definition of the Lagrange coefficient $\Delta_{t,s}$ see equation 3.6:

$$\Delta_{t,s}(y) = \prod_{k \in s, k \neq t} \frac{y-k}{t-k} \quad (3.6)$$

An extra hash function $H: \{0, 1\}^* \rightarrow G_1$ is also used in the CP-ABE development to map any attribute designated as a binary string to a random group element. The recommended PABE scheme is now built as follows:

Setup: The GPK is generated as $(G_0, g, h, f, e(g, g)^\alpha)$ and the master SK MSK is generated (β, g^α) by this procedure.

With generator g and prime order p , G_0 is a bilinear group with generator g , $h = g^\beta$ and $f = g^{1/\beta}$ and two random integers $\alpha, \beta \in Z_p$.

Key generation: About a set of attributes S, this process generates the PRK for each user in the manner described in equation 3.7 below:

$$\text{PRK}=\text{D}=g^{(\alpha+i)/\beta}, \forall s \in S: D_s = g^i \cdot H(s)^{i_s}, D'_s = g^{i_s} \quad (3.7)$$

The process selects $i_s \in \mathbb{Z}_p$ and the $i \in \mathbb{Z}_p$ for every attribute $s \in S$ at random.

Encrypt: This process encrypts a file using the specified access policy known as the tree access structure T. A polynomial q_x is selected for every node x in the tree T, regardless of whether it is internal or external. This process begins at the root node of the tree T and ends with assigning the degree d of the polynomial q_x to each node x, which is equivalent to $k_x - 1$ (k_x being the threshold value of node x).

To set $q_{root}(0) = n$ for the root node, a random number $n \in \mathbb{Z}_p$ is selected, d_{root} and is chosen to define polynomial fully q_{root} . In the tree, $q_x(0)$ for a node 'x' is defined as:

$$q_x(0) = q_{parent(x)}^{index(x)}.$$

In a similar d_x , selects d to define polynomial fully q_x , the ciphertext CT is built as follows in equation 3.8:

$$\text{CT} = (T, C = \text{Me}(g, g)^{\alpha n}, C = h^n \forall k \in K : C_k = g^{q_k(0)}, c'_k = H(\text{att_val}(k) q_k^{(0)}) \quad (3.8)$$

Where K denotes the collection of external nodes comprising attribute values in T.

Decrypt: This algorithm merely calls the function DecryptNode (CT, PRK, root) for the access tree T's root node. If the set of attributes S satisfies the access policy, we set $R = \text{DecryptNode}(\text{CT}, \text{PRK}, \text{root}) = e(g, g)^{i q_{root}(0)} = e(g, g)^{ij}$. The decryption process is then computed in equation 3.9.

$$C/(e(C, D)/R) = C/(e(h^j g^{(\alpha+i)/\beta} / e(g, g)^{ij} = M \quad (3.9)$$

The ABE setup, key generation, encryption, and decryption operations in our implementation are handled by the capable library, while pairing-based cryptographic operations are handled by the PBC library [102]. MSK and GPK files are generated during setup. Each user's PRK is generated using these GPK and MSK. The values of an attribute

are used as input for generating the PRK for an individual. Access policy must be applied to the data with the encryption command to store a file on Location1 in encrypted form. An access policy is a collection of requirements expressed as logical operators combined with logical formulas on attributes.

3.5 ENCRYPTION KEY SECURITY-AWARE RELIABLE DATA TRANSMISSION IN IOT-ENABLED CLOUD APPLICATION

IoT signifies an intricate network of interconnected physical devices embedded with software, sensors, and other technologies, facilitating RT communication and data collection with other systems and devices across the Internet [103]. IoT has transformed sectors, encompassing industrial automation, agriculture, smart cities, and HC in recent years by facilitating decentralized intelligence and pervasive sensing capabilities. Nevertheless, the vast scale of data created by IoT systems introduces significant drawbacks concerning SDT, particularly when incorporated with CC, which is often used to manage, store, and analyze this data [104].

With its elastic resources and scalable architecture, CC is widely utilized in IoT environments to perform complex computations, host storage solutions, and offer platform services for applications [105]. Yet, this cloud-assisted IoT architecture is highly vulnerable to diverse cybersecurity threats [106]. These comprise data breaches, unauthorized access, and authentication failures, compromising the integrity, availability, and confidentiality of transmitted data [107]. Hence, in modern computing systems, ensuring a robust security framework for data transmission over IoT-cloud networks has become a critical concern.

To overcome these challenges, a wide range of cryptographic methods have been adopted in recent research to improve the data transmission security in IoT-enabled cloud systems. Methods, including ECC and Post-Quantum Cryptography (PQC) [109] have shown promising outcomes in ensuring encryption resilience against advanced threats, such as quantum attacks [108]. Furthermore, to facilitate secure and efficient data transfer, some frameworks have implemented Complement Hashing Algorithms (CHA) [110] and lightweight authentication protocols [111] ,[112].

Yet, a review of the current literature indicates that while minimizing transmission delay and ensuring data confidentiality during exchange are the aims of the numerous models [113],

very few have explicitly verified the security of encryption keys during their generation and transmission [114]. This is a critical vulnerability since compromised encryption keys enable attackers to decrypt sensitive information even if the data is well-encrypted [115].

Albeit with notable limitations, recent approaches have addressed SDT in IoT-cloud frameworks. For instance, a model was introduced in [116] for secure data transfer employing public integrity verification procedures that included KeyGen, SigGen, GenProof, and VerifyProof mechanisms. Although low computational complexity was achieved by the model, it lacked provisions to verify whether the legitimate sender sent the received data. Likewise, in [117] the method proposed introduced a Fuzzy Dynamic Trust-centric Routing Protocol to enhance the security of low-power and lossy IoT networks. It did not ensure RT updates across the network nodes although this improved trust modeling within IoT-cloud communications, leading to data routing and validation inconsistencies. In [118] a Decisional Diffie-Hellman Inversion (DDHI) protocol was employed For secure file storage as well as AC. The approach was not compatible with heterogeneous environments despite effectively mitigating chosen ciphertext attacks, limiting its broader applicability. In another contribution [119] the authors implemented Elliptic Galois Cryptography (EGCrypto) to improve the embedding efficiency in SDT. The resulting framework suffered from a limited network lifetime although their method demonstrated strong encryption performance. In [120] quantum-enhanced SDT was explored using a visually meaningful cryptosystem based on quantum walks. The method ensured confidentiality and data integrity but introduced significant delays during encryption and decryption owing to algorithmic complexity. Similarly, in [121] a scalable routing mechanism based on pheromone-inspired Ant Colony Optimization was presented for fog computing environments. Despite enhanced routing decisions, this method failed to verify data authenticity post-decryption, undermining end-to-end security. In [122] the model incorporated a Decentralized Trust Management System (DTMS) along with an Efficient, Provably Secure Data Selection Sharing Scheme (EPSDSS). The method did not accommodate dynamic network conditions though effective in secured data selection, resulting in misleading data exchange among distributed nodes.

A more advanced framework [123] used a Secured Identity Access and Data Transmission Scheme (SIA-DTS) supported by three-factor authentication and the Diffie-Hellman key

exchange. The system introduced high complexity and latency under large-scale deployments while ensuring robust authentication. Also, asymmetric encryption and decryption were used in [124] to thwart man-in-the-middle attacks. Nevertheless, weak key management mechanisms permitted potential exploitation by third parties. Finally, a Merkle Tree-based model was adopted in [125] for tamper-proof data integrity checks. The system exhibited high energy consumption during data transfers across network nodes although data authenticity was improved. In summary, key security during transmission remains a largely unresolved issue while significant advancements have been made in cryptographic protections for cloud-assisted IoT environments. The current study addresses this gap by proposing a novel approach centered on QGE-HMAC and ELP-DNA, which collectively ensure encryption key authenticity, resilience against forgery, and lightweight authentication.

- i. Despite the growing interest in secure communication for IoT-cloud infrastructures, the following key challenges remain largely unaddressed:
 - ii. Existing models did not verify whether the encryption key was compromised or intercepted during transmission.
 - iii. Conventional works, such as [116] did not ascertain the data sender's authenticity.
 - iv. As per [117] several frameworks did not implement RT updates across all network nodes, leading to misrouting or erroneous data acceptance.
 - v. The majority of the existing protocols did not consider energy efficiency or clustering mechanisms, leading to reduced longevity of IoT deployments.
- i. The current research sets forth the following objectives to overcome these persistent challenges:
 - ii. To propose a QGE-HMAC to evaluate whether an encryption key has been tampered with during key sharing or transmission.
 - iii. To employ an ELP-DNA that ensures the received data is indeed from the intended and authenticated sender.

- iv. To construct a GCK-DAG for timely network topology updates, ensuring reliable data routing and reduced latency.

To improve IoT network lifetime through FCSDBSCAN, enhancing energy-efficient node clustering and optimizing network performance.

3.5.1 Proposed Scheme

During transmission, the security of encryption keys remains the most critical drawback in IoT-cloud environments. During the key-sharing phase, few studies have explored RT verification of encryption key integrity although prior research has addressed encryption strength, data confidentiality, and network resilience. To bridge this gap, the current research introduces a novel and robust framework for encryption key security-aware and reliable data transmission using a hybrid cryptographic strategy. To ensure secure transmission and improve network lifetime, immediate node updates, and authentication precision, the proposed system integrates advanced cryptographic and clustering algorithms.

The proposed approach is built upon a multi-phase architecture designed for IoT-enabled cloud environments. The architecture comprises advanced mechanisms, such as QGE-HMAC and ELP-DSA, which collectively ensure end-to-end data integrity and secure key exchange. The cryptographic operations are supported by a layered optimization framework that utilizes GCK-DAG and FCSDBSCAN for network reliability and lifetime enhancement (Figure 3.10) exhibits the full architectural flow of the proposed system.

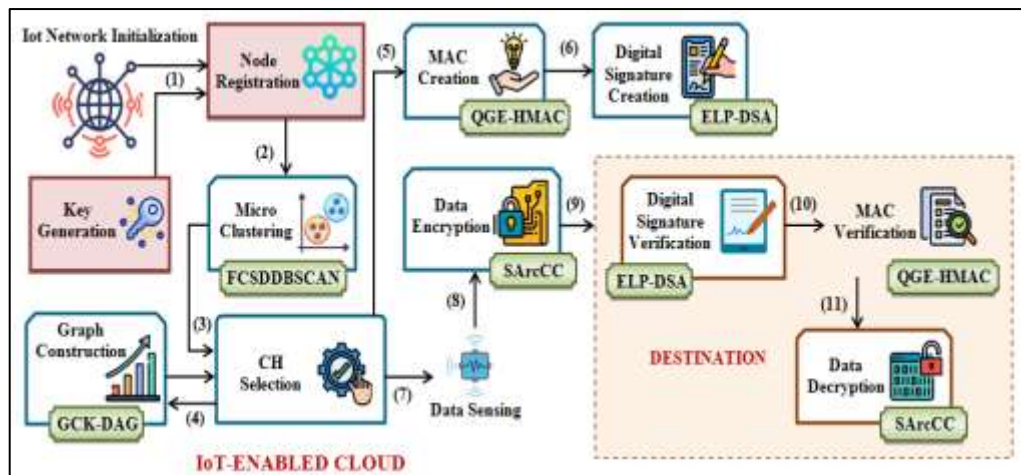


Figure 3.10: Proposed Approach's Block Diagram.

3.5.1.1 IoT network initialization and registration

The first step in beginning a secure and intelligent cloud-assisted IoT system involves initializing the network environment, which includes setting up foundational components, such as communication protocols, gateway and node configuration, and implementing secure connectivity mechanisms. These steps ensure that the IoT infrastructure is resilient, scalable, and able to handle secure data exchange.

Following network initialization, each IoT node is systematically registered into the network. This process comprises validating and inputting critical credentials, including unique device identifiers (Device IDs), digital certificates, and authentication keys. These credentials are the foundational elements to verify the legitimacy and origin of data sources within the network as equation 3.10.

$$\mathfrak{I}_t = \langle \mathfrak{I}_1, \mathfrak{I}_2, \dots, \mathfrak{I}_T \rangle \text{ Here, } t = 1 \text{ to } T \quad (3.10)$$

Thus, for the (T) number of registered IoT nodes $(\mathfrak{I}_t)$, keys like PRK and PK are generated.

3.5.1.2 Micro clustering

Following the IoT nodes' registration and their corresponding cryptographic key generation, the system proceeds to the micro-clustering phase. In this stage, using a proposed FCSDDSCAN algorithm, the network of registered IoT devices is organized into localized groups or clusters. This advanced clustering method is an enhancement of the conventional DBSCAN, which has shown drawbacks in real-world IoT scenarios due to its sensitivity to input parameters, such as epsilon (ϵ) and MinPts (Minimum Points).

DBSCAN operates by grouping spatially proximate data points (IoT devices), which exhibit similar attributes in these cases. It suffers from low clustering efficiency in high-density, heterogeneous IoT environments while it effectively isolates noise and identifies arbitrary-shaped clusters. Due to suboptimal parameter selection, issues, such as over-clustering, under-clustering, and inaccurate merging of closely related clusters primarily arise. To address these limitations, the proposed FCSDDSCAN algorithm introduces two additional evaluation mechanisms, the Silhouette Dunn Index (SD) and Fleiss Cohen's Agreement Score (FC). These techniques dynamically optimize the clustering parameters, ensuring

higher precision in identifying core clusters and mitigating overfitting and underfitting data points.

Further, the $(\mathfrak{I}_t)$ is subjected to the proposed FCSDDDBSCAN that performs micro clustering. The DBSCAN considerably groups the nodes in the larger network, which means grouping together the small, tightly packed sets of IoT devices based on their proximity and data characteristics. This effectively recognizes small, localized clusters within a larger network. Nevertheless, due to the improper epsilon (λ) and MinPts (ϕ) , the DBSCAN has low clustering efficiency. Hence, the proposed method introduces Silhouette Dunn (SD) and Fleiss Cohen's (FC) techniques. Here, the cluster merging and the over and under-clustering are evaded. The neighborhood's radius around a point (epsilon) is calculated as equation 3.11.

$$\lambda = \varpi \cdot \left(\frac{1}{T} \sum_{t=1}^T \mathfrak{I}_t \right) + (1 - \varpi) \cdot \frac{\min_{1 \leq t \neq x \leq Y} \partial(\mathfrak{I}_t, \mathfrak{I}_x)}{\max_{1 \leq t \leq Y} \Theta(\mathfrak{I}_t)} \quad (3.11)$$

Here, (ϖ) signifies the weight factor, (x) represents the parameter, the distance is represented as (∂) , the number of clusters is indicated as (Y) , and the cluster diameter is denoted as (Θ) . Likewise, the minimum number of points needed to form a cluster (MinPts) is calculated in equation 3.12.

$$\phi = \varpi_1 \cdot \left(\frac{\alpha_o - \alpha_e}{1 - \alpha_e} \right) + \varpi_2 \cdot \left(\frac{\alpha_o - \alpha_e}{1 - \alpha_e} \right) \quad (3.12)$$

Here, the observed and expected agreements are indicated as (α_o) and (α_e) , correspondingly. For each data point, the points within the epsilon radius are recognized. Subsequently, the core point is selected by considering the number of neighbors more than or equivalent to MinPts. As per the core point, the clustering is carried out. Here, the clustered network is mentioned as (Ω) .

3.5.1.3 CH selection

After forming micro clusters among the registered IoT nodes, the next critical step is recognizing an optimal Cluster Head (CH) for every single cluster. The CH is pivotal in aggregating, managing, and forwarding information from sensor nodes to the cloud, acting as an energy-efficient gateway. In this research, the CH is selected using the Bobcat Optimization Algorithm (BOA)—a nature-inspired metaheuristic based on the tracking and hunting behavior of bobcats in the wild.

The BOA is selected due to its dual-phase balancing capability, exploration, and exploitation, which efficiently solves complex optimization tasks like CH selection by balancing global and local search strategies. The CH is selected from Ω by employing the BOA. The BOA simulates the hunting behavior of bobcats in the wild. It also efficiently resolves complex issues due to its balancing capability. Here, the nodes (members) in the Ω are assumed as the bobcat. Primarily, in the search space, the $q = 1 \text{ to } Q$ number of bobcats ($\Omega_{q,p}$) is initialized regarding the lower bound (ℓw_p) and upper bound (μp_p) as in equation 3.13.

$$\Omega_{q,p} = \ell w_p + \Re * (\mu p_p - \ell w_p) \quad (3.13)$$

Here, the number of problem variables is signified as $(p = 1, 2, \dots, P)$, and $(\Re)$ indicates the random value. Then, the fitness value (ψ) is defined as the minimum distance and maximum energy. It helps to identify the optimal solution $(\chi \Omega_{q,p})$. The bobcat's position is updated concerning the tracking and movement behavior of the bobcat toward the prey in the exploration equation 3.14 and equation 3.15.

$$\Omega_{q,p}^x = \Omega_{q,p} + (1 - 2\Re) * (\chi \Omega_{q,p} - \xi_{q,p} \times \Omega_{q,p}) \quad (3.14)$$

$$\Omega_{q,p} = \begin{cases} \Omega_{q,p}^x, & \text{If } (\psi^x \leq \psi) \\ \Omega_{q,p}, & \text{else} \end{cases} \quad (3.15)$$

In the exploitation phase, the bobcat's position is updated centered on the chase simulation betwixt the bobcat and the prey 3.16 and equation 3.17.

$$\Omega_{q,p}^Y = \Omega_{q,p} + \frac{1-2\Re}{1+\iota} * \Omega_{q,p} \quad (3.16)$$

$$\Omega_{q,p} = \begin{cases} \Omega_{q,p}^Y, & \text{If } (\psi^Y \leq \psi) \\ \Omega_{q,p}, & \text{else} \end{cases} \quad (3.17)$$

Where, $(\Omega_{q,p}^X)$ and (ψ^X) show the updated bobcat's position and fitness in the exploration, respectively, $(\Omega_{q,p}^Y)$ and (ψ^Y) represent the updated bobcat's position and fitness in the exploitation, correspondingly, $(\xi_{q,p})$ denotes the random value from the list $\{1, 2\}$, and the selected CH is shown as (Φ^{CH}) .

3.5.1.4 Graph construction

Synchronized communication and instantaneous updates across the dynamic network become essential once the optimal CHs are identified for each micro cluster. A graph-based model manages data flow among IoT devices and their corresponding CHs. To create a scalable, loop-free network routing framework, the proposed model uses a Directed Acyclic Graph (DAG) structure enhanced by the Gini Coffman Kahn (GCK) technique.

A DAG is a topological construct consisting of nodes (vertices) and directional edges representing unidirectional data flow between devices. By definition, a DAG has no cycles, meaning there is no path in which a node can revisit itself. This property is critical in IoT-cloud communication in which data loops can result in redundancy, delay, or communication failure.

The DAG model facilitates efficient RT data propagation, dependency resolution, and task scheduling, particularly when interdependent nodes operate in heterogeneous network conditions. However, the computational cost of building and maintaining the DAG becomes non-trivial as the network scales (i.e., with increasing nodes and edges).

- i. To address this limitation, the proposed scheme introduces a GCK-DAG that integrates fairness-based ordering and indegree-based optimization to reduce the complexity of graph updates. This method ensures that all topological alterations, such as the removal

or addition of IoT nodes, are immediately reflected in the network graph without violating acyclicity or dependency constraints.

- ii. Nodes (ϕ^*): Represent IoT devices, CHs, or key communication units.
- iii. Edges ($\rightarrow$): Represent directed communication links from one node to another.
- iv. In-degree: The number of incoming edges to a node utilized to identify root nodes.
- v. Out-degree: The number of outgoing edges from a node used for tracing data flow.

Using Kahn's algorithm, the topological sort of the graph is constructed, where nodes with zero in-degree are inserted into a queue and processed in order of dependency. At each step, the node is removed, its edges are deleted, and the in-degree of adjacent nodes are updated. If any adjacent node reaches zero in degrees, then it is added to the queue. This process continues until all nodes are sorted or the graph becomes cyclic (in which case it is rejected).

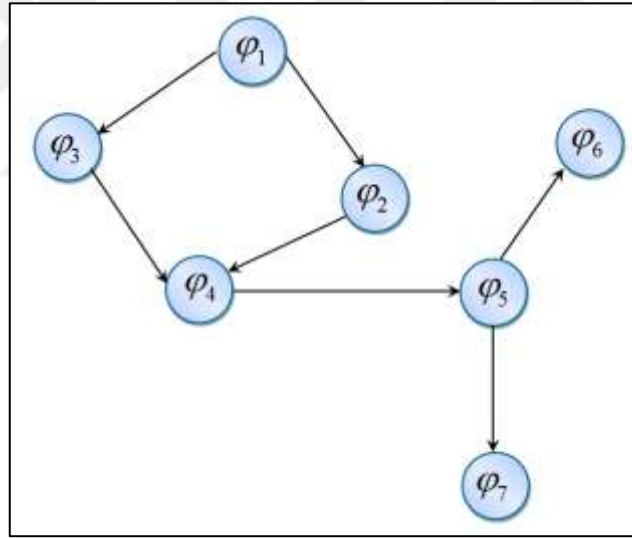


Figure 3.11: A Typical GCK-DAG With 7 Nodes.

The above (Figure 3.11) exemplifies a typical GCK-DAG with 7 nodes.

- i. $\phi_1 \rightarrow \phi_2 \rightarrow \phi_5 \rightarrow \phi_7$
- ii. $\phi_1 \rightarrow \phi_3 \rightarrow \phi_4 \rightarrow \phi_5$
- iii. $\phi_2 \rightarrow \phi_4, \phi_6 \rightarrow \phi_5$

This configuration exhibits multiple input paths converging on node ϕ_5 , which acts as a central processing unit or aggregator. The graph enforces acyclicity by allowing data to flow only downstream, ensuring deterministic task execution, collision-free communication, and logical ordering.

The edges (χ_{edge}) signify the links between the $(k=1toK)$ numbers of IoT nodes (φ_k) see equation 3.18 and equation 3.19.

$$\varphi_k = |\varphi_1, \varphi_2, \dots, \varphi_K| \text{ Here, } k = 1toK \quad (3.18)$$

$$\chi_{edge} = \|\langle \varphi_1, \varphi_2 \rangle, \langle \varphi_1, \varphi_3 \rangle, \langle \varphi_2, \varphi_4 \rangle, \langle \varphi_3, \varphi_4 \rangle, \langle \varphi_4, \varphi_5 \rangle, \langle \varphi_5, \varphi_6 \rangle, \langle \varphi_5, \varphi_7 \rangle\| \quad (3.19)$$

Regarding topological order (linear order), the vertex (φ_1) has to appear before (φ_3) for the directed edge $\langle \varphi_1, \varphi_3 \rangle$. Therefore, the topological order is acquired via Gini Coffman Kahn's technique. Here, the in-degree is assigned for all the vertexes. Thereafter, the vertex with zero in-degree ($in-deg$) is added to the queue (Qu) and then removed from the graph. Next, by considering the low in-degree, each node is inserted into the queue. It is articulated as follows equation 3.20.

$$Qu = \{k \in K \mid in-deg(\varphi_k) = 0\} \& in-deg(k) \leftarrow in-deg(k) - 1 \quad (3.20)$$

Here, the topological sequence (Ξ) is expressed as follows in equation 3.21.

$$\Xi = \langle \varphi_1 \rightarrow \varphi_3 \rightarrow \varphi_2 \rightarrow \varphi_4 \rightarrow \varphi_5 \rightarrow \varphi_6 \rightarrow \varphi_7 \rangle \quad (3.21)$$

Similarly, the adjacency matrix (Λ) assists in highlighting the dependency (τ) among the IoT nodes as see in equation 3.22.

$$\Lambda = \begin{bmatrix} & \varphi_1 & \varphi_2 & \varphi_3 & \varphi_4 & \varphi_5 & \varphi_6 & \varphi_7 \\ \varphi_1 & 0 & \tau_1 & \tau_2 & 0 & 0 & 0 & 0 \\ \varphi_2 & 0 & 0 & 0 & \tau_3 & 0 & 0 & 0 \\ \varphi_3 & 0 & 0 & 0 & \tau_4 & 0 & 0 & 0 \\ \varphi_4 & 0 & 0 & 0 & 0 & \tau_5 & 0 & 0 \\ \varphi_5 & 0 & 0 & 0 & 0 & 0 & \tau_6 & \tau_7 \\ \varphi_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \varphi_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.22)$$

At last, the constructed graph is represented as (Z_{graph}) .

Advantages of GCK-DAG:

- i. Instant reflection of topology changes without recomputation of the full network graph.
- ii. Guarantees non-redundant, loop-free communication flow.
- iii. Supports task execution order based on dependency levels.
- iv. Facilitates intelligent path selection by minimizing redundant hops and computational Bottlenecks (BNs).

In summary, the GCK-DAG not only enhances the scalability and responsiveness of the network but also ensures that IoT data communication remains robust and context-aware, even under dynamic node changes and partial system failures.

3.5.1.5 MAC creation

In modern IoT cloud systems, MAC is a critical cryptographic component utilized for validating the integrity and authenticity of transmitted data. Nevertheless, though broadly utilized, conventional HMAC approaches are susceptible to certain vulnerabilities, particularly those associated with SK leakage during transmission or storage. A compromised SK can result in erroneous MAC generation, enabling malicious actors to alter or forge messages without detection.

For overcoming these drawbacks, this research proposes a novel framework integrating Quantum Gibbs Entropy (QGE) with HMAC, forming a robust framework denoted as QGE-HMAC. This hybrid model optimizes the secrecy of the MAC generation process by

dynamically generating the SK based on quantum entropy measures, increasing resistance to cryptanalytic attacks.

The MAC is created from the receiver's PK (encryption key) (Pub) of the nodes in Ω using the proposed QGE-HMAC to prevent the encryption key attack. The HMAC offers a high authentication level. It wields the SK and HVs to preserve the encryption key. Nevertheless, the SK is vulnerable to attacks, causing improper creation of MAC. Therefore, for generating the SK (δ^{key}) , the QGE is employed. Similar to traditional HMAC but fortified with QGE principles, the MAC is further generated by hashing the combination of this entropy-derived SK and the PK of the receiver using inner and outer paddings.

The process of creating MAC is described as equation 3.23 and equation 3.24.

$$\delta^{key} = \Gamma_{race}(\partial \mathfrak{S} \ln \partial \mathfrak{S}) \quad (3.23)$$

$$\Delta_{MAC} = \kappa((\delta^{key} \oplus \wp_{out}) \parallel \kappa((\delta^{key} \oplus \wp_{in}) \parallel Pub)) \quad (3.24)$$

Here, (Γ_{race}) notates the trace, (κ) represents the hash function, $(\wp_{out})$ implies the outer padding, $(\wp_{in})$ signifies the inner padding, and $(\partial \mathfrak{S})$ indicates the density matrix. Thus, the created MAC is represented as (Δ_{MAC}) . Perversely, the MAC is recomputed on the receiver side and then verified by contrasting it with the (Δ_{MAC}) .

This QGE-HMAC scheme renders multiple benefits:

- i. It reduces key reuse vulnerabilities through entropy-based SK generation.
- ii. It binds the MAC to the recipient's identity, preventing man-in-the-middle attacks.
- iii. It leverages quantum entropy to introduce randomness beyond conventional PRNG-based key generation.

3.5.1.6 Digital signature creation

Digital signatures ensure data authenticity, integrity, and non-repudiation in IoT-enabled cloud networks during transmission. A digital signature validates that the data has originated

from a legitimate sender and is unaltered in transit. For this purpose, the current research introduces an improved signature generation framework using the ELP-DSA.

Traditional DSA are known for their computational efficiency in generating multiple digital signatures rapidly. Yet, standard DSA implementations often suffer from hash function inefficiencies, leading to potential hash collisions and ultimately causing information loss or degraded verification reliability.

To mitigate these issues, the proposed ELP-DSA incorporates Exponential Linear Probing (ELP) with the hashing stage of DSA. ELP overcomes collisions by dynamically adjusting the hash position using a non-linear exponential increment, enhancing the uniform distribution of HVs and maintaining signature uniqueness across diverse inputs.

Here, from (Δ_{MAC}) , the digital signature and the user's credentials are created using the proposed ELP-DSA. DSA, which is relatively faster in generating a large number of digital signatures, is used for signature creation. Nevertheless, owing to the improper hash function, DSA has information loss. Hence, the proposed method employs ELP that resolves the hash collision by adding arbitrary quadratic polynomials into the hash on DSA. The process of ELP-DSA is expressed as follow equation 3.25 and equation 3.26.

Primarily, the prime number (A_{num}) and the prime divisor (B_{div}) are selected to generate the key. Furthermore, the generator $(\aleph^\circ)$ is chosen.

$$\|A_{num}, B_{div}\| \xrightarrow{\text{initialize}} |1024 \text{ bits}, 160 \text{ bits}| \quad (3.25)$$

$$\aleph^\circ = E^{(A_{num}-1)/B_{div}} \bmod A_{num} \quad (3.26)$$

Here, E indicates the integer. After that, the PRK (σ_{pv}) and the PK (Jh) are generated as equation 3.27 and equation 3.28.

$$\sigma_{pv} = |0 < \sigma_{pv} < B_{div}| \quad (3.27)$$

$$Jh = \aleph^\circ \bmod A_{num} \quad (3.28)$$

Next, the ELP is used to perform the hash function (HJ) , generating the HV (O) of the (Δ_{MAC}) see equation 3.29.

$$HJ[(\sigma_{pv}, Jh), L] = \{HJ(\sigma_{pv}, Jh) + Inc\} \bmod Sx \quad (3.29)$$

Here, (L) exemplifies the number of probes, (Inc) indicates the exponential increment function, and (Sx) signifies the size of the hash table as in equation 3.30, equation 3.31 and equation 3.32.

$$x = (\aleph^{\circ} \bmod A_{num}) \bmod B_{div} \quad (3.30)$$

$$y = Rd^{-1}(O + \sigma_{pv} \cdot x) \bmod B_{div} \quad (3.31)$$

$$D_{sig} = (x, y) * \Delta_{MAC} \quad (3.32)$$

Here, (D_{sig}) specifies the digital signature, the parameters are indicated as (x, y) , and (Rd^{-1}) denotes the random integer. When $((0 < x < B_{div}) \& \& (0 < y < B_{div}))$ is true, the signature is considered as valid; or else, the signature is regarded as invalid equation 3.33, equation 3.34, equation 3.35 and equation 3.36.

$$F_1 = y^{-1} \bmod B_{div} \quad (3.33)$$

$$F_2 = (O\Delta_{MAC} \cdot F_1) \bmod B_{div} \quad (3.34)$$

$$F_3 = (x \cdot F_1) \bmod B_{div} \quad (3.35)$$

$$F_4 = (\aleph^{\circ} \bmod A_{num}) \bmod B_{div} \quad (3.36)$$

Here, (F_1) , (F_2) , (F_3) , and (F_4) notate the signature parameters. If (F_4) is equal to (x) , then the signature is valid. Or else, the signature is invalid.

The ELP-DSA's pseudo code is rendered in (Figure 3.12).

Input: $MAC(\Delta_{MAC})$
Output: Digital signature (D_{sig})

Begin
Initialize $(Jh), (\sigma_{pv})$ and (Δ_{MAC})
For 1 to each node,
 Select $\|A_{num}, B_{div}, N^o\|$
 #key generation
 $\sigma_{pv} = |0 < \sigma_{pv} < B_{div}|$
 $Jh = N^o \bmod A_{num}$
 Generate hash value,
 $HJ(\sigma_{pv}, Jh, L) = (HJ(\sigma_{pv}, Jh) + Inc) \bmod Sx$
 Create digital signature
 $D_{sig} = (x, y) * \Delta_{MAC}$
 #signature verification
 Compute $F_4 = (N^o \bmod A_{num}) \bmod B_{div}$
 IF $(F_4 == x)$
 {
 Valid
 }
 Else
 {
 Invalid
 }
 End IF
End For
Return
End

Figure 3.12: Pseudocode for ELP-DSA.

Therefore, the (D_{sig}) is sent to the destination. This signature is further attached to the encrypted data and transmitted to the destination. The ELP-DSA renders resilient digital authentication and considerably mitigates the risk of hash collision, making it appropriate for comprehensive IoT applications requiring robust, lightweight cryptographic signatures.

3.5.1.7 Data Sensing

Once the digital signature is generated and authenticated successfully, the system proceeds with the data acquisition phase, wherein RT environmental or application-specific data is sensed using embedded IoT sensor devices. These sensors, distributed across the IoT-cloud framework, collect raw data from their respective environments—ranging from physical

measurements, including temperature, humidity, and gas concentration to industrial control parameters in smart facilities. Sensor-based data relevant to secure IoT communication is modeled in this study using publicly available benchmark datasets. These datasets act as validated testbeds for simulating RT IoT sensing conditions and assessing the robustness of the proposed cryptographic and routing framework.

i. Water Treatment Plant Dataset

This dataset, hosted on the UCI ML Repository, contains time-series measurements from a real-world water treatment facility. It captures operational variables, namely chemical concentrations, inflow/outflow rates, and quality control indicators, demonstrating typical data types encountered in industrial IoT scenarios. Dataset link:

<https://archive.ics.uci.edu/dataset/106/water+treatment+plant> [126].

ii. Gas Sensor Array Drift Dataset

This dataset encompasses readings from gas sensors' arrays that are bare to diverse chemical agents over extended periods, showing how sensor behavior can drift over time. It is highly pertinent for testing anomaly detection and long-term data authentication strategies in dynamic IoT environments. Dataset link:

<https://archive.ics.uci.edu/dataset/224/gas+sensor+array+drift+dataset> [127].

For the subsequent encryption phase, the sensed data, whether simulated or collected from RT streaming interfaces, forms the input. These readings are not directly transmitted but are encrypted and signed utilizing the previously described cryptographic mechanisms (QGE-HMAC and ELP-DSA), ensuring the transmitted content's integrity, authenticity, and confidentiality. This process produces a secure sensing-to-transmission pipeline, laying the foundation for a reliable and resilient IoT-cloud data flow system that is robust against interception, tampering, or spoofing attacks.

3.5.1.8 Data encryption

Once the raw data is sensed from the IoT environment, it must be securely encrypted before being transmitted for ensuring data confidentiality and resistance to interception. Conventional methods like ECC are widely used in lightweight cryptographic applications

owing to their capability of providing high levels of security with small key sizes. ECC operates on ECs defined over finite fields and ensures robust security by resolving the EC Discrete Logarithm Problem (ECDLP), which is computationally infeasible.

However, ECC introduces substantial computational complexity, particularly during the scalar multiplication step, which is worst for resource-restricted IoT devices. Hence, the present study proposes an improved curve-based encryption method, termed Sphere Arctangent Curve Cryptography (SArcCC). This algorithm replaces the conventional EC with a sphere-based arctangent curve model, thereby diminishing the cryptographic load while preserving high entropy in key generation and encryption. Therefore, for solving the encryption system's complexity, the proposed method establishes the sphere arctangent curve $(\delta\alpha)$. The data encryption process is described as equation 3.37.

$$\delta\alpha = \partial data * \arctan(\partial data) - \frac{1}{2} \ln \left(1 + \sum_{n=1}^N \partial data^2 \right) + Cst \quad (3.37)$$

Here, $(n=1toN)$ implies the iteration, and (Cst) represents the constant. To perform encryption and decryption, the (Pub) and PRK (Mpv) are generated as equation 3.38

$$Pub = Mpv * \rho(\delta\alpha); \quad (1 \leq Mpv \leq r' - 1) \quad (3.38)$$

Where, (ρ) indicates the curve parameter, and (r') represents the maximum range. Here, via the two CTs (W_1, W_2) , the data is encrypted equation 3.39 and equation 3.40.

$$W_1 = rd^\infty * \rho \quad (3.39)$$

$$W_2 = \partial data + rd^\infty \times Pub \quad (3.40)$$

Where, (rd^∞) notates the random value. Contrarily, the data is decrypted utilizing the below expression equation 3.41.

$$\partial data = W_2 - Mpv * W_1 \quad (3.41)$$

Therefore, the encrypted data is symbolized as $(\partial data_{en})$. The SArcCC's pseudo-code is given in (Figure 3. 13).

Input: IoT data $(\partial data)$
Output: Encrypted data $(\partial data_{en})$

Begin
Initialize $(\partial data)$, (Pub) and (Mpv)
For 1 to each $(\partial data)$ do,
 Apply sphere arctangent curve,

$$\delta \alpha = \partial data \arctan(\partial data) - \frac{1}{2} \ln \left(1 + \sum_{n=1}^N \partial data^2 \right) + Cst$$

 Perform key generation,
 $Pub = Mpv * \rho(\delta \alpha); \quad (1 \leq Mpv \leq r' - 1)$
 Encrypt data,
 $W_1 = rd_{\infty} * \rho$
 $W_2 = \partial data + rd_{\infty} \times Pub$
 Execute decryption $\partial data = W_2 - Mpv * W_1$
End For
Return $(\partial data_{en})$
End

Figure 3.13: Pseudocode of SArcCC.

Subsequently, the $(\partial data_{en})$ is transmitted to the destination.

3.5.1.9 Digital signature verification

For validating the sending node's authenticity, the system initiates the digital signature verification phase once the encrypted data is received at the receiver's end. This process ensures that the data is generated from a legitimate and trusted source and has not been tampered with during transmission. The verification mechanism is centered on the ELP-DNA, as detailed in Section 3.5.1.6.

Upon receipt, the destination node extracts the digital signature pair transmitted with the encrypted message. The message hash is recalculated by employing the sender's PK using the same ELP-based hashing function employed during signature generation. Then, the system performs the following verification steps:

- i. Recompute the signature verification component
- ii. Calculate the expected

- iii. Compare the recomputed with the original

This step provides non-repudiation, signifying that the sender can't deny the sent message. It also guarantees that the message is unaltered in transit, preserving data integrity. Using the QGE-HMAC mechanism described in Section 3.5.1.10, the system can safely verify the encryption key's integrity if the digital signature is valid.

The proposed framework makes sure that authenticated nodes only can communicate within the IoT-cloud environment by integrating ELP-DSA at the verification phase, thereby maintaining a trustworthy SDT infrastructure.

3.5.1.10 MAC verification

Following successful digital signature verification, the system advances to the next critical security step, MAC verification, to ensure the authenticity and integrity of the encryption key used during data transmission. This step is vital to confirm that the receiver's PK is not tampered with or substituted during the process of key exchange. Using the QGE-HMAC approach, the verification is carried out, as introduced in Section 3.5.1.5.

MACs are generated in conventional cryptographic systems using fixed SKs and standard hash functions. Nevertheless, the SK can become a vulnerable point in such approaches, potentially compromising the entire authentication mechanism. To address this, QGE-HMAC dynamically derives the SK using quantum entropy principles, increasing randomness and significantly enhancing key security. By introducing entropy-aware MAC verification, the QGE-HMAC mechanism strengthens the overall authentication chain. QGE-HMAC protects against encryption key compromise, which is often exploited in cyber-attacks targeting IoT cloud infrastructures. This process ensures end-to-end trust, even in highly dynamic and distributed systems.

By combining:

- i. QGE-HMAC,
- ii. ELP-DSA, and
- iii. SArcCC.

The proposed scheme ensures that the communication pipeline is robust against cyber threats, such as key substitution attacks, data tampering, impersonation, and eavesdropping.

Hence, the decryption phase affirms that the IoT data has been securely and authentically transmitted, completing the end-to-end encryption framework within the cloud-assisted IoT environment.

3.5.1.11 Data decryption

Data decryption is the concluding phase of the proposed SDT framework, which is executed at the receiver's end upon successfully validating message authenticity and encryption key integrity. This step is pivotal to ensuring that the data received is confidential and untampered, upholding the core security principles of confidentiality, integrity, and authenticity.

Prior to decryption, the encrypted data undergoes two mandatory validation stages:

- i. As described in Section 3.5.1.6, digital signature verification ensures that the data originates from a legitimate and trusted node using ELP-DSA.
- ii. As elaborated in Section 3.5.1.10, MAC verification confirms that the encryption key has not been altered during transmission using QGE-HMAC.

After the successful completion of these security validations, the system proceeds to decrypt the received ciphertext utilizing the PRK of the recipient.

Decryption via SArcCC

As introduced in Section 3.5.1.8, the decryption algorithm is powered by the proposed SArcCC. This cryptographic scheme is specially designed to provide robust security with reduced computational overhead—ideal for IoT environments, where devices have limited processing resources. Thus, the proposed encryption-decryption framework renders end-to-end SDT and assurances lightweight, scalable, and RT applicability for cloud-assisted IoT networks.

3.5.2 Summary

Here, the detailed methodology of the proposed SDT framework is presented for cloud-assisted IoT networks, highlighting hybrid encryption and authentication mechanisms. The process began with IoT network initialization and node registration, followed by key generation utilizing quantum entropy and public-private key cryptography. By utilizing the FCSDDBSCAN algorithm, microclustering of nodes was carried out for improving energy efficiency as well as network lifetime. Subsequently, to identify optimal CHs, the BOA was employed. To manage network dynamics and task dependencies, a GCK-DAG was constructed. To generate secure MACs, the proposed QGE-HMAC was introduced. At the same time, using ELP-DSA, digital signatures were generated and verified. Sensed data was encrypted through the lightweight SArcCC and transmitted alongside the digital signature and MAC. Upon reception, the system carried out digital signature verification and MAC validation to ensure authenticity and key integrity. At last, utilizing the receiver's PRK generated via the SArcCC mechanism, the original data was decrypted. The approach ensured end-to-end security, authentication, and confidentiality, tailored for modern IoT-cloud environments' computational constraints and scalability demands.

3.6 PROPOSED RELIABLE CLOUD-ASSISTED IOT NETWORK MONITORING

The development of cloud-assisted IoT environments has transformed how modern digital ecosystems operate. These environments have arisen as a promising domain for research and practical deployment due to their ability to interconnect numerous devices for diverse RT applications across HC, smart cities, environmental monitoring, and industrial automation [128]. The IoT paradigm is a huge network of uniquely identifiable, interlinked smart devices able to collect, exchange, and process data without human intervention [129]. This inter-device communication fosters the growth of intelligent systems that can react to environmental stimuli, perform analysis, and trigger autonomous actions.

CC complements the IoT infrastructure by proposing scalable, on-demand access to computing resources and massive storage abilities, enabling remote access and data-driven DM processes [130]. Along with the handling of the larger volume of data generated by IoT devices, cloud platforms also support computationally intensive services, such as analytics, ML, and RT monitoring [131] Despite these advantages, several critical security challenges

are introduced by integrating cloud platforms into IoT environments. In particular, cloud-based IoT systems suffer from vulnerabilities related to data confidentiality, unauthorized access, and system intrusions, largely owing to the extended attack surface and decentralized nature of connected devices [132]. These concerns are further exacerbated by insufficient standardization in data management protocols across diverse cloud service providers [133].

The primary security requirement for IoT systems is device authentication, which ensures that only legitimate devices can communicate within the network. This process is pivotal to preventing impersonation attacks and unauthorized data access, enhancing the IoT infrastructure's overall security and efficiency [134]. Data authenticity is another critical aspect, which is the assurance that the data received or stored isn't altered or manipulated during transmission or storage. Ensuring authenticity reinforces trust in system operations and safeguards against man-in-the-middle attacks and data injection threats [135].

Researchers have turned to lightweight cryptographic primitives to achieve such security goals without overburdening the resource-constrained IoT nodes, which offer an optimal balance between computational complexity and cryptographic strength [136]. Along with encryption mechanisms, IDS are vital in identifying anomalies and malicious behavior within cloud-assisted IoT networks. For RT intrusion detection, existing studies have incorporated artificial intelligence and deep learning methods, such as Recurrent Neural Networks (RNN), Gated Recurrent Units (GRU), and Long Short-Term Memory (LSTM) networks [137]. Nevertheless, even though these approaches exhibit promise in detecting attacks, they often lack scalability and adaptability when deployed over long-term IoT monitoring infrastructures.

Furthermore, numerous prevailing frameworks did not incorporate data authentication with RT network monitoring or maintain historical integrity for forensic or analytical use. Therefore, there is a gap in the development of unified solutions supporting continuous, authenticated, and secure monitoring of IoT devices in cloud-assisted architectures. To overcome these limitations, a comprehensive and reliable cloud-assisted IoT network monitoring (IoT-NM) system that combines hybrid encryption mechanisms with an LCA-BST structure for intelligent data authentication and anomaly detection is proposed in this research. The approach introduces an end-to-end security model supported by optimized

device tracking, RT monitoring, and lightweight cryptographic verification, making it appropriate for larger-scale, dynamic IoT environments.

Despite the exponential growth of cloud-assisted IoT ecosystems, the majority of the current frameworks face substantial limitations in ensuring reliable, secure, and authenticated network monitoring. Existing methodologies failed to render long-term, traceable data authentication mechanisms essential for forensic analysis, regulatory compliance, and trustworthy system behavior. A key issue is insufficient robust integration between device-level security mechanisms and network monitoring tools. Without this integration, IoT systems remain susceptible to identity spoofing, unauthorized access, and undetected tampering.

For instance, many traditional systems fail to monitor Outdated Devices (ODs) that are no longer compliant with updated security policies or display unusual patterns owing to obsolescence. This rigorously impacts network reliability and leaves the system susceptible to exploitation [138]. In addition, as the number of linked IoT devices continues to develop, the volume of data traffic has increased dramatically. Some current approaches can't efficiently handle high-throughput data streams, leading to network collisions and performance degradation [139].

Another critical concern lies in the data integrity at the time of transmission. Traditional models often cannot detect or prevent subtle data alteration, including replay attacks or stealthy injection of malicious information. The inability to trace or authenticate transmitted data compromises system integrity and raises concerns in sensitive applications, such as HC, critical infrastructure, and industrial automation [140].

Most notably, the long-term operational security of IoT-cloud environments is limited by the lack of a unified, lightweight, and secure network monitoring framework that combines RT intrusion detection, device authentication, and data integrity verification. As such, there is a pressing need for a scalable, intelligent, and hybrid cryptographic model that can address these challenges holistically. In response to the limitations mentioned above, this study intends to design and develop a hybrid, lightweight, and intelligent cloud-assisted IoT Network Monitoring (IoT-NM) system, which ensures SDT, RT device authentication, and

efficient anomaly detection. The proposed research's core objectives are expressed as follows:

- i. To develop a robust IoT Network Monitoring tool centered on an LCA-BST architecture, which enables dynamic device status monitoring and supports secure data authentication for RT and historical analysis.
- ii. To employ an intelligent device validation mechanism using Hyperbolic Tangent Trapezoidal Fuzzy (HTT-Fuzzy) logic, permitting for accurate identification of ODs in the network and enabling proactive alert generation to system administrators.
- iii. To ensure computational efficiency and load balancing by introducing a novel QASOA, which distributes data processing tasks across nodes to diminish latency and avoid congestion in resource-constrained networks.
- iv. To integrate hybrid encryption through CPJTABE-THKSCC, which offers lightweight yet highly secure data encryption, mitigating risks of forgery and unauthorized access in multi-user cloud-assisted IoT environments.
- v. Together, the basis for a comprehensive and scalable security framework that addresses next-generation cloud-assisted IoT systems' authentication, reliability, and efficiency needs is formed by the above-mentioned objectives.

3.6.1 Proposed Methodology

A comprehensive and intelligent cloud-assisted IoT-NM framework that integrates hybrid encryption, dynamic authentication, lightweight anomaly detection, and RT device monitoring is introduced in the proposed research. To address prevailing limitations in data security, identity verification, intrusion detection, and load balancing in large-scale IoT environments, this architecture is designed. While enabling efficient and scalable monitoring across heterogeneous IoT nodes, the core objective is to ensure end-to-end data authentication security. (Figure 3.14) illustrates the proposed methodology's complete design, outlining each functional component and its role in achieving secure and reliable communication.

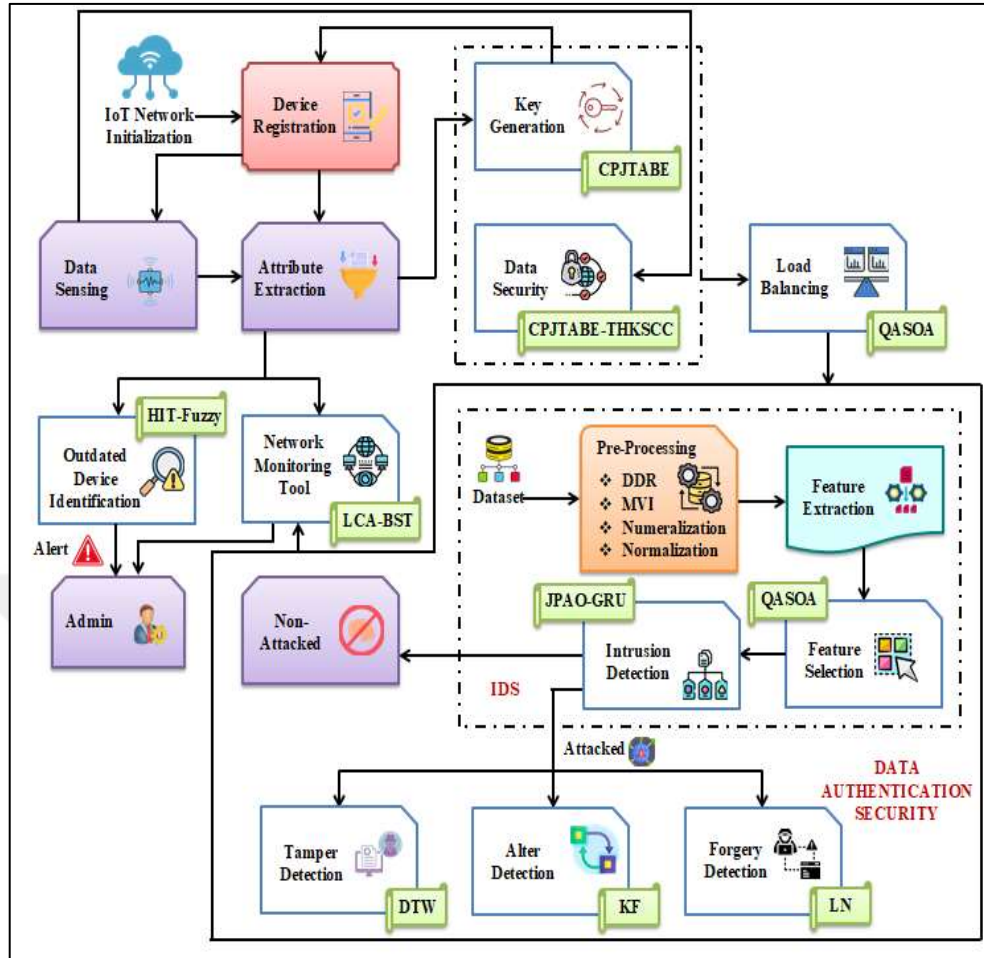


Figure 3.14: Research Methodology's Design.

As illustrated in (Figure 3.14) the proposed methodology presents a multi-layered architecture tailored for ensuring secure, efficient, and intelligent network monitoring within a cloud-assisted IoT environment. The process begins with IoT network initialization and device registration in which each node is uniquely identified and profiled via attribute extraction. Subsequently, these attributes are employed for authentication and behavioral analysis. A hybrid encryption mechanism combining CPJTABE and THKSCC is employed to safeguard data during transmission. Simultaneously, by using the QASOA, load balancing across devices is achieved. QASOA also aids in feature selection during intrusion detection. Data collected by IoT sensors is continuously monitored for anomalies, with outdated or suspicious devices identified through an HIT-Fuzzy logic system. These inputs feed into the LCA-BST-based network monitoring tool, which enables RT device tracking and hierarchical authentication. The system leverages the JPAO-GRU for intrusion detection, following a rigorous preprocessing stage that comprises normalization and dimensionality

reduction. Detected attacks are classified into tampering, alteration, or forgery utilizing specialized detection techniques like Dynamic Time Warping (DTW), Kalman Filtering (KF), and Logistic Normalization (LN). While enabling proactive and scalable network defense in dynamic IoT-cloud systems, this framework ensures data authentication security.

3.6.1.1 IoT network initialization and device registration

The proposed secure cloud-assisted IoT framework is initiated with the initialization of the IoT network and the secure registration of devices. To establish reliable connectivity among distributed IoT nodes, the system configures the network topology, communication protocols, and gateway specifications. After the environment is initialized, each IoT device experiences a registration process by deploying its unique credential set. This set typically includes a device ID, manufacturer signature, authentication token, and access permissions see equation 3.42.

$$\gamma_n \xrightarrow{\text{IoT device}} \langle \gamma_1, \gamma_2, \dots, \gamma_N \rangle \text{ Where, } n = 1 \text{ to } N \quad (3.42)$$

Here, N displays the number of γ_n . Now, for the registered devices (γ_n) , the encryption keys are produced using the proposed CPJTABE. Following registration, each verified device is assigned a unique encryption key pair generated utilizing the CPJTABE scheme. This encryption scheme, detailed in Section **Data security**, facilitates fine-grained AC by binding encryption policies to device attributes. Attribute-centric encryption is assisted by the CPJTABE model. It ensures that specific data can be decrypted only by the devices with matching credentials, thus enforcing a context-aware as well as policy-driven security model across the network. This initialization and registration process sets the stage for secure data sensing, transmission, and monitoring in the succeeding layers of the proposed framework.

3.6.1.2 Data sensing

The next essential step involves the sensing of data from the IoT environment. Here, the registered IoT devices, equipped with numerous embedded sensors, gather RT data based on their respective domains, namely temperature, pressure, humidity, motion, gas concentration, or system performance metrics. For enabling context-aware operations, automated responses, and higher-level analytics within the cloud infrastructure, this sensed

data is critical. From the γ_n , the IoT data is sensed and transferred to a cloud platform to enable DM.

$$\delta_s = (\delta_1, \delta_2, \dots, \delta_s) \quad (3.43)$$

Here in equation 3.43 above $s = 1 \text{ to } S$ illustrates the number of sensed IoT data δ_s . Then, the gathered data δ_s is transferred to the cloud platform where it can be processed, analyzed, and acted upon. Centralized storage, computation, and DM, which are vital for scalable deployment in smart environments, are assisted by this cloud-centered architecture. Before the data reaches the cloud, it is routed via the proposed security mechanisms, comprising encryption, signature verification, and anomaly detection, for ensuring its authenticity, integrity, and confidentiality.

3.6.1.3 Attribute extraction

The framework proceeds with the extraction of relevant attributes from both the devices and the data following the sensing of IoT data. Attribute extraction is a critical step in enabling policy-driven encryption, authentication, and DM within the cloud-assisted IoT environment.

A set of device-specific attributes is extracted from the registered IoT devices. These attributes encompass the device ID (a unique identifier for each node), device type (e.g., sensor, actuator, or gateway), firmware version, geographical location coordinates, and network address (e.g., IP or MAC address). For performing AC, device-specific encryption, anomaly identification, and system audits, these attributes are essential.

Simultaneously, a separate set of data-level attributes is identified from the sensed data. These include the data type (e.g., numeric, categorical, multimedia), the data source (i.e., originating device or node), and the data sensitivity level (e.g., public, private, confidential), which are deployed to enforce customized encryption policies and determine routing or storage strategies within the cloud infrastructure.

Then, the device attributes $(\mathfrak{S}_{att})$ like device ID, device type, firmware version, location data, and network address are extracted from γ_n . Likewise, the data attributes $(\mathfrak{S}_{att})$, such as data type, data source, and data sensitivity, are extracted from δ_s . See equation 3.44.

$$\alpha \rightarrow \|\mathfrak{S}_{att}, \mathfrak{S}_{att}\| \quad (3.44)$$

Where, α notates the extracted attributes.

3.6.1.4 Network monitoring tool

In order to facilitate intelligent and hierarchical IoT-NM, the proposed framework integrates a hybrid tree-based structure known as the LCA-BST. This structure facilitates efficient tracking, categorization, and authentication of devices centered on their operational attributes. The core objective is to maintain a synchronized, searchable, and scalable monitoring architecture that reflects RT network dynamics while supporting efficient query operations.

By using device-level attributes, namely device location, device ID, device type, and timestamp of registration, a Hierarchical Tree (HT) is first constructed. The structure follows a Binary Search Tree (BST) methodology, which inherently assists fast retrieval as well as insertion operations. Nevertheless, traditional BSTs are known to suffer from synchronization limitations in distributed environments, especially under conditions of frequent data updates and unbalanced branching. To address this, the Lowest Common Ancestor (LCA) mechanism is introduced to enhance the BST with context-aware lineage tracing and structural balance. In the LCA-BST, the device location is treated as the root node, device type as the intermediate child nodes, and device ID combined with timestamps as leaf nodes. This configuration ensures that each device is structured hierarchically centered on its physical as well as logical properties.

Thus, the LCA is introduced to ensure a faster lookup. The HT (τ_v) is structured by pondering attributes like device location (root), device type (child node), and device ID with timestamps (leaf node). Primarily, the root node is inserted at the top of the tree. If the value (v) to be inserted is less than the present node's value (Ω) , then the value is shifted to the left

child. If the value to be added is larger than the current node's value, then the value is shifted to the right child see equation 3.45

$$\tau_v \rightarrow \begin{cases} v \leq \Omega, & \text{left} \\ v \geq \Omega, & \text{right} \end{cases} \quad (3.45)$$

The proposed work introduces the LCA to understand the common ancestor, ensuring consistency. The LCA (ℓ^∞) is the deepest node $(\alpha\varphi)$ that is common to both nodes (v_1, v_2) see equation 3.46.

$$\ell^\infty(\tau_v) = \{\alpha\varphi \in v(\Omega) \mid \alpha\varphi \text{ is ancestor for both } (v_1, v_2)\} \quad (3.46)$$

Thus, the status of the IoT devices is constantly monitored by the generated HT.

3.6.1.5 Outdated device identification

It is essential to detect and isolate ODs that operate with obsolete configurations, outdated firmware, or legacy communication protocols to sustain the integrity and operational efficiency of the cloud-assisted IoT network. These devices not only degrade performance but also pose significant security risks by becoming potential entry points for attackers. The HTT-Fuzzy logic system is introduced in the proposed framework to intelligently identify such ODs based on RT device metadata. The traditional fuzzy logic system is known for its simplicity and human-like reasoning capabilities, making it suitable for classification tasks in uncertain environments. However, conventional fuzzy systems often face tuning limitations, especially when modeling smooth transitions between states or adapting to dynamically changing inputs. By employing a Hyperbolic Tangent Trapezoidal (HTT) Membership Function, the HTT-Fuzzy system enhances the standard fuzzy model, which offers greater smoothness in the transition between fuzzy states and better adaptability during fuzzification.

Also, the proposed HTT-Fuzzy is utilized by considering the device details $(\mathfrak{S}_{att})$ to identify the OD, thereby reducing the communication delay. The fuzzy algorithm is simple as well as understandable. Yet, the fuzzy algorithm had tuning issues. Therefore, the HTT

membership function (λ°) is employed to enhance the fuzzification performance equation 3.47.

$$\lambda^\circ(\mathfrak{I}_{att}) = \begin{cases} \frac{1}{2}(1 + \tanh(\zeta(\mathfrak{I}_{att} - x))), & x \leq \mathfrak{I}_{att} < y \\ 1, & y \leq \mathfrak{I}_{att} < z \rightarrow \Lambda \\ \frac{1}{2}(1 + \tanh(\beta(u - \mathfrak{I}_{att}))), & z \leq \mathfrak{I}_{att} \leq u \end{cases} \quad (3.47)$$

Here in equation 3.48. Λ designates the fuzzy data, (x, y, z, u) exhibits the constant values, and (ζ, β) displays the transition smoothness. Next, the fuzzy rules (ϕ_{rule}) are framed to identify the OD.

$$\phi_{rule}(\Lambda) = \begin{cases} \text{If } (D_{age} \ \& \ Fi = new, N_{type} = 5G, C_{quality} = \mathfrak{w}), & \text{not-outdated} \\ f(D_{age} \ \& \ Fi = old, N_{type} = 3G, C_{quality} = \hbar), & \text{outdated} \end{cases} \quad (3.48)$$

Where, D_{age} , Fi , $C_{quality}$, and N_{type} signify the device age, firmware age, communication protocol, and network type, respectively, $5G$ portrays the fifth generation, $3G$ designates the third generation, $\mathfrak{w}$ indicates the Zigbee 3.0 and HTTP/3, and $\hbar$ signifies the Zigbee 1.2 and HTTP 1.1 Contrarily, the crisp data is obtained via the defuzzification. Then, alerts about the OD are sent to the admin.

3.6.1.6 Data security

The proposed methodology employs a hybrid cryptographic mechanism combining CPJTABE with THKSCC to assure robust data confidentiality and fine-grained AC in cloud-assisted IoT environments. This dual-layered encryption model is designed to render a flexible, lightweight, and highly secure data protection framework tailored for distributed IoT nodes operating under constrained resource.

3.6.1.7 CPJTABE: Ciphertext-Policy Jacobi Theta Attribute-Based Encryption

The CP-ABE model is recognized for its flexibility in enforcing AC policies. It encrypts data in such a way that only users whose attributes satisfy a predefined access structure are able

to decrypt it. Nevertheless, conventional CP-ABE schemes often struggle with higher computational overhead, thereby making them inappropriate for RT IoT deployments.

To tackle this, the proposed work presents the Jacobi theta function in the CP-ABE formulation, causing CPJTABE. It also enhances security and the expressiveness of attribute encoding without considerably augmenting the computational burden.

Concurrently, the δ_s is encrypted by using the proposed CPJTABE-THKSCC. CPABE had higher flexibility. Still, the CPABE had computational complexity. Therefore, the proposed method employs the Jacobi theta function to increase the security level. Here, centered on the device attributes, the encryption keys are produced using the CPJTABE. The keys are preserved to ensure LE using the THKSCC. ECC had faster computation. Yet, it had a negative point in the EC. Thus, the proposed work introduces the Twisted Hessian-Kummer Surface Curve (THKSC).

Originally, the system parameters are generated by a TA. The bilinear map function (σ_f) is computed from the bilinear groups (Z_1, Z_2) of prime order (ρ) as equation 3.49.

$$\sigma_f : Z_1 \times Z_1 \rightarrow Z_2 \quad (3.49)$$

Subsequently, MSK (S^{key}) is randomly chosen, followed by PK (Θ) creation see equation 3.50 and equation 3.51.

$$S^{key} = \varsigma^\circ \in Fd_\rho \quad (3.50)$$

$$\Theta = \left(Z_1, Gr, Pb = Gr^{\varsigma^\circ}, Pv = Gr^{\frac{1}{\varsigma^\circ}} \right) \quad (3.51)$$

Here, ς° , Pb , and Pv displays the parameters, Fd states the finite field, and Gr illustrates the generator. The TA chooses a random SK for each device γ_n with an attribute set α . After that, the PRK component $(\rho\kappa_j)$ is generated by using the Jacobi theta function for each attribute equation 3.52.

$$\rho\kappa_j = \sum_{j=-\infty}^{\infty} \exp\left(Gr^{(\pi\varsigma^\circ + rd^2)} \cdot \lambda_{hash}(\alpha\alpha t_j)^{2\pi rd}\right) \quad (3.52)$$

The device's PRK $(U_{private})$ is obtained as below equation 3.53.

$$U_{private} = (\rho\kappa_j, \forall \alpha\alpha t_j \in \alpha) \quad (3.53)$$

Where, $j = 1 \text{ to } J$ specifies the number of attributes $\alpha\alpha t_j$, λ_{hash} portrays the hash function, and rd displays the random value. To encrypt the data, the CT components (Q, R, T) are computed equation 3.54, equation 3.55 and equation 3.56.

$$Q = \delta_s \cdot \sigma_f(Gr, Gr)^{\varsigma^\circ I} \quad (3.54)$$

$$R = Gr^I \quad (3.55)$$

$$T = \lambda_{hash}(\alpha\alpha t_j)^I, \forall \alpha\alpha t_j \in ac_{pcy} \quad (3.56)$$

Here in equation 3.57. I signifies the random encryption secret and ac_{pcy} portrays the access policy. The CT (En) is illustrated as,

$$En = (ac_{pcy}, Q, R, T) \quad (3.57)$$

During decryption, if the access policy is satisfied by their attributes, the user with an attribute set decrypts the data only see equation 3.58.

$$\sigma_f(R, \rho\kappa_j) = \sigma_f(Gr, Gr)^{\varsigma^\circ rd} \quad (3.58)$$

The original data (δ_s) is retrieved utilizing the below expression equation 3.59.

$$\delta_s = \frac{Q}{\sigma_f(R, \rho\kappa_j)} \quad (3.59)$$

3.6.1.8 Load balancing

Effective load balancing is essential to maintain system responsiveness, minimize transmission delays, and ensure the optimal utilization of computational resources in large-scale cloud-assisted IoT environments. Provided the non-uniform distribution of sensed data and fluctuating network traffic, load balancing assists in preventing processing congestion and improving overall Quality of Service (QoS).

To address this, the proposed work presents a novel QASOA—a bio-inspired optimization framework designed to achieve dynamic and intelligent load distribution across encrypted data streams. The model draws inspiration from the natural social behavior of quokkas, which illustrates efficient foraging as well as cooperative movement strategies.

i. Base Model: Quokka Swarm Optimization (QSOA)

The conventional QSOA simulates the group behavior of quokkas navigating environmental terrains. Each individual quokka signifies a candidate solution (e.g., a potential data assignment path or resource mapping). While QSOA is known for its simplicity and precision, it suffers from premature convergence in which the search agents settle too early around suboptimal local minima.

ii. Improvement via Airy Function (AF)

The proposed QASOA integrates the Airy Function (AF) into the solution update phase to enhance global convergence and exploration capabilities. The Airy function is a special function deployed in quantum mechanics and signal processing for representing smooth transitions and oscillatory behavior in potential fields. It enhances the balance between exploration (diversity) and exploitation (intensification) in the solution space.

Here, the encrypted data En is balanced by using the QASOA. The QSOA simulates the behavior of quokka animals. The QSOA renders a precise solution. Nevertheless, it had premature convergence issues. Therefore, the proposed work presents the Airy Function (AF). Here, the En is assumed as the quokka (member), which is initialized in the search area. Later, the minimum response time is deemed as the fitness value. To improve the convergence rate, the AF (L_{bl}) is used equation 3.60.

$$L_{bl}(En) = \frac{1}{\pi} \int_0^\infty \cos\left(\frac{dc^3}{3} + En(dc)\right) + \sin\left(\frac{dc^3}{3} + En(dc)\right) \partial\Diamond$$
(3.60)

Here, dc signifies the integration variable and $\partial\Diamond$ portrays the differential parameter. Then, the drought (O^{dgt}) is updated as below equation 3.65.

$$U_{drought} = \frac{tm + hm}{0.8 \times O^{dgt}} + Wi * L_{bl} \cdot Pi$$
(3.61)

Next, the updated quokka's position $(En^\otimes)$ is obtained equation 3.66.

$$En^\otimes = En + U_{drought} \cdot \eta^1$$
(3.62)

Where, η^1 specifies the nitrogen ratio, $U_{drought}$ shows the updated drought, tm denotes the temperature ratio, hm proves the humidity ratio, Wi signifies the differences in weight between the quokka and leader, and Pi specifies the differences in position between the leader and quokka. The pseudocode of the proposed QASOA (Figure 3.15) is given below.

Input: Encrypted data En Output: Load balanced data $(\chi\tau)$
Begin Initialize $En, dc, \partial\Diamond$ and $U_{drought}$ Perform population initialization Assume fitness value Apply AF, $L_{bl}(En) = \frac{1}{\pi} \int_0^\infty \cos\left(\frac{dc^3}{3} + En(dc)\right) + \sin\left(\frac{dc^3}{3} + En(dc)\right) \partial\Diamond$ Update drought solution, $U_{drought} = \frac{tm + hm}{0.8 \times O^{dgt}} + Wi * L_{bl} \cdot Pi$ Update quokka's location, $En^\otimes = En + U_{drought} \cdot \eta^1$ Repeat until converge Return $(\chi\tau)$ End

Figure 3.15: The Pseudocode of the Proposed QASOA.

Then, the load-balanced data (χ^τ) is fed to the proposed IDS.

3.6.1.9 Intrusion detection system

The system proceeds to detect potential anomalies or malicious activities using an advanced IDS after the encrypted data is securely balanced across the cloud nodes. The IDS is crucial in recognizing unauthorized access, data tampering, packet injection, and behavior anomalies that could compromise data confidentiality, integrity, and system availability.

The proposed framework employs a JPAO-GRU model to achieve high detection accuracy while maintaining computational efficiency. This hybrid deep learning architecture combines the temporal pattern recognition capability of GRU networks with a probabilistic attention mechanism, augmenting the system's ability to detect subtle and long-term deviations in IoT data behavior.

a. Dataset

To train the proposed IDS, the CSE-CIC-IDS2018 dataset (D_v) is collected. Here, χ^τ is deemed as the input during training.

b. Pre-processing

Then, the D_v is pre-processed to enhance the data quality.

The raw IoT data must be cleaned, transformed, and standardized via a comprehensive data preprocessing pipeline before feeding the gathered data into the JPAO-GRU-based IDS. This step assures the input data's quality, consistency, and suitability for accurate anomaly detection and classification.

The first phase includes Data Duplication Removal (DDR) in which the system identifies and eliminates redundant or repeated data entries that can skew model learning and upsurge computational overhead. This ensures that each data point contributes uniquely to the pattern recognition process.

Following DDR, the dataset may still contain missing values, often resulting from transmission loss, sensor errors, or packet corruption. The Missing Value Imputation (MVI) technique is applied to address this. Rather than removing incomplete records—which could

lead to information loss—the system imputes missing entries using the available (non-missing) values' mean for the corresponding feature. While ensuring continuity in the data sequence, this approach maintains dataset integrity.

Next, by using appropriate encoding schemes, categorical and symbolic data types are converted into numerical feature vectors. This is essential for compatibility with the GRU-based neural network model, which necessitates numerical input.

After encoding, the data is subjected to normalization via Min-Max scaling or Z-score standardization to ensure that all features contribute proportionately and are mapped to a common numerical range. This prevents bias in learning owing to varying feature scales.

The method of eliminating redundant data is called DDR. For imputing the missing value in the de-duplicated data in MVI, the mean values of the non-missing data are utilized. Next, feature vectors are converted into numerical values, followed by normalization. Therefore, the pre-processed data is shown as (Ξ_{pre}) .

c. Feature extraction

The system includes a dedicated feature extraction phase to optimize the proposed IDS's learning accuracy and classification efficiency. This step concentrates on isolating and selecting the most informative characteristics from the pre-processed IoT traffic data, thus mitigating noise and dimensionality while maximizing the relevance of input signals to the detection model.

The extracted features $(K\Phi)$ include:

- i. Destination Port: Recognizes the service or application the packet is targeting. Unusual port activity can signal unauthorized access attempts or port scanning.
- ii. Protocol: Specifies the communication protocol utilized (e.g., TCP, UDP, ICMP), which helps in identifying protocol-specific attack vectors.
- iii. Timestamp: Records the time of data packet generation, useful for sequence alignment and temporal anomaly detection.

- iv. Flow Duration: Measures the time taken for a complete data session or flow, with unusually long or short durations often linked to botnet or DoS activity.
- v. Total Flow Bytes: Represents the amount of data exchanged in a given session, beneficial to detect volumetric attacks.
- vi. Average Packet Size: Assists in capturing anomalies in payload distribution, often indicative of data injection or stealth attacks.

From Ξ_{pre} , features like destination port, protocol, timestamp, flow duration, flow bytes, and average packet size are extracted to improve the classifier's performance.

d. Feature selection

Feature selection is the next critical step after the extraction of relevant features from the pre-processed dataset. Feature selection involves recognizing the best subset of features significantly contributing to classification performance while eliminating redundant or non-informative attributes. Effective feature selection diminishes the computational load, minimizes overfitting, and augments the learning model's generalization capability. Later, the optimal features (Ψ_{opt}) are selected from $K\Phi$ using the proposed QASOA, which is explained in Section 3.6.1.7. QASOA is a nature-inspired, populace-centered metaheuristic that models the quokkas' collective foraging behavior, augmented with the Airy Function to prevent premature convergence and maintain search diversity. Each feature subset in the population is signified as a candidate solution (quokka), and the classification accuracy achieved using that feature subset is used as the fitness value for selection. Maximizing the classifier's performance while minimizing the feature count is the objective.

3.6.1.10 Intrusion detection using JPAO-GRU

Next, Ψ_{opt} is fed into the proposed JPAO-GRU, detecting the intrusion activities in an IoT network. The GRU had higher computational efficiency. Nevertheless, it had over-fitting issues and low learning efficiency. Therefore, the Jeffreys prior regularization and Aranda-ordaz activation are employed in the proposed method. The JPAO-GRU's architecture is given in (Figure 3.16).

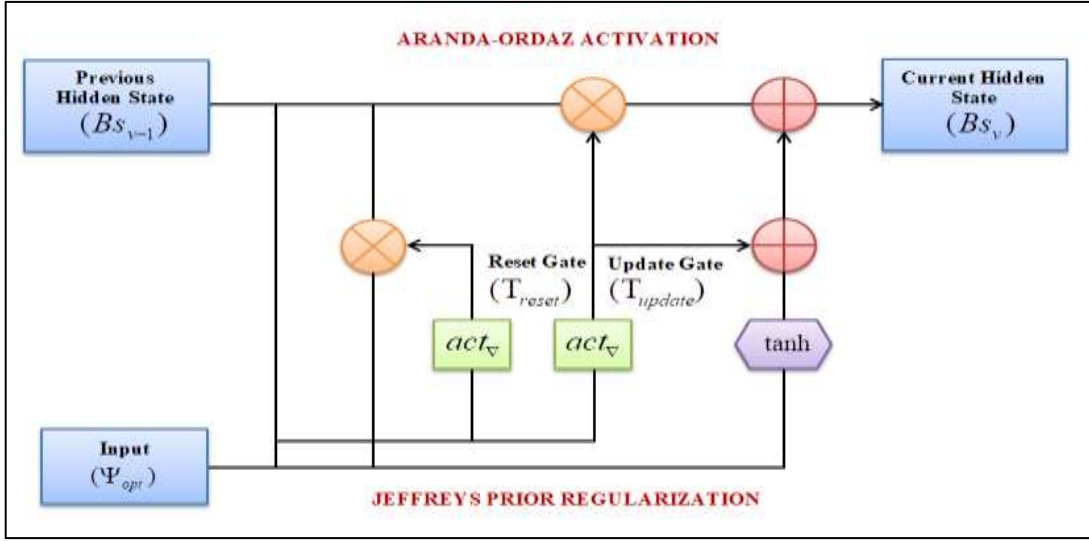


Figure 3.16: The Structural Representation of the Proposed JPAO-GRU.

Conventional GRUs are favored for their lower computational overhead and potential to model temporal dependencies using gating mechanisms. Nevertheless, GRUs often suffer from overfitting on small or noisy datasets and suboptimal learning when faced with highly nonlinear patterns. The proposed JPAO-GRU introduces two key improvements to address these issues:

The Aranda-ordaz activation (act_{∇}) assists in upgrading the training process in equation 3.67.

$$act_{\nabla}(\psi_{opt}) = 1 - \left(1 + \lambda \diamond \exp(\psi_{opt})\right)^{-\frac{1}{\lambda \diamond}} \quad (3.63)$$

Here, $\lambda \diamond$ specifies the fixed parameter. Similarly, the Jeffreys prior regularization is used to initialize the weight (Γ_{wt}), diminishing the overfitting issues equation 3.68.

$$\Gamma_{wt} = Lk_{fun}(\theta | \psi_{opt}) \propto Lk_{fun}(\psi_{opt} | \theta) \pi(\theta) \quad (3.64)$$

Where, Lk_{fun} signifies the likelihood function. The reset gate (T_{reset}) determines the less significant information that needs to be removed from the gates see equation 3.69.

$$T_{reset} = act_{\nabla} * (\Gamma_{wt} \cdot (Bs_{v-1}, \psi_{opt})) + ov \quad (3.65)$$

The update gate (T_{update}) is used to recognize the information that needs to be added to the succeeding gates equation 3.70.

$$T_{update} = act_{\gamma} * (\Gamma_{wt} \cdot (Bs_{v-1}, \psi_{opt})) + o\upsilon \quad (3.66)$$

Here in equation 3.71 $o\upsilon$ portrays the bias and Bs_{v-1} states the prior hidden state. Later, the hidden state $(\hat{Bs}_v)$ of the candidate is updated centered on the reset gate.

$$\hat{Bs}_v = \tanh(\Gamma_{wt} \cdot (T_{reset} \times Bs_{v-1}, \psi_{opt})) \quad (3.67)$$

Based on the update gate, the hidden state is updated in equation 3.72.

$$Bs_v = (1 - T_{update}) \cdot Bs_{v-1} + T_{update} \cdot \hat{Bs}_v \quad (3.68)$$

Finally, the proposed JPAO-GRU classifies the data into attacked (μ^{att}) or non-attacked (Non^{att}) . The pseudocode of the proposed JPAO-GRU is shown in (Figure 3.17).

```

Input: Optimal features  $\psi_{opt}$ 
Output: Intrusion detection
Begin
  Initialize  $\psi_{opt}$ ,  $act_{\gamma}$ ,  $\Gamma_{wt}$  and  $Bs_v$ 
  For 1 to each  $\psi_{opt}$  do,
    Activate  $act_{\gamma}(\psi_{opt}) = 1 - (1 + \lambda \odot \exp(\psi_{opt}))^{-\frac{1}{\lambda \odot}}$ 
    Initialize weight,
       $\Gamma_{wt} = Lk_{jam}(\theta | \psi_{opt}) \propto Lk_{jam}(\psi_{opt} | \theta) \pi(\theta)$ 
    Evaluate reset gate
       $T_{reset} = act_{\gamma} * (\Gamma_{wt} \cdot (Bs_{v-1}, \psi_{opt})) + o\upsilon$ 
    Determine update gate  $(T_{update})$ 
    Update candidate hidden state & hidden state
       $\hat{Bs}_v = \tanh(\Gamma_{wt} \cdot (T_{reset} \times Bs_{v-1}, \psi_{opt}))$ 
       $Bs_v = (1 - T_{update}) \cdot Bs_{v-1} + T_{update} \cdot \hat{Bs}_v$ 
    Display detection outcomes
  End For
Return  $(\mu^{att}, Non^{att})$ 
End

```

Figure 3.17: The Pseudocode of the Proposed JPAO-GRU.

The JPAO-GRU thus renders a highly adaptable, regularized, and expressive deep learning framework to detect intrusions in cloud-assisted IoT networks. It offers improved accuracy, learning efficiency, and resilience to overfitting by combining advanced activation and probabilistic regularization techniques. Then, data authentication is done to reveal how the data is manipulated.

3.6.1.11 Data authentication

Following intrusion detection, the system performs a multi-dimensional data authentication process for evaluating the nature as well as the integrity of the detected attacked data. This phase is designed to identify whether the intrusion corresponds to tampering, alteration, or forgery, thereby permitting the system to classify the intrusion type and trace its origin more precisely.

Each of these categories reflects a different mode of compromise:

- i. Tampering refers to the intentional modification of legitimate data values.
- ii. Alteration involves unintentional or accidental corruption, often owing to noise, faulty transmission, or sensor error.
- iii. Forgery refers to the injection of entirely fake or impersonated data into the system, often mimicking valid sources.

Regarding tampering (intentional), alteration (accidental), and forgery (intentional and fake), the data authentication is assessed for attacked data μ^{att} . Dynamic Time Warping (DTW) is used to gauge the similarity between 2 time-dependent sequences. By analyzing the deviations in the warping paths, the tampered data is recognized. The Kalman Filter (KF) is employed for alternation detection to predict sensor readings and detect deviations from expected values. To measure expected variations, Laplacian Noise (LN) is added to sensor readings. By analyzing statistical inconsistencies, the forged IoT data is detected. Lastly, the current status of the attacked and non-attacked nodes is updated in the HT to ensure IoT-NM. Therefore, data authenticity is enhanced by the research methodology.

3.6.2 Summary

In this section, a comprehensive methodology for secure and intelligent cloud-assisted IoT-NM through an integrated framework combining hybrid encryption, intelligent intrusion detection, and multi-layered data authentication is presented. The process starts with IoT network initialization and device registration, followed by attribute extraction and secure key generation utilizing the CPJTABE-THKSCC encryption model. Load balancing is achieved by the QASOA, thereby ensuring efficient task distribution. By using the JPAO-GRU model, the Intrusion detection is performed. It is then enhanced with Aranda-Ordaz activation and Jeffreys's prior regularization to improve learning efficiency and generalization. Preprocessing steps, namely DDR, imputation, normalization, feature extraction, and optimal feature selection prepare the data for high-accuracy classification. By using DTW, Kalman Filter, and Laplacian Noise mechanisms, the detected attacked data is further analyzed to authenticate the data as tampered, altered, or forged. Lastly, the node status is updated within the LCA-BST-based monitoring tool, reinforcing trust and traceability. Overall, the methodology ensures end-to-end security, efficient data handling, and resilient authentication for dynamic IoT-cloud infrastructures.

4. RESULTS

4.1 RESULTS

4.1.1 Cipher Text-Policy Attribute-Based Encryption

CP-ABE is a highly functional and efficient version of ABE. Ensuring security and granular access control of outsourced data is a crucial aspect of this scheme. However, in a previous iteration of CP-ABE, the access policy is supplied in plaintext and ciphertext [141]-[142]. This could compromise the user's privacy by disclosing personal information. The authenticity of the data owner and the integrity of the outsourced data cannot be confirmed in the current CP-ABE scheme. All insiders are regarded as authorized users in an ABE scheme to access the original message. Consequently, it might be able to re-encode the generated plaintext with a different or comparable access policy. Insider theft is identified using the short signature method to prevent this kind of situation. Verifying the legitimacy of shared data and verifying data owner authentication are the two key responsibilities of the short signature approach. We demonstrate the security of this method against an adaptive message attack.

- i. $\text{Setup}()(\lambda) \rightarrow (\text{PK}, \text{MSK})$

The key generation center is in charge of carrying out this algorithm. Two finite prime order R random cyclic symmetric groups, G_1 and G_2 , with a y generator, are chosen by the KGC. The bilinear map is defined as $e: G_1 * G_1 \rightarrow G_2$, taking λ into account as a public security parameter. KGC selects two exponents, ϕ and $\gamma \in \mathbb{Z}_R$, at random. Based on the security parameter (λ), a master secret key (MSK) and a public key (PK) are generated.

- ii. $\text{KeyGen}(\text{PK}, \text{MSK}, A) \rightarrow \text{SK}$

To produce a secret key for the authorized user (U_t), this procedure runs the public key (PK), master secret key (MSK), and set of attributes (SA) as input. The secret key is generated based on two different random numbers, $M_t, M_j \in \mathbb{Z}_R$.

- iii. $\text{SignKey}(\text{sk}) \rightarrow \text{pk}$

A random number, x , is selected by the data owner from the range 0 to $R - 1$. The result of this function is the public verification key ($k = yx$) and private key (x), where y is the $G1$ generator.

iv. Encryption

$\text{EncipherSign}(PK, P, AP) \rightarrow \sigma, CT$

We recommend that access control in the system be achieved by inserting the access policy into the ciphertext. Interior nodes define the threshold gates, while leaf nodes define user credentials and attributes. The message P is encrypted using the public key before the process commences.

v. Decryption and Verification

$\text{DecipherVerify}(PK, SK_{ut}, CT, \sigma, pk) \rightarrow P,$

When a cloud user's attribute matches an embedded access policy's attribute within the ciphertext, the decryption process is successful. If it's not, the cloud user won't be able to decode the encrypted data. An insider attack can be prevented by using the BLS short signature to verify the authenticity of the data owner.

4.1.2 Hybrid Cipher Text Policy Attribute-Based Encryption (H-CP-ABE)

Two techniques offer more precise access control over encrypted data on the cloud: key-policy attribute-based encryption (KP-ABE) [143] and cipher-policy attribute-based encryption (CP-ABE) [144]. The encryptor can't determine who can decrypt the encrypted data, and KP-ABE struggles with handling multiple levels of attribute authorities. It also lacks flexibility in managing attributes and scalability. Decryption is available to all customers whose private key complies with the access policy. Furthermore, the main issuers are the only option available to customers. Data owners are the only trusted authority (TAs) in existence. This scheme cannot be used with multiple data owners because each user would receive numerous keys from different owners. Key management in this scheme is challenging. KP-ABE is not appropriate for some applications, such as message broadcasting encryption. When an access structure is linked to the ciphertext and the user's private key is defined by a set of attributes, CP-ABE is found to be completely appropriate. Only when the

private key properties match the ciphertext's access structure during encryption can a user decrypt the message. A combination of CP-ABE and KP-ABE is called HbABE. We employed Zhiguo Wan et al.'s "A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing (HASBE)" [145] to ensure appropriate key management. The concept of hierarchical attribute-based encryption (HABE) was initially introduced by [146] as a way to combine hierarchical identity-based encryption (HIBE) and CPABE to provide fine-grained access control in cloud storage services. Zhiguo Wan provided a system model for managing and distributing the essentials. An excellent cryptographic scheme is hybrid attribute-based encryption. For the exclusive user group, like company executives. This plan aims to safeguard the crucial documents and information kept inside the organization, with access restricted to those with permission. Key generation and management are done in a secure hierarchical key generation center. An access structure, consisting of a set of attributes represented as a tree structure, is assigned to each authorized user.

The company's head office, a reliable source, will be in charge of developing the tree's universal access structure. We employ a similar system model [147], which outlines a useful hierarchical model for creating and administrating the key.

i. Libraries and Imports

The code uses the Charm library, which provides tools for cryptographic operations.

ii. HybridABEnc Class

This class represents a hybrid ABE scheme, inheriting from the `ABEnc` class.

It initializes with an encryption scheme (`scheme`) and a cryptographic group (`groupObj`).

iii. Setup () Method

The public key PK and master secret key MK are produced by this algorithm, which accepts a security parameter as input. Senders of messages encrypt their messages using PK. MK is only known to the domain authority and is used to generate user secret keys. Assigns responsibilities to the `setup` method of the `abenc` ABE scheme. The public parameters are configured using this method.

iv. Keygen () Method (MK, K2) $\rightarrow$ (SK)

The master secret key MK and the access tree K2 linked to the data owner are the inputs used by this algorithm. If and only if it matches T, it outputs a secret key (SK) that allows the user to decrypt a message encrypted under an access tree structure. Assigns responsibilities to the `keygen` method of the `abenc` ABE scheme. By using an object attribute, a master secret key ({mk}), and a master public key ({pk}), this method creates a secret key.

v. Encrypt () Method (PK, CT1, T) $\rightarrow$ CT2

This algorithm accepts the following inputs: a message (CT1), a public parameter (PK), an access structure (T), and the encrypted message (CT). It then generates the ciphertext (CT2) so that the message can only be decrypted by the user who possesses an access tree and satisfies the access structure associated with the ciphertext during the encryption process.

creates a random key ({key}) and uses the underlying ABE scheme ({abenc}) to encrypt the object attribute ({object}). creates a symmetric encryption scheme based on the key's SHA-1 hash. uses the symmetric encryption scheme to encrypt the message ({M}). Provides the random key along with the ciphertext ({c2}).

vi. Decrypt() Method (CT2, PK, SK) $\rightarrow$ CT1

The cipher text CT1, which contains the access policy T, and the public parameter PK are input to the decryption algorithm along with the private key SK for access control K2. If K3 is a subset of T, it returns CT1. Second-level encryption and decryption.

Uses the symmetric encryption scheme that is instantiated from the key to decrypt the ciphertext ({ct}).

vii. Instantiation of Crypto Objects

generates the CP-ABE scheme ({cpabe}) and a cryptographic group ({group}).

Initializes the CP-ABE scheme with the hybrid ABE scheme ({hyb_abe}).

Access Policy and Messages

For access conditions, a tree access structure with internal nodes acting as logical operators and external nodes having distinct attributes is specified. Only users whose attributes specified at the time of key generation satisfy the access structure can access a ciphertext file. The values specified in the attribute list are combined with logic operators to create logical formulae that serve as access conditions. Access conditions can be defined using logic operators like AND, OR, and t of n threshold gate, which requires t out of n attributes to match the access conditions. Hence, an AND operator represents n of n threshold gates, while an OR operator represents 1 of n threshold gates. A user needs a key to match attributes to decrypt cipher text. At the moment of encryption, the terms for file access are established and integrated into the file. defines a Boolean expression ($\{access_policy\}$) that represents an access policy.

The data owner uses the access policy T to encrypt a message M . If the users' attribute set satisfies the T , users decrypt the ciphertext CT . Three different message types are used for encryption: ``msg_str{`, ``msg_int{`, and ``msg_float{`.

viii. Key Generation and Encryption

`Setup()` is called to generate a master public key (``master_public_key``) and master secret key (``master_key``). Key generation and encryption are performed for string, integer, and float inputs.

ix. Decryption

Decryption is performed for each ciphertext, producing the decrypted values (``dec_str``, ``dec_int``, ``dec_float``).

4.1.3 Hybrid Cipher Text Policy Attribute-Based Encryption (ABE)

The definition and security model of our hybrid -CPABE are provided in this section. Such a system applies symmetric encryption, an encrypt-then-mac mechanism, and a circuit ciphertext-policy attribute-based encryption scheme to guarantee verifiable delegation, fine-grained access control, and confidentiality.

i. Attribute-Based Encryption (ABE)

A sort of public key cryptography known as attribute-based encryption allows the message to use User2's public key. After that, User2 decrypts it with her private key user to be uniquely identified by a collection of attributes, such as their name, department, title, or ID. Two keys are used in a traditional public key cryptography system: a private or secret key that is only known to the message's recipient and a public key that is known to all. To send User2 a secure message, User1 encrypts.

A cryptosystem in which predicates over attributes are used to express access policies. Ciphertexts are related to access policies, and secret keys are related to sets of attributes.

ii. Hybrid CP-ABE

All authorized users of the company are aware of the private key used in the first level of data encryption, which encrypts files using symmetric data encryption.

The concepts of KP-ABE and CP-ABE will be covered in the second level. The data will be encrypted using two access trees, and the ciphertext will not be linked to any additional attributes. Each authorized user in the organization has been given an access structure that serves as both their identity and their access. When a user wishes to encrypt data, they associate an access structure they received from their domain authority with the data and encrypt it using their access structure, which is the data owner's private key. Integrates symmetric key encryption with ABE. Encrypts attributes using ABE and the message itself using symmetric encryption.

iii. CP-ABE (Cipher Text-Policy ABE)

The Ciphertext-Policy Attribute-Based Encryption (CP-ABE) was first presented by [148]. It is also an altered ABE version. The main difference between CP-ABE and Key-Policy Attribute-Based Encryption is that the former operates in reverse order. A set of descriptive attributes is linked to the ciphertext during the encryption process in a key policy attribute-based encryption, where the access structure is the user's private key. The user compares his private key (access structure) with the ciphertext to decrypt it. Characteristics linked to the encrypted text. The ciphertext is decrypted if it matches; if not, it is not. Policy-Cipher: Attribute-Based Encryption (CP-ABE) operates contrary to KP-ABE. The user's private key

is used to encrypt the message using the set of descriptive attributes, and only if the access structure linked to the ciphertext and his private key match can the user decrypt the message. This is in contrast to the set of attributes being associated with the ciphertext. For instance, if $(1 \text{ AND } (2 \text{ OR } 3))$ is the encrypted data's access structure. One can decrypt data if a user's private key contains a set of attributes that are $(1 \text{ AND } 2)$.

iv. Symmetric Encryption

Data encryption and decryption using symmetric encryption share a common key. This encryption technique is safe and comparatively quick. The primary drawback of symmetric key encryption is that it requires the two parties involved in communication to share the key. If an attacker gets hold of the key, he can decrypt the data.

The random key's (key) SHA-1 hash is used to create a symmetric encryption scheme. After converting the message M to bytes, the ciphertext c_2 is obtained by applying the symmetric encryption scheme.

Encrypts and decrypts data using a common key. In this case, the message itself is encrypted using a randomly generated key.

4.1.4 Integrated lightweight and hidden Hybrid Cipher Text Policy Attribute-Based Encryption (IL-HH-CP-ABE)

IL-HH-CP-ABE leverages pairing-based cryptography, specifically the Bilinear Pairing on elliptic curve groups. This provides a mathematical foundation for efficient encryption and decryption operations. The algorithm builds upon the CipherText Policy Attribute-Based Encryption (CP-ABE) foundation. CP-ABE allows data to be encrypted with a policy expressed in terms of attributes, and decryption requires a secret key associated with a set of attributes that satisfy the policy [149]. IL-HH-CP-ABE introduces a hidden policy component, allowing for additional privacy and flexibility in access control. The hidden policy aspect enables more complex and expressive access policies without revealing the specifics of the policy to unauthorized entities. The algorithm incorporates lightweight design principles, aiming for computational efficiency and reduced resource consumption. This is particularly important for devices with limited processing power and storage capacity.

IL-HH-CP-ABE adopts a hybrid encryption approach, combining both symmetric and asymmetric encryption techniques. This hybrid design enhances the efficiency of the encryption and decryption processes.

Now, we present some data regarding the performance of the cpabe toolkit. Measurements of the times required for private key generation, encryption, and decryption are obtained by using the programs cpabe-keygen, cpabe-enc, and cpabe-dec on various problem sizes.

As anticipated, the time complexity of cpabe-keygen is linear in the number of attributes linked to the key it is issuing. Regarding the number of leaf nodes in the access policy, the cpabe-enc running time is nearly linear. A few multiplications make up the polynomial operations at internal nodes, which have little effect on the running time. For even the most severe problem cases, both are still very possible. Depending on how many attributes a key has or how many leaves there are in a policy tree, cpabe-keygen and cpabe-enc execute in a predictable length of time. The ciphertext's unique access tree and the properties in the private key determine the cpabe-dec's performance, which a few optimizations can enhance.

- a. ABEnc: This is a module for Attribute-Based Encryption (ABE) in the Charm cryptography library.
 - i. CPabe_BSW07: This is a specific CP-ABE scheme based on the Boneh-Sahai-Waters (BSW) construction.
 - ii. PairingGroup, GT: These are used for pairing-based cryptography, where the GT type represents elements in the target group of a bilinear pairing.
 - iii. AuthenticatedCryptoAbstraction: This instantiates a symmetric encryption scheme with authentication.
 - iv. HashPair: Alias for a pairing-friendly hash function.
 - v. Psutil: Used to get system-related information (e.g., memory usage).
 - vi. Json, time: Standard Python libraries for JSON processing and time-related functions.
- b. Class Definition (`ILHHCPABE``)

```

class ILHHCPABE(ABEnc):
    def __init__(self, scheme, groupObj):
        ABEnc.__init__(self)
        self.abenc = scheme
        self.group = groupObj

    def setup(self):
        return self.abenc.setup()

    def keygen(self, pk, mk, attributes):
        return self.abenc.keygen(pk, mk, attributes)

    def encrypt(self, pk, M, attributes):
        key = self.group.random(GT)
        c1 = self.abenc.encrypt(pk, key, attributes)

        # instantiate a symmetric enc scheme from this key
        cipher = AuthenticatedCryptoAbstraction(sha1(key))

        if isinstance(M, str):
            M = M.encode() # Convert string to bytes
        elif isinstance(M, (int, float)):
            M = str(M).encode() # Convert numeric types to bytes

        c2 = cipher.encrypt(M)
        return c2, key

```

c. `init`: Initializes the class with the provided scheme and group object.

i. Setup Method

This algorithm returns a public key (PK) and a master secret key (MK) after receiving the security parameter (κ) as input.

The `self.abenc` CP-ABE scheme's setup method is called the `setup` method.

ii. Keygen Method

This algorithm returns a secret key SK_L as associated with L after receiving as inputs PK, MK, and an array of attributes L. The public key ({pk}), master key ({mk}), and list of attributes are used by the `keygen` method to create a secret key.

Pk: Public key of the IL-HH-CP-ABE scheme.

MK: Master key of the IL-HH-CP-ABE scheme.

Attributes: A list of attributes associated with the secret key.

The key generation process involves invoking the underlying CP-ABE scheme's keygen method with the public key (pk), master key (mk), and the specified attributes. This results in the generation of a secret key. An authority or trusted entity typically performs key generation. The generated secret key allows a user with the specified attributes to decrypt messages encrypted with corresponding access policies.

iii. Encrypt Method

This algorithm accepts a message M and an access structure W as inputs. It produces a ciphertext C with the characteristic that a user possessing $Self$ can decrypt C only if $L|=W$.

The message ($\{M\}$) is encrypted using the `{encrypt}` method, which also creates a random key ($\{key\}$).

The CP-ABE scheme (`{self.abenc}`) is used to encrypt the attributes (`{attributes}`). The key and an authenticated encryption scheme are used to perform the symmetric encryption.

iv. Decrypt Method

This algorithm's inputs are PK , C (which W encrypted), and SKL . If SKL is associated with $L|=W$, it returns M .

A public key (`{pk}`), secret key (`{sk}`), ciphertext (`{ct}`), and a key are required for the ``decrypt`` method. The original message is obtained by decrypting the ciphertext using an authenticated encryption scheme.

d. Setting up the Scheme and Encryption

```
group = PairingGroup("SS512")
cpabe = CPabe_BSW07(group)
ilhhcpabe = ILHHCPABE(cpabe, group)

access_policy = '((four or three) or (two or one))'
msg_str = "hello world this is an important message."
msg_int = 123
msg_float = 123.45
(master_public_key, master_key) = ilhhcpabe.setup()
```

The CP-ABE scheme (`{cpabe}`) is instantiated, a pairing group is created, and an instance of `{ILHHCPABE}` (`{ilhhcpabe}`) is created using the pairing group and the CP-ABE scheme.

establishes the IL-HH-CP-ABE scheme in order to acquire the master key (`{master_key}`) and master public key (``master_key``).

e. Key Generation

The key generation process produces the secret key needed to decrypt messages. The secret key is linked to a collection of characteristics and is generated from the master key and public key of the IL-HH-CP-ABE scheme.

Generates a secret key associated with specific attributes, enabling the decryption of messages encrypted with corresponding access policies.

```
start=time.time()
secret_key_str = ilhhcpabe.keygen(master_public_key, master_key, ['ONE'])
end=time.time()
print('key generation time:',end-start)
```

Measures the IL-HH-CP-ABE scheme time taken for key generation.

f. Encryption

The encryption process is responsible for encrypting a message using the IL-HH-CP-ABE scheme. The encryption involves both attribute-based encryption and symmetric encryption.

Pk: Public key of the IL-HH-CP-ABE scheme.

M: Message to be encrypted.

Attributes: A set of attributes associated with the ciphertext.

A random key (key) is generated from the group GT. This key is used in the CP-ABE encryption (c1), which incorporates the specified attributes.

A symmetric encryption scheme is instantiated using the SHA-1 hash of the random key (key). The message M is converted to bytes, and the symmetric encryption scheme is applied to obtain the ciphertext c2. The output consists of the ciphertext c2 and the generated key. The key is necessary for decryption and should be securely shared with authorized parties.

Encryption is performed by a data owner who wishes to share a message securely with users possessing specific attributes. The ciphertext c2 is sent to the recipients. Combines attribute-based encryption and symmetric encryption to secure a message. The ciphertext is produced along with a key for later decryption.

```

start=time.time()
mem1 = (psutil.virtual_memory().available)
cipher_text_str, key_str = ilhhcpabe.encrypt(master_public_key, msg_str, access_policy)
mem2 = (psutil.virtual_memory().available)
end=time.time()
Encryption_time=end-start
hpabe_mem_enc = (mem1 - mem2)/1024
print ("Memory Usage:", hpabe_mem_enc)
print('Encryption_time:',end-start)

```

Measures the time taken for encryption and memory usage using the IL-HH-CP-ABE scheme.

g. Decryption:

The decryption process is responsible for recovering the original message from a ciphertext. It involves using the secret key associated with the attributes specified during key generation.

Pk: Public key of the IL-HH-CP-ABE scheme.

SK: Secret key for decryption.

Ct: Ciphertext to be decrypted.

Key: The key used during encryption.

An authenticated symmetric encryption scheme is instantiated using the SHA-1 hash of the key (key). This step is necessary to ensure the integrity of the symmetrically encrypted message. The ciphertext ct is decrypted using the symmetric encryption scheme. The decrypted message is obtained. The output is the decrypted message, which was originally encrypted with the corresponding attributes. Decryption is performed by a user possessing the secret key associated with specific attributes. The user can recover the original message from the received ciphertext. Utilizes the secret key to decrypt a ciphertext. Involves symmetric decryption and recovering the original message. These processes collectively implement the IL-HH-CP-ABE algorithm, providing a flexible and secure mechanism for attribute-based encryption with additional features for lightweight and hidden aspects. The scheme is designed to enhance privacy and access control in scenarios where fine-grained attribute-based access policies are crucial.

```

mem1 = (psutil.virtual_memory().available)
dec_str = ilhhcpabe.decrypt(master_public_key, secret_key_str, cipher_text_str, key_str)
mem2 = (psutil.virtual_memory().available)
Encryption_time=end-start
end=time.time()

hpabe_mem_dec = (mem1 - mem2)/1024
print ("Memory Usage:", hpabe_mem_dec)
Decryption_time=end-start
print('Decrypted String:', dec_str.decode())
print('Decryption_time:',Decryption_time)

Memory Usage: -176.0
Decrypted String: hello world this is an important message.
Decryption_time: 0.7180812358856201

```

Measures the time taken for decryption and memory usage using the IL-HH-CP-ABE scheme.

h.The Result after applying:

i. Encryption

Encryption efficiency depends on the complexity of the access policy. CP-ABE generally requires more computations during encryption due to the need to satisfy complex access structures. H-CPABE introduces a hidden policy, potentially reducing the computational cost during encryption. This can lead to improved efficiency compared to CP-ABE. HH-CPABE introduces hybrid encryption, combining symmetric and asymmetric encryption. This can enhance efficiency compared to CP-ABE and H-CPABE, especially for larger messages.

The times refer to the time needed to complete the encryption procedures. Additionally, ten users with various message sizes have their encryption times measured. Table 4.1 demonstrates that, compared to the suggested method, the suggested CP-ABE, H-CPABE, and HH-CPABE require less encryption time. Because the recommended method uses hybrid encryption techniques, which produce the key quickly, it operates similarly to traditional encryption. The only difference is that instead of encrypting a single character, it encrypts a digraph or a set of initial options. First, a 5*5 vector reference table is created. The informational encrypted message is represented by the letters in the grid.

Additionally (Figure 4.1) shows six random users' encryption time estimates in an IoT environment. The suggested method led to less encryption in the multiuser with multi-authority scenario. Access policy problems with the traditional CP-ABE methods lead to

abnormal increases in key generation times for each user and longer cryptographic times. Reduced encryption times are achieved by the suggested method's parallel generation of the public and private keys and its automatic alteration of key sizes by attribute authorities in response to message size.

Table 4.1: Encryption Time Comparison.

No. Of Attributes	CP-ABE	H-CPABE	HH-CPABE	Proposed Method
5	3.0	2	1.0	0.60
10	4.5	3	1.5	1.25
15	5.2	4	1.8	1.52
20	8.0	6	2.6	2.24
30	12.0	8	3.5	3.27
40	13.7	10	4.2	4.25

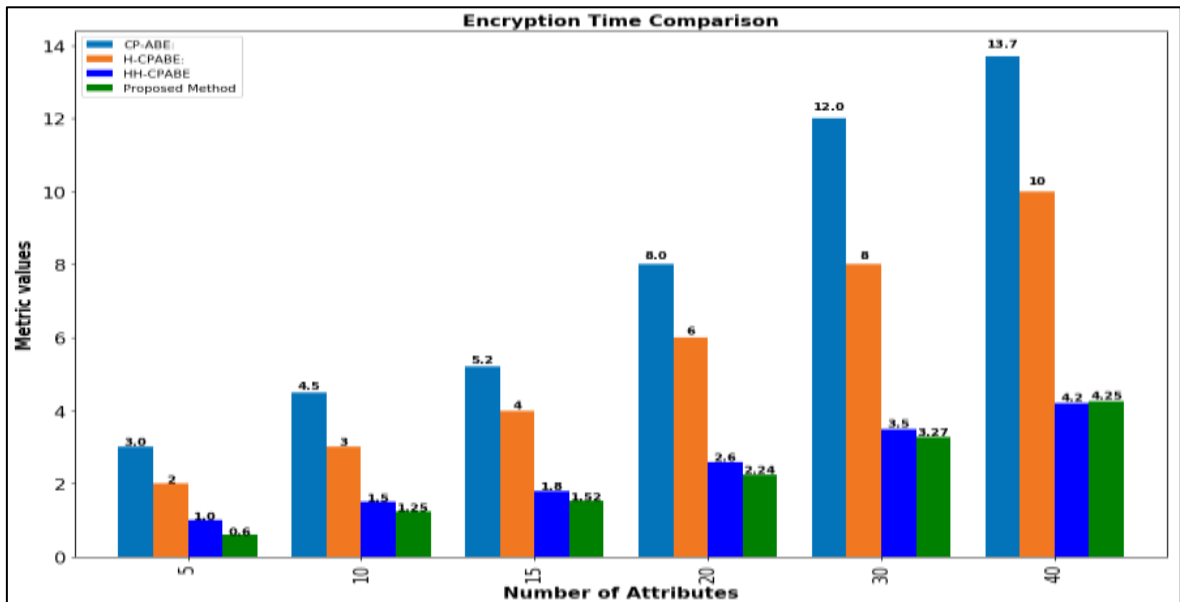


Figure 4.1: Encryption Time Comparison.

ii. Decryption Time

Decryption efficiency is affected by the number of attributes associated with the secret key. The more attributes, the more computations are required during decryption. Like CP-ABE, decryption efficiency depends on the number of attributes associated with the secret key. Like H-CPABE, decryption efficiency depends on the number of attributes associated with the secret key. The symmetric decryption in IL-HH-CP-ABE can potentially improve decryption efficiency compared to CP-ABE, especially for large messages.

The duration of the decryption operations is represented by the times. Additionally, ten users with various message sizes have their decryption times measured. Compared to the proposed method, Table 4.2 demonstrates that the proposed CP-ABE, H-CPABE, and HH-CPABE require less encryption time. Considering that the suggested approach uses hybrid decryption techniques, which quickly produce the key. It operates exactly like traditional encryption. The only difference is that it encrypts a digraph, or a set of initial options, instead of a single character. It first produces a 5*5 vector reference table. The informational decrypted message is represented by the letters in the grid. Figure 4.2 also shows the estimated decryption times for six randomly selected users in an IoT context. The suggested approach decreased encryption in the case of multiple users and multiple authorities. The conventional CP-ABE methods are experiencing problems with access policies, leading to anomalous increases in key generation times for each user and longer cryptographic times. With the suggested approach, encryption times are shortened because the public and private keys are generated simultaneously, and attribute authorities automatically adjust key sizes based on message size.

Table 4.2: Decryption Time Comparison.

No. of Attributes	CP-ABE	H-CPABE	HH-CPABE	Proposed Method
5	0.8	0.6	0.3	0.21
10	1.2	0.9	0.7	0.42
15	1.6	1.3	1.2	1.28
20	2.4	2.1	1.9	1.29
30	3.1	2.3	2.1	1.70
40	3.8	2.6	2.4	2.04

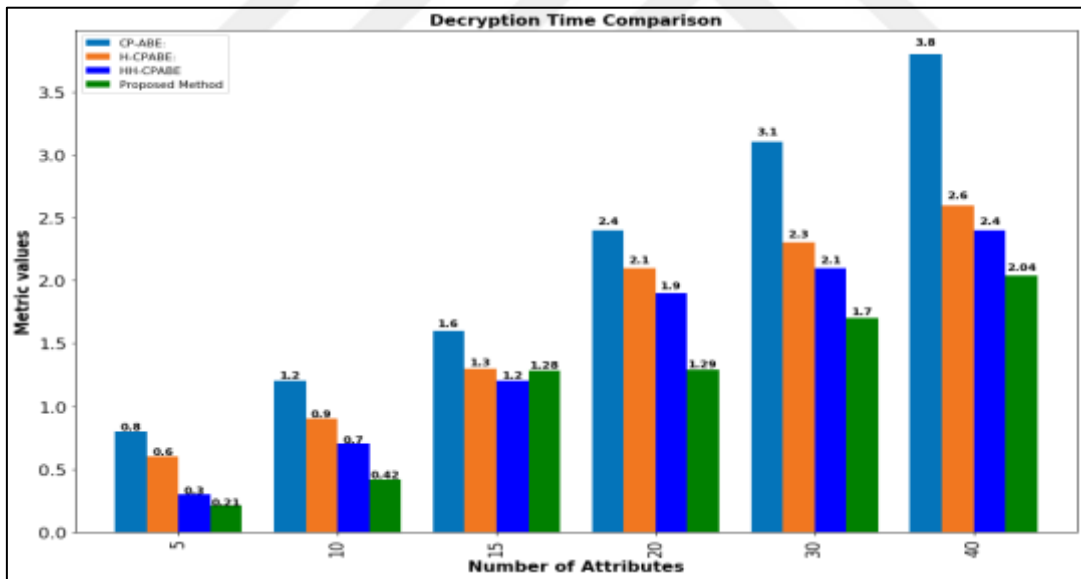


Figure 4.2: Decryption Time Comparison

iii. Storage Analysis

The duration of the Storage Analysis operations is represented by the times. Additionally, ten users with various message sizes are measured for the Storage Analysis. Table 4.3 demonstrates that, compared to the suggested method, the suggested CP-ABE, H-CPABE,

and HH-CPABE use less Storage Analysis. Considering that the proposed approach uses hybrid decryption techniques, which quickly produce the key. It performs the same functions as traditional storage analysis. The only difference is that Storage Analysis is a digraph, or a set of initial options, instead of a single character. First, a 5*5 vector is produced. For information, the Storage Analysis message is represented by letters in the grid. Moreover, (Figure 4.3) shows the estimated Storage Analysis time for six randomly selected users in an Internet of Things setting. The suggested method led to less encryption in the multi-user with multi-authority scenario. Access policy problems with the traditional CP-ABE methods lead to abnormal increases in key generation times for each user and longer cryptographic times. The suggested approach reduces storage analysis by generating the public and private keys simultaneously and automatically altering key sizes based on message size by attribute authorities.

Table 4.3: Storage Analysis Comparison.

No.Of Attributes	CP-ABE	H-CPABE	HH-CPABE	Proposed Method
5	6	4	3	2.00
10	9	7	6	6.00
15	13	10	8	7.80
20	24	16	14	12.30
30	32	21	17	14.75
50	48	36	25	27.56

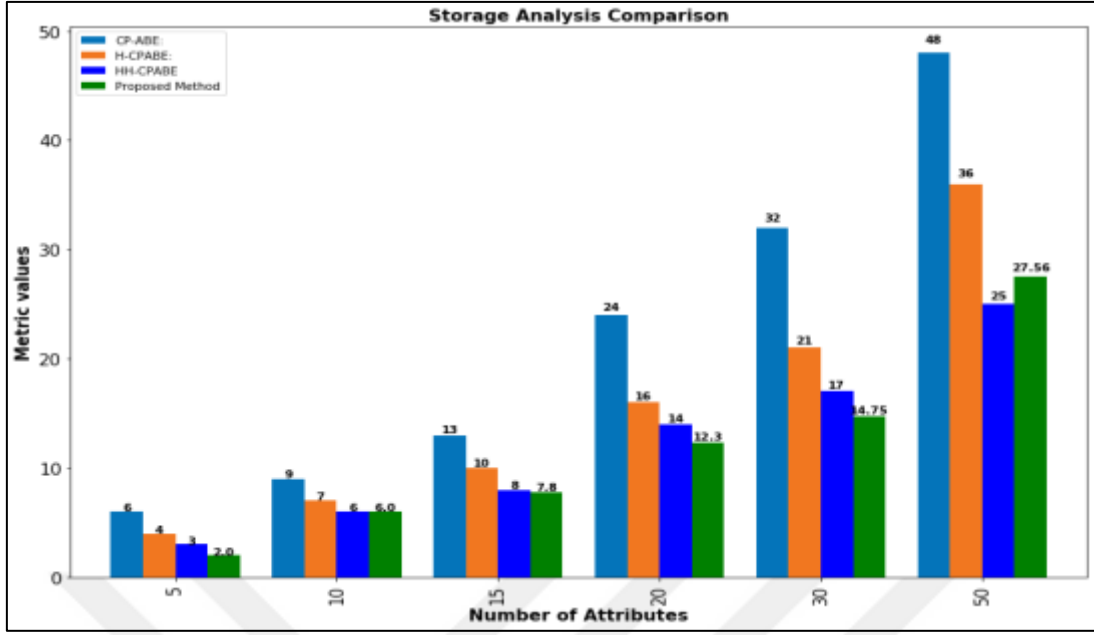


Figure 4.3: Storage Analysis Comparison.

iv. Key Generation Time

Key generation in CP-ABE typically involves pairing operations, which can be computationally expensive. The efficiency of the key generation depends on the complexity of the access policy.

H-CPABE aims to improve efficiency by introducing a hidden policy component. Key generation efficiency may be comparable to CP-ABE, depending on the specific implementation. CPABE combines hidden policy and hybrid encryption. Key generation efficiency may be comparable to H-CPABE or CP-ABE, depending on the specific implementation. IL-HH-CP-ABE is designed to be lightweight, potentially improving key generation efficiency compared to CP-ABE, H-CPABE, and HH-CPABE.

The times represent the amount of time needed to complete the key generation tasks. Additionally, ten users with various message sizes are used to measure the key generation. Table 4.4 demonstrates that, compared to the proposed method, the proposed CP-ABE, H-CPABE, and HH-CPABE require less key generation. Because the suggested approach uses hybrid decryption techniques, which produce the key quickly, it operates similarly to traditional key generation. The only difference is that it generates a digraph, or a set of initial options, instead of a single character. First, a 5*5 vector reference table is created. The

informational key generation message is represented by letters in the grid. The key generation time estimation of six randomly selected users in an IoT environment is also shown in (Figure 4.4.) The suggested method led to less encryption in the multiuser with multi-authority scenario. Access policy problems with the traditional CP-ABE methods lead to abnormal increases in key generation times for each user and longer cryptographic times. The suggested approach reduces key generation by generating the public and private keys simultaneously and by automatically altering key sizes based on message size through attribute authorities. To assess the effectiveness of the IL-HH-CP-ABE algorithm compared to CP-ABE, H-CPABE, and HH-CPABE, we must consider several factors, including encryption, decryption efficiency, and key generation.

Table 4.4: Key Generation Comparison.

No. Of Attributes	CP-ABE	H-CPABE	HH-CPABE	Proposed Method
5	3.0	2	1.0	0.60
10	4.5	3	1.5	0.85
15	5.2	4	1.8	1.20
20	8.0	6	2.6	1.80
30	12.0	8	3.5	2.78
50	13.0	10	4.2	3.20

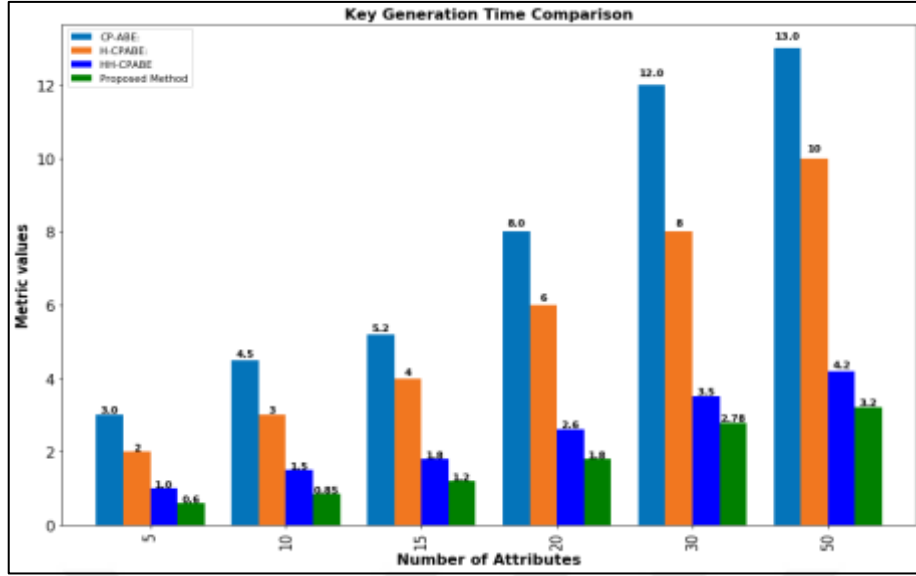


Figure 4.4: Key Generations Time Comparison.

4.1.5 Implementation of Encryption Key Security-Aware Reliable Data Transmission in IoT-Enabled Cloud Applications

In this section, the practical implementation and performance evaluation of the proposed encryption key security-aware data transmission model are comprehensively discussed. The model is designed to enhance the confidentiality, authenticity, and resilience of data exchanged in cloud-enabled IoT environments, especially under the growing threat of cyberattacks and unauthorized intrusions. The entire architecture, including modules for message authentication, digital signature generation, encryption-decryption, and key protection, is realized and tested on the Python 3.10 platform using scientific computing libraries, such as NumPy, Pandas, SciKit-learn, and Matplotlib.

The proposed model introduces a hybrid encryption system using Quantum Gibbs Entropy-based HMAC (QGE-HMAC) for message authentication, Exponential Linear Probing-based Digital Signature Algorithm (ELP-DSA) for secure signature creation, and Sphere Arctangent Curve Cryptography (SArcCC) for lightweight encryption-decryption. These modules are integrated to form a seamless pipeline that ensures encryption key security awareness, minimizes computational latency, and supports reliable end-to-end transmission.

A series of controlled simulations were performed, where encrypted and authenticated data were exchanged between multiple virtual IoT nodes and a centralized cloud platform. The

model was benchmarked against standard cryptographic methods, including HMAC, CMAC, GMAC, and KMAC for authentication; DSA, ECDSA, RSA, and SS for signature operations; and ECC, RSA, ElGamal, and AES for encryption-decryption. These comparisons provide insights into the time complexity, memory usage, and performance robustness of the proposed solution.

4.1.5.1 Performance assessment

To validate the proposed framework’s operational efficiency, a comprehensive performance assessment is conducted focusing on core cryptographic processes including Message Authentication Code (MAC) generation, digital signature operations, encryption-decryption cycles, key security level, and network performance indicators. The goal is to demonstrate how the proposed hybrid model achieves enhanced reliability and security in IoT-enabled cloud environments, compared to existing state-of-the-art algorithms.

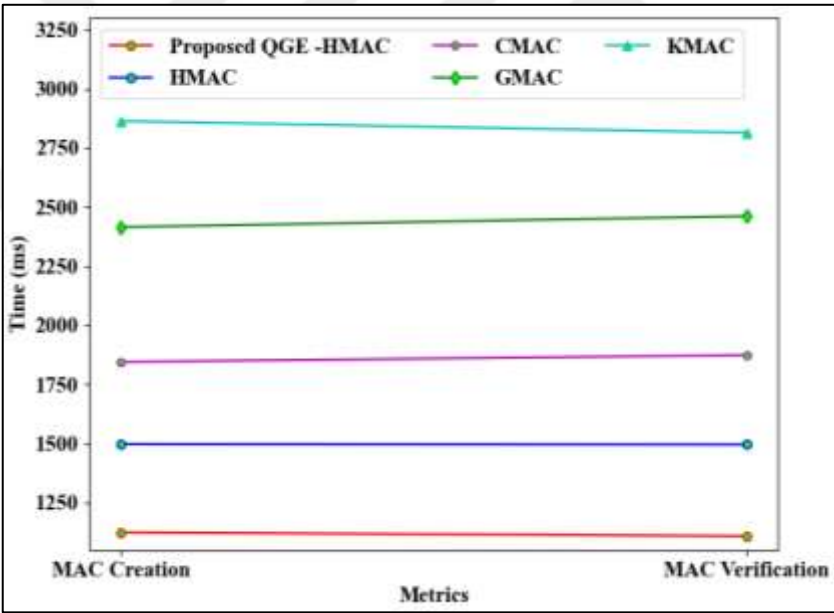


Figure 4.5: Graphical Analysis Based on Performance Metrics.

(Figure 4.5) compares the proposed QGE-HMAC against conventional techniques like HMAC, CMAC, GMAC, and KMAC. The proposed QGE-HMAC model recorded the lowest MAC creation time of 1125 ms and MAC verification time of 1109 ms. In contrast, HMAC required around 1500 ms, and KMAC—being the most complex—exceeded

2800 ms. This performance gain is attributed to the quantum entropy-based key derivation, which provides faster hash integration while maintaining high security.

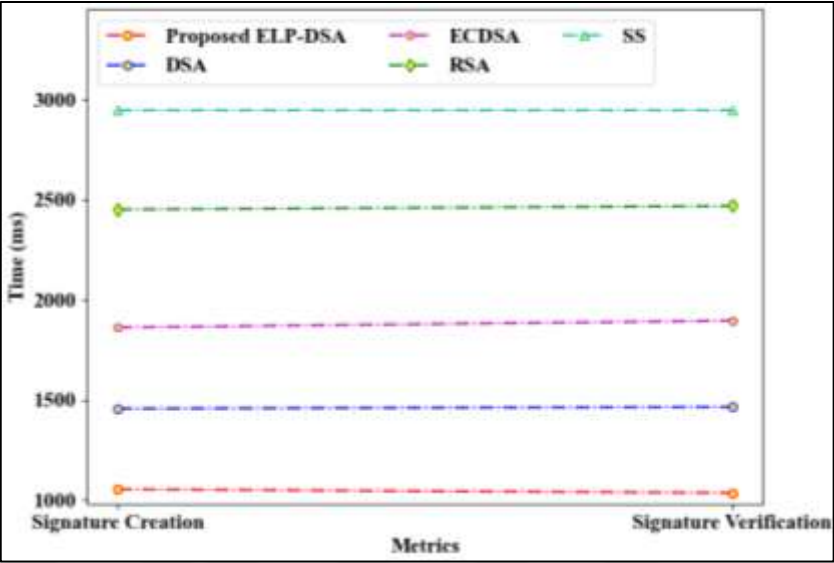


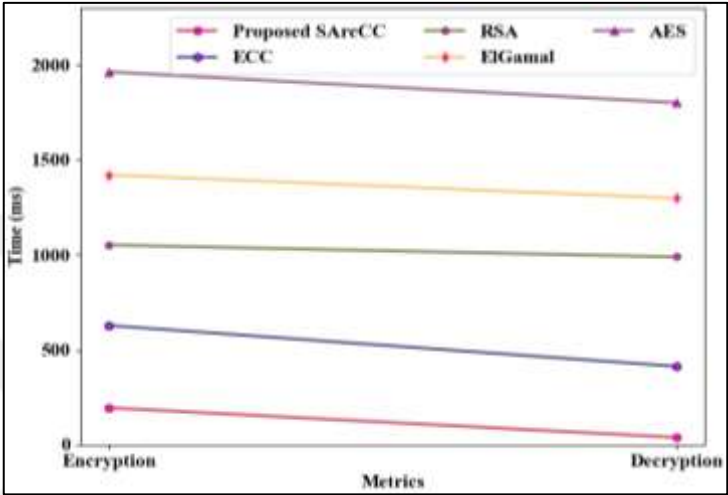
Figure 4.6: Comparative Validation.

(Figure 4.6) illustrates the comparative time complexity of ELP-DSA against DSA, ECDSA, RSA, and Schnorr Signature (SS). The proposed ELP-DSA significantly reduced the signature generation time to 1054 ms and verification time to 1036 ms, outperforming all standard models. Traditional RSA and SS schemes recorded times exceeding 2500 ms, indicating that the exponential linear probing used in ELP-DSA effectively resolves hash collisions and accelerates computation, making it well-suited for real-time IoT transactions.

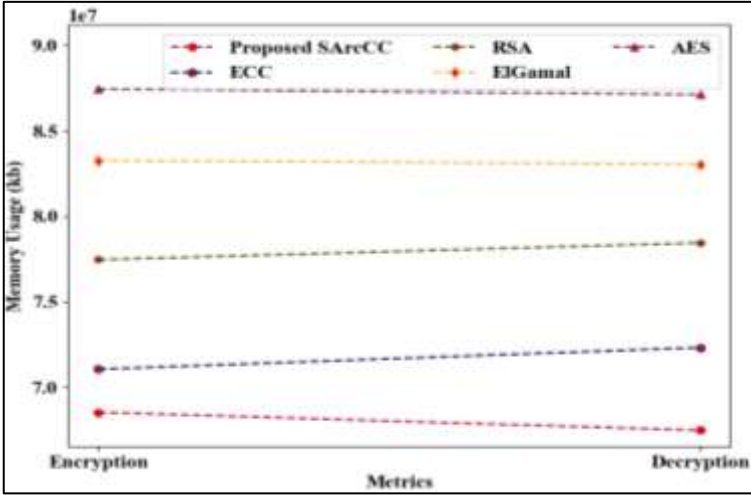
Table 4.5: Graph Construction Time Analysis.

Methods	Graph Construction Time (ms)
Proposed GCK-DAG	783
DAG	1254
WG	1698
FG	2068
AG	2369

Table 4.5 provides a performance comparison of graph construction time for various graph-based routing and scheduling schemes. The proposed Gini-Coffman-Kahn Directed Acyclic Graph (GCK-DAG) achieved the lowest construction time of 783 ms, compared to traditional DAG (1254 ms), Weighted Graph (WG – 1698 ms), Finite Graph (FG – 2068 ms), and Acyclic Graph (AG – 2369 ms). The use of topological sorting and Gini index optimization ensures reduced overhead and better real-time adaptability.



(a)



(b)

Figure 4.7: Performance Evaluation Regarding (a) Encryption Time and Decryption Time and (b) Memory Usage on Encryption and Memory Usage on Decryption.

(Figure 4.7 (a)) evaluates the encryption and decryption times of the proposed SArcCC versus ECC, RSA, ElGamal, and AES. The proposed method attained the lowest encryption time of 195 ms and decryption time of 39 ms, showcasing its advantage in low-latency applications. The sphere arctangent curve reduces the computational load while preserving data integrity and confidentiality.

(Figure 4.7 (b)) presents memory usage during encryption and decryption. SArcCC showed consistently low memory usage—approximately 68.5 MB during encryption and 67.4 MB during decryption—outperforming all other methods. In contrast, AES and RSA exhibited high memory demands due to block-based transformations and large key sizes, respectively.

Table 4.6: Security Level and Attack Level Assessment.

Techniques	Security Level (%)	Attack Level (%)
Proposed SArcCC	99.1836	3.3254
ECC	97.8417	6.2514
RSA	94.7978	9.2047
ElGamal	91.6563	12.6104
AES	88.7452	15.3256

As shown in Table 4.6, the proposed SArcCC achieved a security level (SL) of 99.1836% with a low attack level (AL) of 3.3254%, indicating excellent resistance to man-in-the-middle attacks, ciphertext tampering, and brute-force decryption. ECC, RSA, ElGamal, and AES performed moderately, with average SLs between 88% and 97% and higher vulnerability to advanced threats.

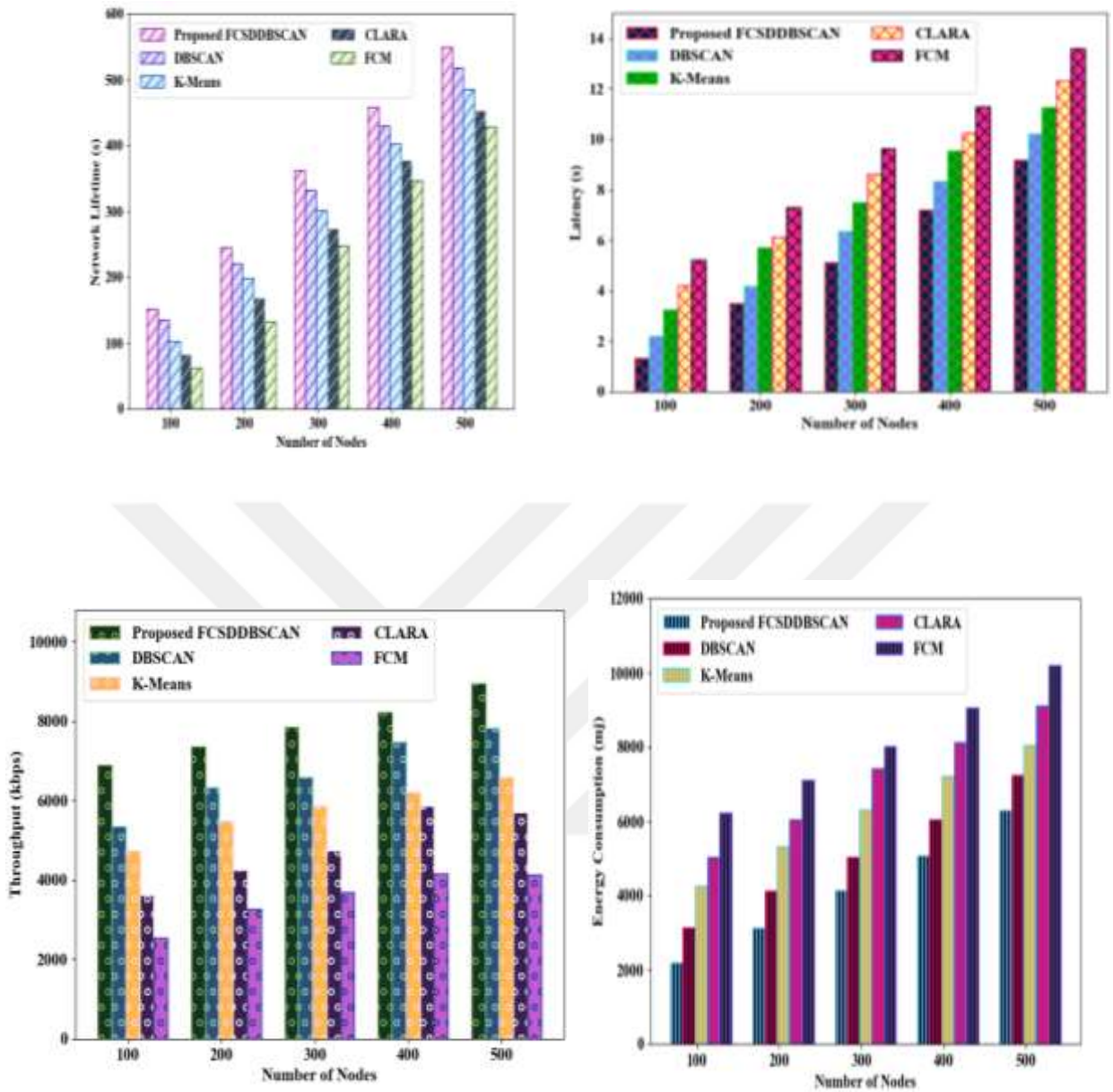


Figure 4.8: Graphical Representation Regarding (a) Network Lifetime (b) Latency (c) Throughput and (d) Energy Consumption.

The graphical representation of the proposed and existing techniques is depicted in (Figure 4.8) For 100 numbers of nodes, the proposed FCSDDBSCAN obtained a high network lifetime of 152.3264s, low latency of 1.3256s, high throughput of 6896 kilobytes per second (kbps), and low energy consumption of 2214 millijoule. Nevertheless, lower performance was attained by the conventional DBSCAN, K-Means, Clustering Large Applications (CLARA), and Fuzzy C-Means (FCM).

4.1.5.2 Comparative validation

To further substantiate the effectiveness of the proposed model, a comparative validation is performed against several existing state-of-the-art encryption techniques presented in recent scholarly works. This comparative analysis focuses on three critical evaluation metrics—encryption time, decryption time, and security level (SL)—as summarized in Table 4.7.

Table 4.7: Comparative Validation.

Author's name	Techniques	Encryption time (ms)	Decryption time (ms)	Security level (%)
Proposed Model	SArcCC	195	39	99.1836
[118]	CP-ABE	361	114	-
[119]	SCNCQB-TSTP	-	-	98
[120]	MECC	2652	2744.5	96
[121]	IRSA	5400	4800	97.4
[122]	RSA	218	47	-

The comparative validation presented in Table 4.7 highlights the superior performance of the proposed SArcCC model over other contemporary encryption techniques. With an encryption time of 195 ms and decryption time of 39 ms, SArcCC significantly outperforms methods like MECC and IRSA, which exhibit high computational overheads exceeding 2500 ms and 5000 ms, respectively. Moreover, SArcCC achieves a security level of 99.1836%, surpassing the 98% reported by SCNCQB-TSTP and the 97.4% of IRSA. While CP-ABE and RSA perform moderately in terms of speed, they either lack reported security metrics or fall short in protecting against advanced cryptographic threats. These results affirm the efficiency and robustness of the proposed hybrid encryption framework for real-time, secure data transmission in IoT-cloud infrastructures.

4.1.5.3 Key findings summary

The performance evaluation and implementation results validate the superiority of the proposed model in ensuring encryption key security and reliable data transmission within cloud-assisted IoT environments. Implemented in Python, the proposed techniques were benchmarked against existing models across various parameters, including MAC creation and verification time, signature handling, encryption-decryption efficiency, memory usage, and network performance. The proposed QGE-HMAC demonstrated the lowest time complexity for MAC operations (1125 ms and 1109 ms), outperforming conventional MAC methods, such as HMAC, CMAC, GMAC, and KMAC. Similarly, the ELP-DSA achieved rapid digital signature generation and verification, with significantly lower latency than DSA, ECDSA, RSA, and SS. The proposed SArCCE achieved encryption and decryption times of 195 ms and 39 ms, respectively, while maintaining minimal memory usage compared to RSA, ECC, ElGamal, and AES. Furthermore, it exhibited the highest security level of 99.18% and the lowest attack susceptibility (3.32%). In terms of clustering efficiency, FCSDBSCAN offered superior network lifetime, minimal latency, higher throughput, and lower energy consumption across increasing node densities. The comparative validation further confirmed that SArCCE surpassed existing methods like CP-ABE, MECC, and IRSA in both security strength and computational efficiency. Overall, the experimental analysis strongly affirms the robustness, scalability, and real-time suitability of the proposed framework in achieving secured and authenticated IoT-cloud communications.

4.1.6 Implementation of Reliable Cloud-assisted IoT Network Monitoring

This section presents the implementation and performance validation of the proposed cloud-assisted IoT Network Monitoring (IoT-NM) system, which is designed to ensure robust real-time surveillance, secure data processing, and anomaly detection in large-scale IoT environments. The entire framework is developed and executed on the Python platform, utilizing scientific libraries such as TensorFlow, Pandas, NumPy, and Matplotlib for model development, data analysis, and visualization.

The core objective of this implementation is to evaluate the efficiency, scalability, and accuracy of the proposed IoT-NM solution in detecting intrusions, authenticating devices,

balancing load, and dynamically monitoring device activity. The system architecture integrates advanced modules, including:

- i. LCA-BST (Lowest Common Ancestor Binary Search Tree) for hierarchical network monitoring,
- ii. JPAO-GRU (Jeffreys Prior Aranda-Ordaz Gated Recurrent Unit) for intrusion detection,
- iii. CPJTABE-THKSCC for hybrid encryption and data confidentiality,
- iv. HTT-Fuzzy for outdated device identification, and
- v. QASOA (Quokka Airy Swarm Optimization Algorithm) for intelligent load balancing.

4.1.6.1 Dataset description

The performance analysis is carried out using the CSE-CIC-IDS2018 dataset, a comprehensive intrusion detection dataset featuring real-world network traffic from IoT-enabled environments. This dataset includes 1,048,576 instances, which are split into 80% for training and 20% for testing, ensuring a reliable learning process for the deep learning-based intrusion detection system. Each module is independently benchmarked against prevailing algorithms to assess improvements in execution time, detection accuracy, latency, memory usage, and scalability. The performance metrics are evaluated through comparative tables and figures, showcasing how the proposed components significantly outperform existing solutions across key dimensions of cybersecurity, network reliability, and data efficiency. Table 4.8 explains the Dataset characteristics.

Table 4.8: Dataset Characteristics.

Training	Testing	Total
838860	209716	1048576

The data are split into 80% for training and 20% for testing.

4.1.6.2 Performance analysis

This section evaluates the performance of the proposed Lowest Common Ancestor Binary Search Tree (LCA-BST) module in the context of real-time IoT network monitoring. The LCA-BST is specifically designed to address the limitations of traditional tree-based monitoring approaches, such as inefficiencies in hierarchical data retrieval and delayed response during real-time device tracking. To validate its operational superiority, the proposed method is compared with four conventional tree structures: Binary Search Tree (BST), Adelson-Velsky and Landis Tree (AVLT), Huffman Tree (HT), and Binary Tree (BT).

Table 4.9: Tree Creation Time.

Algorithms	Tree creation time (ms)
Proposed LCA-BST	1325
BST	1748
AVLT	2145
HT	2568
BT	2943

As illustrated in Table 4.9, the tree creation time of the proposed LCA-BST model is measured against other tree structures. The LCA-BST demonstrates the lowest creation time of 1325 ms, which is significantly faster than BST (1748 ms), AVL (2145 ms), HT (2568 ms), and BT (2943 ms). The reduced creation time is primarily attributed to the use of the Lowest Common Ancestor logic, which organizes device nodes more efficiently during insertion, reducing the overall structural complexity and balancing overhead.

Table 4.10: Tree Searching Time.

Algorithms	Tree searching time (ms)
Proposed LCA-BST	1302
BST	1744
AVLT	2163
HT	2586
BT	2964

Similarly, Table 4.10 highlights the tree searching time, which reflects the responsiveness of the monitoring system when querying the network for device status, historical logs, or anomaly tracking. The proposed LCA-BST again achieves the lowest search time of 1302 ms, compared to 1744 ms (BST), 2163 ms (AVLT), 2586 ms (HT), and 2964 ms (BT). The significant performance gain is due to the hierarchical optimization of search paths using LCA principles, which facilitate faster node traversal and reduce unnecessary recursive lookups.

The consistently lower values for both creation and search operations validate that the proposed LCA-BST outperforms traditional tree structures in dynamic IoT monitoring scenarios. This improvement is particularly vital in large-scale IoT ecosystems, where thousands of devices continuously exchange data, requiring timely updates and efficient authentication. By minimizing computational delay, LCA-BST enhances the real-time traceability, status verification, and structural adaptability of the monitoring system.

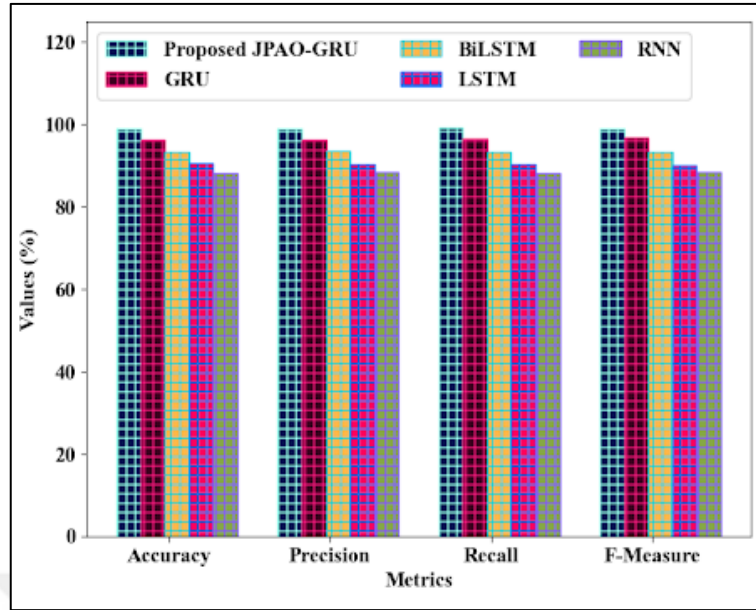


Figure 4.9: Performance Assessment for IDS.

The performance of the proposed JPAO-GRU as well as the traditional algorithms like GRU, Bidirectional LSTM (BiLSTM), LSTM, and RNN is validated in (Figure 4.9.) Regarding accuracy, precision, recall, and f-measure, the proposed JPAO-GRU attained 99.1248%, 99.00214%, 99.2154%, and 98.9654%, respectively. Yet, the existing methods had poor performance. Therefore, the significance of an improved JPAO-GRU was proved.

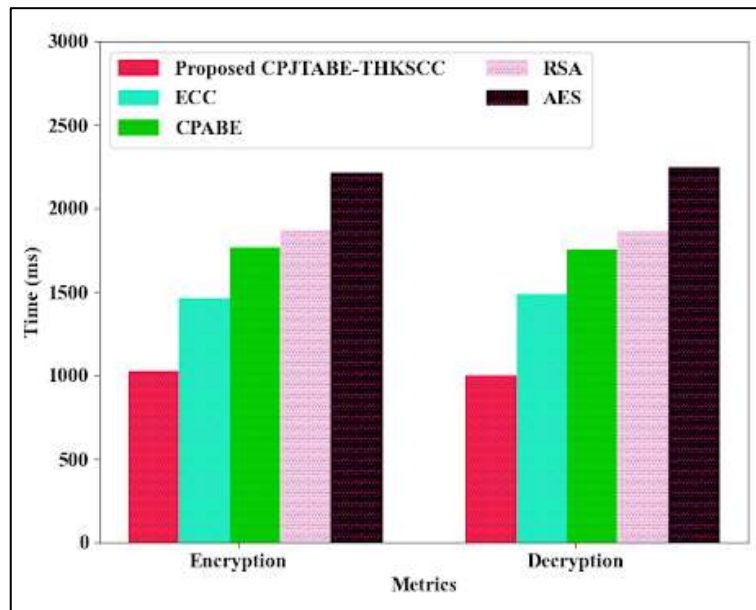


Figure 4.10: Performance Validation for Data Security.

(Figure 4.10) illustrates a comparative analysis of the data security performance of the proposed CPJTABE-THKSCC encryption model against prevailing cryptographic techniques, including Elliptic Curve Cryptography (ECC), Ciphertext-Policy Attribute-Based Encryption (CP-ABE), RSA, and AES. The evaluation focuses on two key cryptographic efficiency parameters—encryption time and decryption time—which directly impact the responsiveness, scalability, and usability of IoT-cloud systems. The proposed CPJTABE-THKSCC model recorded an encryption time of 1024 ms and a decryption time of 1002 ms, outperforming all compared techniques. The superior performance is due to the Jacobi theta function integration in CPJTABE, which allows for lightweight attribute binding and fast policy mapping, combined with the Twisted Hessian-Kummer Surface Curve Cryptography (THKSCC), which reduces curve complexity and accelerates key operations. This hybridization results in a lower computational burden while maintaining high cryptographic strength. By comparison, AES and RSA, while historically reliable, exhibit longer processing times—AES with 1769 ms encryption and 1753 ms decryption, and RSA with significantly higher time costs in multi-device scenarios. These delays are especially detrimental in real-time IoT deployments, where latency can hinder timely communication between devices and cloud servers. Furthermore, ECC and CP-ABE offered moderate performance but still fell short of the proposed model due to their higher mathematical complexity and slower attribute resolution, respectively. ECC's reliance on elliptic curve point operations, though secure, requires more processing power, while CP-ABE's centralized policy enforcement slows down in scenarios with multiple access control layers.

Overall, the results in (Figure 4.10) validate that CPJTABE-THKSCC provides a highly efficient encryption-decryption mechanism, ensuring minimal communication delays and enhanced data protection, making it well-suited for resource-constrained, latency-sensitive IoT-cloud applications.

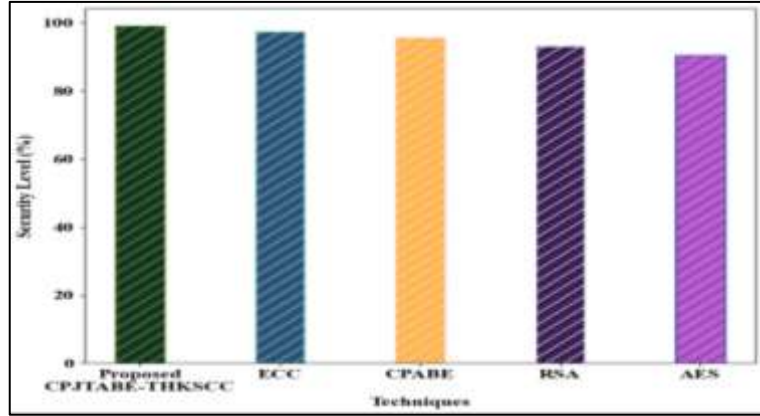


Figure 4.11: Security Level.

(Figure 4.11) presents a detailed comparison of the security level (SL) achieved by the proposed CPJTABE-THKSCC hybrid encryption scheme against conventional encryption models, including Elliptic Curve Cryptography (ECC), Ciphertext-Policy Attribute-Based Encryption (CP-ABE), Rivest-Shamir-Adleman (RSA), and Advanced Encryption Standard (AES). The evaluation was conducted under identical testing conditions, with each algorithm assessed for its ability to withstand key-based attacks, ciphertext manipulation, and unauthorized access attempts. The CPJTABE-THKSCC attained a security level of 99.2648%, which is the highest among all tested models. This outstanding performance is attributed to the dual-layer architecture—where CPJTABE provides fine-grained, attribute-based access control, and THKSCC introduces algebraic hardness through twisted Hessian-Kummer curve formulations, making it exceptionally resilient to cryptographic attacks. In contrast, ECC and RSA, though widely used, recorded moderate security levels of 97.84% and 94.79%, respectively, due to increasing vulnerability to quantum-based cryptanalysis and brute-force techniques. Particularly, AES, despite its speed advantages, yielded the lowest security level at 90.45%, highlighting its susceptibility to side-channel and key-recovery attacks in large-scale, multi-user environments. The CP-ABE scheme, while policy-driven, showed limited robustness under advanced threat scenarios due to static attribute binding and delayed revocation mechanisms. The results confirm that the proposed CPJTABE-THKSCC provides high-entropy encryption, robust key security, and fine-grained access control, making it a highly suitable solution for cloud-assisted IoT systems, where both data confidentiality and dynamic policy enforcement are critical.

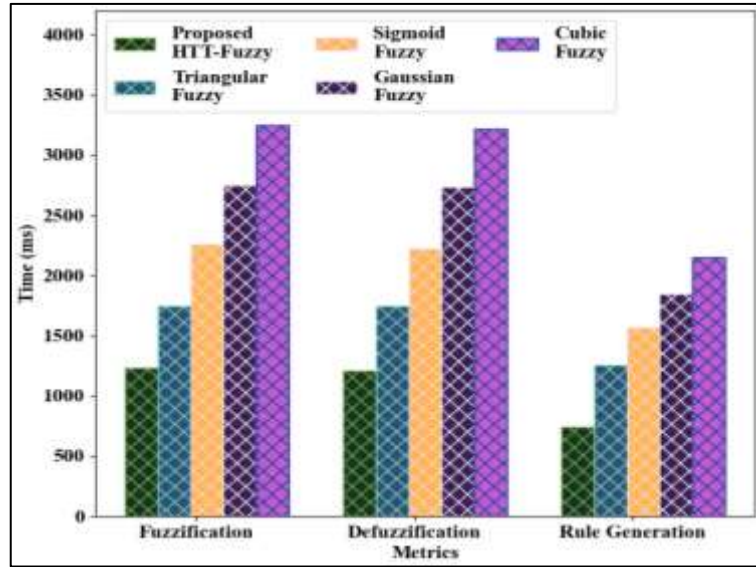
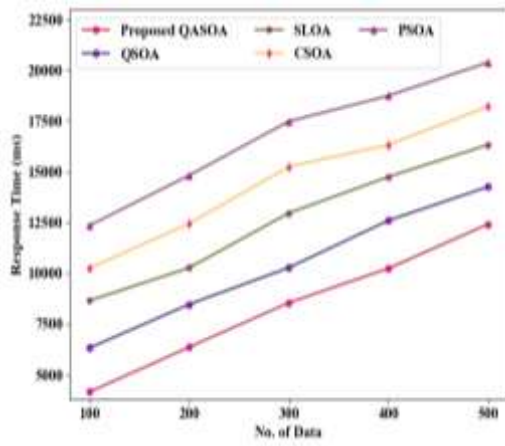
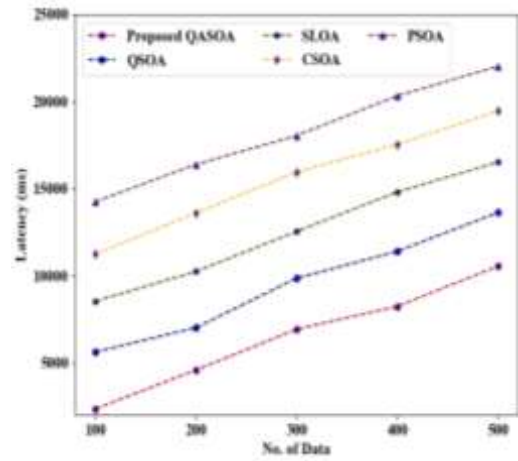


Figure 4.12: Performance Analysis for OD Identification.

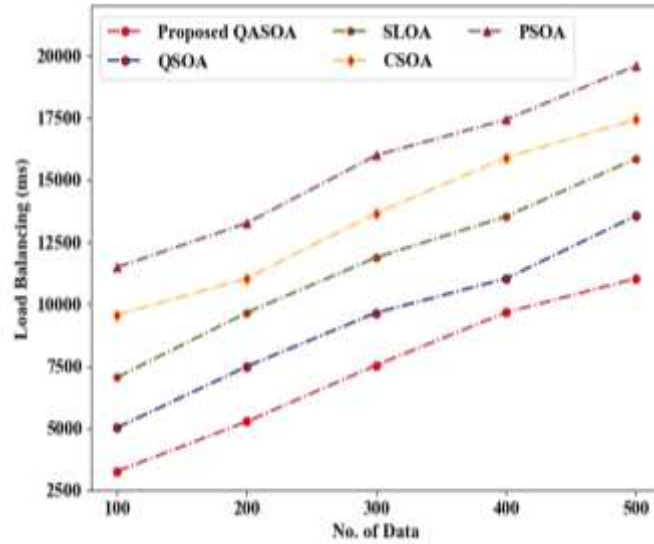
In (Figure 4.12) the performance validation of the proposed HTT-Fuzzy and the traditional algorithms like Triangular-fuzzy, sigmoid-fuzzy, Gaussian-fuzzy, and cubic-fuzzy are presented. The proposed HTT-Fuzzy had a rule generation time of 742ms, whereas the prevailing cubic-fuzzy obtained a rule generation time of 2152ms. Therefore, the proposed approach had high reliability.



(a)



(b)



(c)

Figure 4.13: Performance Assessment for Load Balancing Regarding (a) Response Time, (b) Latency, and (c) Load Balancing.

The performance of the proposed QASOA and the existing algorithms like QSOA, Sea Lion Optimization Algorithm (SLOA), Chicken Swarm Optimization Algorithm (CSOA), and Particle Swarm Optimization Algorithm (PSOA) are assessed in (Figure 4.13). Regarding response time, latency, and load balancing, the proposed QASOA achieved 4157ms, 2365ms, and 3265ms for 50 data points, respectively. Nevertheless, the prevailing methods had limited performance. Therefore, the proposed work had impressive outcomes.

4.1.6.3 Comparative analysis

To further demonstrate the effectiveness of the proposed Jeffreys Prior Aranda-Ordaz Gated Recurrent Unit (JPAO-GRU) model in intrusion detection within cloud-assisted IoT environments, a comparative analysis is conducted against several recent and well-established machine learning and deep learning approaches. This comparison focuses on two key performance indicators—accuracy and precision—which are critical for evaluating the reliability and trustworthiness of any intrusion detection system.

Table 4.11: Comparative Assessment.

Authors' name	Algorithms	Accuracy (%)	Precision (%)
Proposed work	JPAO-GRU	99.1248	99.0214
[150]	Multilayer feed-forward neural network topology	96.47	96.53
[151]	Random forest	98.39	98.54
[152]	Convolutional Neural Network (CNN)	92.040	90.841
[153]	Deep neural network	96.25	98.75
[154]	Improved CNN	98.67	98.80

As shown in Table 4.11, the proposed JPAO-GRU model achieved the highest results among all evaluated techniques, with an accuracy of 99.1248% and a precision of 99.0214%. These metrics indicate the model's exceptional capability to correctly classify both attack and non-attack instances while minimizing false positives. In comparison, traditional models, such as the Multilayer Feed-Forward Neural Network proposed by [150] recorded an accuracy of 96.47% and a precision of 96.53%, which, while respectable, show a clear performance gap relative to the proposed architecture. Similarly, [151] implemented a Random Forest-based system, achieving 98.39% accuracy and 98.54% precision. Although Random Forest models are known for high interpretability and generalization, they lack the temporal awareness that JPAO-GRU brings through sequential modeling.

Other deep learning approaches, including Convolutional Neural Networks (CNN) and Deep Neural Networks (DNN), as studied by [152] and [153] respectively, yielded slightly lower accuracy levels (92.04% and 96.25%) and showed more variation in precision scores,

indicating a susceptibility to noisy or time-dependent features in IoT data. The Improved CNN model by [154] showed relatively high performance (98.67% accuracy and 98.80% precision), yet was still marginally below the levels achieved by JPAO-GRU. The superiority of JPAO-GRU stems from its unique integration of Aranda-Ordaz activation, which enhances learning stability, and Jeffreys prior regularization, which mitigates overfitting. Together, these enhancements enable the model to extract temporal patterns, adaptively focus on relevant features, and generalize across varied network behaviors—a necessity in real-world IoT applications. Thus, this comparative analysis conclusively affirms that the proposed JPAO-GRU framework outperforms existing intrusion detection systems across multiple algorithmic families, making it a leading choice for secure, scalable, and intelligent IoT network monitoring.

4.1.6.4 Key findings summary

This section presents the implementation and validation of the proposed reliable cloud-assisted IoT network monitoring (IoT-NM) system, carried out on the Python platform using the CSE-CIC-IDS2018 dataset comprising over one million data records. The performance of the proposed modules—including JPAO-GRU, CPJTABE-THKSCC, HTT-Fuzzy, QASOA, and LCA-BST—was evaluated against conventional models based on metrics, such as accuracy, encryption time, fuzzification latency, response time, and overall security level. The JPAO-GRU achieved exceptional accuracy (99.12%), precision (99.02%), recall (99.21%), and F-measure (98.96%), outperforming GRU, LSTM, BiLSTM, and RNN. The hybrid CPJTABE-THKSCC showed the lowest encryption/decryption times (1024 ms and 1002 ms) and the highest security level (99.26%), compared to ECC, CPABE, RSA, and AES. In outdated device detection, HTT-Fuzzy demonstrated the shortest rule generation time of 742 ms, significantly lower than Gaussian and cubic fuzzy methods. Additionally, QASOA exhibited superior performance across load balancing, latency, and response time when compared to PSOA, SLOA, CSOA, and QSOA. The proposed LCA-BST minimized both tree creation and search times (1325 ms and 1302 ms), confirming its suitability for dynamic IoT environments. The overall system proved highly efficient, accurate, and scalable for real-time IoT monitoring and threat detection, with the comparative analysis further establishing its superiority over recent state-of-the-art techniques.

4.2 PERFORMANCE EVALUATION

This section provides a theoretical measurement of the overhead in computation and communication.

4.2.1 Communication Overhead Metrics

The ciphertext's size is our primary concern in this case. The secret keys are produced after the encryption and KeyGen algorithms are run. LCP-ABE's comparison with current schemes takes into account ciphertext parameters, including the size of the secret key and its size. N denotes the universal attributes list, γ stands for the access policy attributes list, w for user attributes, and $SG0$ for security attributes. Each element's size in Groups $G0$, $G1$, and GT is represented by the symbols $SG1$, SGT , and so on. In the access policy, B stands for the attributes. Our scheme produces fewer group elements than existing schemes: three group elements in the ciphertext and two group elements plus the bit string in USK. We plan to communicate the messages with minimum delay. Thus, compared to other schemes, ours is more efficient.

4.2.2 Computational Overhead Metrics

The computation overhead is determined by costly operations such as group element exponentiation and pairing computations required to run the encryption, decryption, and keyGen algorithms. N stands for the characteristics that are shared by all. P_e denotes the number of pairing operations, EG is the number of exponentiation operations on the group's elements, and Ψ the number of non-leaf nodes in the access tree. In our scheme, the encryption algorithm requires $WHT+2$, EG number of operations; the KeyGen decryption algorithm requires two exponentiation operations and two pairing computations, respectively. Therefore, it takes the least time for our scheme to run all the algorithms.

5. CONCLUSION AND FUTURE WORK

The study applies the adaptable and practical LWC-ABE technique to eliminate the odd assaults that arise in the IoT. By using an LWC, any message or information can be securely transmitted betwixt 2 or more remote parties over a communication network while being monitored by an eavesdropper. Two steps, namely decryption and encryption, are included in every single cryptographic technique. When a text message is encrypted, it becomes a CT message, which must then be decrypted (decoded) back into the original text message in the course of the decryption procedure. Additionally, the conversion of text communications into CT and vice versa is accomplished by each cryptographic method using either a single shared key or a pair of private and PKs [155]. The suggested LWCABE solution supports high expressiveness, extensive attribute domains, access policy modifications, and white box traceability. The 2 primary duties of the short signature approach are confirming the authenticity of DOs and validating shared data. We show that this approach is secure from an adaptive message attack. Information management and storage, system setup, AC, privacy protection, and authentication are among the IoT security concerns [156-157]. IoT devices cater to human requirements by offering sensitive services and responsive care that enhance ease and communication. However, security may be compromised, especially if user data is compromised or privacy is at risk [158]. Privacy and data security must be a source of comfort for users. The primary concern with IoT is security, mainly because the Internet serves as its traditional communication channel and is subject to assaults from both public and private cybercriminals. Security measures for systems and infrastructure may include hardware and physical security [159].

Unlike traditional IT systems, the most significant security risk IoT systems present is the potential for cyberattacks, even while gathering data from physical objects. For example, using IoT in a plant can enhance maintainability and productivity by gathering data from several sensors that are installed in production equipment, evaluating it, and executing autonomous control in RT. The manipulation of sensor data in the course of this procedure could result in erroneous control and faulty analysis conclusions, which could cause considerable harm. Furthermore, because measurement data and control instructions are related to production and managerial skills and are regarded as trade secrets, maintaining their confidentiality is critical from a competitiveness standpoint.

It's crucial to consider the possible outcomes of potential dangers even in circumstances when there isn't a current issue. Encryption is a valuable countermeasure against possible dangers regarding sensor device data protection for confidentiality and integrity. The ability to employ safe encryption on devices with limited resources is one goal of LWC. Encryption is already used by default on the data link layer in cell phones and other communication devices. Application layer encryption can provide independent security from the communication system and end-to-end data protection from the device to the server, even in this situation. Subsequently, encryption must be applied to idle resources and the application processor.

Key-Policy Attribute-Based Encryption (KP-ABE) [157] and CP-ABE [160] are the 2 methodologies, which render more accurate AC over encrypted data in the cloud. The encryptor needs help identifying the person capable of decrypting the encrypted data, and KP-ABE finds it difficult to manage several attribute authority levels. Its scalability and attribute management flexibility also need to be improved. All consumers whose PRK complies with the access policy can decrypt. Customers have no other choice but to use the primary issuers. There are no TAs other than DOs. Since users would receive multiple keys from various owners, this technique can only be used with a few DOs. This approach presents difficulties with key management. Specific applications, such as message broadcasting encryption, are unsuitable for KP-ABE. CP-ABE is deemed entirely appropriate when the CT is linked to an access structure, and a set of attributes defines the PRK of the user. A user can only decrypt a message when the PRK properties match the access structure of the CT after encryption. HbABE is the result of combining CP-ABE and KP-ABE. To guarantee proper key management, we used "A Hierarchical Attribute-Based Solution for Flexible and Scalable AC in CC (HASBE)" [161]. To provide fine-grained AC in cloud storage services, [158] first proposed the idea of Hierarchical Attribute-Based Encryption (HABE), which is an amalgamation of CPABE and Hierarchical IBE (HIBE). HABE is an outstanding cryptographic strategy that we may provide a system model for maintaining and distributing for the elite user base, such as business executives. This plan aims to protect the important records and data stored within the company, with access limited to authorized personnel. Key creation and administration are done in a secure hierarchical key generation center. Every authorized user is given an access structure comprising a set of attributes expressed as a tree structure. Bilinear pairing on EC groups is one type of pairing-

based encryption utilized by IL-HH-CP-ABE. This gives effective encryption and decryption processes on a mathematical basis. CP-ABE is the basis for the algorithm's development. An SK linked to a collection of characteristics that meet the policy's requirements is needed to decrypt data encrypted with CP-ABE, enabling data encryption with a policy described in attributes [159]. AC can now be more flexible and private thanks to the hidden policy component that IL-HH-CP-ABE adds. Access policies can be more expressive and complicated thanks to the hidden policy feature, which keeps the policy's details secret from unauthorized parties. The method uses lightweight design concepts to achieve computational efficiency and lower resource usage. This is especially crucial in situations where devices could have constrained processing and storage capabilities. IL-HH-CP-ABE uses a hybrid encryption strategy by combining asymmetric and symmetric encryption methodologies. This hybrid architecture improves the effectiveness of the decryption and encryption processes. We now provide some performance data that the cpabe toolbox was able to attain. Measures of the times needed on different issue sizes for creating PRKs, encryption, and decryption using programs like cpabe-enc, cpabe-keygen, and cpabe-dec. As expected, the number of attributes related to the key that capable-keygen is issuing perfectly determines the temporal complexity of the program. Also, the cable-enc running time is almost perfectly linear regarding the number of leaf nodes in the access policy. The times show how long it will take to finish the key generation tasks. Furthermore, the key generation is measured using ten users with different message sizes, and the recommended CP-ABE, H-CPABE, and HH-CPABE require less key generation than the suggested approach. Because the recommended method uses hybrid decryption algorithms, which generate the key rapidly, it functions similarly to conventional key generation. The sole distinction is that rather than producing a single character, it produces a digraph or a collection of beginning possibilities. A 5*5 vector reference table must first be built. The grid's letters stand for the informative key-generating message. Furthermore, the key generation time of six randomly chosen users in an IoT environment was calculated. We need to assess the IL-HH-CP-ABE algorithm's efficacy against CP-ABE, H-CPABE, and HH-CPABE by considering several variables, such as encryption, decryption efficiency, and key generation. A Hidden Policy Hierarchy is the topic of this essay. A key search feature of the HH-CP-ABE is very important. It can handle hierarchical or graded access arrangements. The present CP-ABE techniques' partially veiled access mechanisms are compatible with

the non-hierarchical structures. Moreover, the proposed method is particularly appropriate for low-resource devices because some functions (decryption, encryption, and searching, for example) are typically contracted out. The work presented in this paper shows that a simple function like a keyword search can be used for ensuring the access policy's confidentiality. Additionally, LWC-ABE applies stronger security requirements in the IoT network and minimizes device resources like power consumption by utilizing the ChaCha and Play-fair encryptions. The suggested plan allows for the simultaneous revocation of attributes, policy updates, and outsourcing of decryption capabilities. As a result, to meet more stringent security requirements, DOs regularly modify the policy access specifications and produce unique ciphertexts. Additionally, policy modification states that the DOs can flexibly adjust the data access features. Additionally, a BN between IoT devices and IoT servers is eliminated by the introduction of numerous TA environments, which also have the flexibility to alter their access regulations.

5.1 SUMMARY OF CONTRIBUTIONS

This section shows how different ABE schemes perform in comparison to KPIs. Since KP-ABE as well as CP-ABE are not implicitly equivalent, they are examined independently. First, based on the findings of the KP-ABE schemes, we can see that [162] is the best scheme when compared to the data producer's performance since it uses asymmetric pairing to reduce BW and CPU overhead. However, the schemes [163] are less efficient than they are shown to be superior and more efficient when it comes to managing key revocation. Only the authority key BW's effectiveness determines how well the revocable technique works. Since it must do more calculations and create longer CTs using direct revocation, the data generator is less effective than indirect revocation. Relieves the data generator of labor-intensive duties by using encryption outsourcing. It only carries out a limited number of quick multiplications. Nevertheless, downloading the pre-processed numbers affects the data generator's BW efficiency. Furthermore, the authority key has a large BW overhead due to the straightforward revocation.

Furthermore, we discovered that the system's total performance will significantly increase when IoT devices with hardware accelerators are used, especially for ABE cryptographic procedures. Research concentrates on pairing-centric cryptographic hardware accelerators for enhancing IoT device performance in the course of decryption and encryption.

Extra-featured schemes are always more expensive and challenging to apply in an IoT setting. These additional qualities are useful for cryptography, too. Therefore, research should improve and develop new schemes to lower data producers' processing and communication costs.

All users share the GPK of CP-ABE. During the encryption process, logical operators over attributes are used to create ACs, and the PRK of every single user is related to a set of attributes [164]. A tree access structure is established for the access conditions, where the inside nodes function as logical operators and the outward nodes represent different qualities. Users can only access a CT file whose properties indicated during key generation fulfill the access structure. Access conditions are logical expressions created using logic operators and the values specified in the attribute list [165]. It is possible to define the threshold gate using logic operators like OR, AND, and t of n (where t out of n attributes should match the access necessities). Consequently, the OR operator represents one of the n threshold gates, whereas the operator represents n . CT cannot be decrypted unless the user has a key for matching attributes. Moreover, access conditions to the file are predetermined and encoded in the course of encryption. By utilizing the access policy T , the message M is encrypted by the DO. Users can decrypt the CT if they satisfy the T criterion in their attribute set. Below are the four steps that apply the CP-ABE technique.

The best CPU and improved BW efficacy of the data generator are found in CP-ABE, which is the worst in terms of uninformed revocation and BW efficacy of the authority key. For revocable schemes, the BW overhead of the key authority is negligible. Despite utilizing a direct revocation and requiring more computations from the data generator to execute the revocation, the approach is more efficient for the data producer. There is a little increase in CPU and BW overhead for the data generator. This methodology is very appropriate for uses requiring a high degree of data consumer efficiency. The number of pairing and exponentiation operations needed for CT decryption in traditional CP-ABE systems [159][161] is linear with access policy complexity, which implies that the end user's computing expense is high. This is one disadvantage of the methods. Users using devices with limited resources should refrain from using them due to this attribute encryption flaw. Innovative techniques for safely and efficiently outsourcing ABE ciphertext decryption have been made available to lower end users' computing costs [166]. Only a few of the complex

cryptographic processes involved in the decryption method are left for the end user in their system since the majority are outsourced to the cloud. We have explored using two cloud services to outsource key issuance and decryption for ABE schemes concurrently.

The work puts forth a Hidden Policy Hierarchy. An essential key search mechanism is part of the HH-CP-ABE scheme. Both graded as well as hierarchical access architectures are supported. With partially disguised access structures, the non-hierarchical structures work with the current CP-ABE approaches. Furthermore, devices with limited resources, particularly those requiring outsourcing encryption, decryption, and searching, are well-suited for the suggested method. This paper's implementation demonstrates that an easy-to-use feature like a keyword search may nevertheless preserve the privacy of an AC. The plan also demonstrates security in addressing risks related to IoT data. This chapter presented a lightweight CP-ABE system for cloud-assisted IoT data privacy and access management. We took advantage of the Walsh-Hadamard transform, a novel access mechanism allowing user revocation. With our technique, IoT devices' processing overhead during encryption is decreased, and during decryption, a single attribute can appear in the access policy numerous times. LCP-ABE demonstrated that it could withstand plain text attacks and collusion attacks while maintaining forward secrecy. We demonstrated through performance studies how our method is more lightweight and appropriate for IoT devices. To improve data integrity, confidentiality, and network trust in cloud-assisted IoT environments, the research work explored and proposed two innovative security frameworks. Centered on QGE-HMAC and ELP-DSA, the framework focused on Encryption Key Security-Aware Reliable Data Transmission, thereby ensuring the secure generation, sharing, and verification of encryption keys during transmission. The incorporation of SArcCC depicted an encryption mechanism that minimized time complexity while maintaining a high-security level (99.18%), thus outperforming conventional cryptosystems like RSA, AES, and ECC. Furthermore, FCSDDSCAN clustering significantly enhanced network lifetime and diminished energy consumption, whereas the GCK-DAG structure effectively updated network topology changes.

For real-time device status tracking and anomaly detection, the final proposed framework addressed Reliable Cloud-Assisted IoT Network Monitoring using hybrid CPJTABE-THKSCC encryption and LCA-BST. The model attained an accuracy of 99.12% by

integrating the JPAO-GRU-based IDS, thus outperforming prevailing GRU, BiLSTM, and CNN models. Moreover, ODs were recognized by the HTT-Fuzzy logic, whereas optimal load balancing was ensured by QASOA, thereby improving the reliability of IoT networks. A holistic technique was achieved by incorporating these two advanced methods for securing both data transmission and device behavior monitoring. Gaps in encryption key integrity, dynamic topology handling, lightweight computation, intrusion detection, and data authenticity validation were collectively addressed by the proposed models.

5.2 LIMITATIONS AND FUTURE SCOPE OF THE RESEARCH

CP-ABE offers multiple ways to provide security for the IoT. Because IoT environments demand dynamic and adaptable AC policies, they render fine-grained AC, which is very pertinent. For instance, different users in a smart home might have changing access permissions and abilities, which could alter over time. Without jeopardizing system security, CP-ABE allows access to be granted or revoked in response to dynamic conditions, such as changes in the user's position or location. Data privacy and confidentiality could be enhanced by it. Devices in an IoT context gather sensitive data, which needs to be safeguarded from unwanted access. By enabling factors like the location or identity of the user to be encrypted with the data, ABE contributes to its security. If an attacker has access to the encrypted data, then they can't decode it without the essential attributes because only users with those attributes could decrypt the data. Furthermore, large-scale IoT installations can benefit from ABE's scalable and efficient solution for managing AC policies. ABE permits defining ACs that are updated and maintained centrally using attributes. This can be especially helpful in IoT setups with several devices with altering abilities and access levels.

Another possibility is provided by ABE's dynamic and adaptive character in IoT applications. In various IoT applications, dynamic attribute-centric AC policies made possible by ABE facilitate effective data sharing and teamwork. Because IoT devices typically work in dynamic environments with frequently changing properties, ABE allows users to control and implement these dynamic AC settings, guaranteeing the security of their data. Moreover, ABE offers the foundation for data-based IoT privacy and security [167]. It makes it possible to directly apply privacy and security laws to the information without worrying about the devices or fundamental infrastructure. This technique allows IoT devices to enforce AC regulations as well as provide data protection even in highly distributed and

heterogeneous IoT environments. ABE makes secure data aggregation and collaboration possible. Like the IoT, many devices generate data that needs to be gathered and examined comprehensively. ABE makes attribute-based data encryption possible from different devices, preserving privacy while enabling data decryption and aggregation. Better insights and outcomes result from encouraging collaborative data investigation and DM in IoT applications. Using ABE in an IoT setting is encouraged by all of the above-discussed opportunities.

There are several obstacles to overcome when implementing ABE in IoT. A notable processing cost is initially linked to ABE, which may pose a problem for IoT devices with limited resources. Complex cryptographic processes like attribute-based access policy evaluation and bilinear pairings are needed for ABE, and they might be computationally demanding for IoT devices with constrained memory and processing capacity. Second, integrating ABE in IoT contexts can provide interoperability issues. It can be difficult to establish a standardized ABE system that is well-suited to all IoT devices since diverse devices may employ various data formats as well as communication protocols. Ensuring device compatibility is crucial to the effective application of ABE in IoT. Third, controlling attribute authority and attribute revocation can also be difficult in IoT environments.

Handling the attributes linked to every single IoT device is complex because these devices are very dynamic as well as may leave or join the network regularly. Furthermore, revocation of characteristics in the event of device loss, theft, or unreliability is complicated. Also, attribute revocation procedures are properly thought out to safeguard the security of jeopardized systems.

Another problem that ABE faces when implemented in a larger-scale IoT setting is scalability. The difficulty of creating and enforcing ACs for every IoT device grows due to the increase in the number of IoT devices. ABE often involves complex cryptographic processes, and expanding these activities to serve many devices can cause resource strain on computers, increase response times, and cause delays. One of the challenges in implementing ABE in the IoT is guaranteeing the privacy of user attributes. If attributes are not correctly handled, they may reveal sensitive information about people or be used to identify them. Attributes may include sensitive information. To safeguard users' privacy, great thought should go into how attributes are created, maintained, and shared. Lastly, interoperability

and standardization issues need to be settled to promote the use and compatibility of ABE in IoT. Standardized protocols, frameworks, and interfaces that make it easy to integrate ABE into different IoT devices as well as platforms can be developed to improve interoperability and enable safe and efficient communication amongst heterogeneous IoT setups.

CP-ABE technique in composite order groups using a partially hidden policy based on LSSS. Centered on Lai's, a more efficient plan is suggested for prime order groups. However, for successful decryption, both authorized users must be tested numerous times for finding the right attributes, which, in certain circumstances, may become extremely complex when a person has multiple values for every single attribute. Even though the public attribute names also cause leaks, all of the aforementioned approaches concentrate on partially disguised access policies. However, their effectiveness is severely limited, meaning that the CT size could be super-polynomial. Delicate data. Several more inner-product predicate encryption techniques, as well as vector encryption, are concealed.

In this paper, researchers provide a new CP-ABE technique that combines edge computing and ECC. The scheme delegated the majority of the computation to the edge IoT agent and took into account dynamic time and location attributes for fine-grained AC. The assumption also validates the suggested method's security. The attributes are managed by several authorities to circumvent the issues associated with a single authority and key escrow.

Furthermore, to attain more precise AC, we limit the CT's accessibility based on the users' location and time. Lastly, the experimental research results demonstrate our strategy's effectiveness and suitability for the power IoT.

Future Scope of the Research: An adaptable and practical LWC-ABE technique is used in this article to eliminate the odd attacks that arise in the IoT. Additionally, it lowers hardware resources like power consumption and employs greater security requirements in the IoT network by utilizing the Playfair and ChaCha encryptions. The recommended approach supports concurrent attribute revocation, policy updating, and outsourcing of decryption capabilities. Consequently, to meet more stringent security requirements, the DOs regularly modify the policy access specifications and produce unique CTs. Additionally, the policy modification states that the DOs can adjust the data access attributes flexibly. Additionally, numerous TA environments, which are BN in IoT servers and devices, are brought into the

IoT network, and they possess the adaptability to modify their access restrictions [168]. According to the simulation outcomes, the suggested LWC-ABE outperformed traditional methods concerning decryption time and encryption time for several message sizes as well as numerous users. Comparing the proposed LWC-ABE to the traditional ABE, less time is spent on encryption and decryption. This research is expanded to include hybrid encryption methods to increase security.

Furthermore, we discovered that the system's total performance will significantly increase when IoT devices with hardware accelerators are used, especially for ABE cryptographic procedures. Furthermore, the research focuses on pairing-centric cryptographic hardware accelerators to enhance IoT device performance during encryption and decryption. Extra-featured schemes are always more expensive and can be challenging to apply in an IoT setting. These additional qualities are useful for cryptography, too. Therefore, research should improve and develop new schemes to lower data producers' processing and communication affordability.

The research presents a low-power CP-ABE technique for cloud-assisted IoT environments that uses PRKs as well as constant-size CTs for IoT devices. Moreover, to safely and discreetly outsource the proxy server's decryption task, privacy-preserving outsourced decryption method can be utilized by the customers. In addition to safeguarding user privacy and data confidentiality, our approach can lessen the user computing burden and the communication overhead of cloud-assisted IoT systems. CP-ABE in IoT contexts can be used better in a few areas in the future. One area of focus is the creation of more effective ABE algorithms that increase the viability of IoT devices with limited resources and lower computational overhead. Moreover, to improve the IoT devices' overall security, ABE is coupled with other security measures, including blockchain. The creation of ABE-based AC frameworks for application in various domains of the IoT, including smart homes, smart cities, and industrial IoT, is another area of focus. The work can also be expanded by including ideas like artificial intelligence and deep learning, which will teach the computer to defend the system from security risks and improve overall robustness and accuracy. For safe and efficient IoT installations, the application of ABE is improved as well as optimized by concentrating on these areas. Furthermore, because the encryption and decryption keys are so big, quantum-resistant ABE algorithms are currently not feasible for IoT devices.

Given that stakeholders will have access to quantum computers in the coming decades, it is suggested that academics concentrate on creating more potent quantum-resistant techniques for ABE.



REFERENCES

- [1] Li, X., Wang, Q., Lan, X., Chen, X., Zhang, N., & Chen, D. (2019). Enhancing Cloud-Based IoT Security through Trustworthy Cloud Service: An Integration of Security and Reputation Approach. *IEEE Access*, 7, 9368–9383. <https://doi.org/10.1109/ACCESS.2019.2896932>
- [2] Joshi, G.P., Acharya, S., Kim, C.S., Kim, B.S., & Kim, S.W. (2014). Smart solutions in elderly care facilities include an RFID system and its integration with wireless sensor networks. *International Journal of Distributed Sensor Networks*, 10(7), Article 713946. <https://doi.org/10.1155/2014/713946>
- [3] Riad, K., Hamza, R., & Yan, H. (2019). Sensitive and Energetic IoT Access Control for Managing Cloud Electronic Health Records. *IEEE Access*, 7, 86384–86393. <https://doi.org/10.1109/ACCESS.2019.2927634>
- [4] La Manna, M., Treccozzi, L., Perazzo, P., Saponara, S., & Dini, G. (2021). Performance evaluation of attribute-based encryption in the automotive embedded platform for secure software over-the-air update. *Sensors*, 21(2), 515. <https://doi.org/10.3390/s21020515>
- [5] Guo, L., Yang, X., & Yau, W. C. (2021). TABE-DAC: Efficient traceable attribute-based encryption scheme with dynamic access control based on blockchain. *IEEE Access*, 9, 8479–8490. <https://doi.org/10.1109/ACCESS.2021.3049549>
- [6] Xiang, X., & Zhao, X. (2022). Blockchain-assisted searchable attribute-based encryption for e-health systems. *Journal of Systems Architecture*, 124, 102417. <https://doi.org/10.1016/j.sysarc.2022.102417>
- [7] Rana, M., Mamun, Q., & Islam, R. (2022). Lightweight cryptography in IoT networks: A survey. *Future Generation Computer Systems*, 129, 77–89. <https://doi.org/10.1016/j.future.2021.11.011>
- [8] Zitouni, N., Sedrati, M., & Behaz, A. (2022). Comparing lightweight algorithms to secure constrained objects in the Internet of Things. In M. E. Auer & T. Tsiatsos (Eds.),

Interactive Mobile Communication, Technologies and Learning (pp. 1040–1051). Springer.

- [9] Adeel, A., Ali, M., Khan, A. N., Khalid, T., Rehman, F., Jararweh, Y., & Shuja, J. (2022). A multi-attack resilient, lightweight IoT authentication scheme. *Transactions on Emerging Telecommunications Technologies*, 33(3), e3676. <https://doi.org/10.1002/ett.3676>
- [10] Kumar, V., Kumar, R., Khan, A. A., Kumar, V., Chen, Y. C., & Chang, C. C. (2022). RAFI: Robust authentication framework for IoT-based RFID infrastructure. *Sensors*, 22(9), 3110. <https://doi.org/10.3390/s22093110>
- [11] Jammula, M., Vakamulla, V. M., & Kondoju, S. K. (2022). Performance evaluation of lightweight cryptographic algorithms for heterogeneous IoT environment. *Journal of Interconnection Networks*, Article 2141031. <https://doi.org/10.1142/S0219265921410310>
- [12] Kumar, A., Ottaviani, C., Gill, S. S., & Buyya, R. (2022). Securing the future internet of things with post quantum cryptography. *Security and Privacy*, 5(2), e200. <https://doi.org/10.1002/spy2.200>
- [13] Prakasam, P., Madheswaran, M., Sujith, K. P., & Sayeed, M. S. (2022). Low latency, area and optimal power hybrid lightweight cryptography authentication scheme for internet of things applications. *Wireless Personal Communications*, 126, 351–365. <https://doi.org/10.1007/s11277-022-09748-1>
- [14] Perazzo, P., Righetti, F., La Manna, M., & Vallati, C. (2021). Performance evaluation of attribute-based encryption on constrained IoT devices. *Computer Communications*, 170, 151–163.
- [15] Allassaf, N., Alkazemi, B., & Gutub, A. (2017). Applicable light-weight cryptography to secure medical data in IoT systems. *Journal of Research in Engineering and Applied Sciences (JREAS)*, 2(2), 50–58. <https://doi.org/10.46565/jreas.2017.v02i02.002>

- [16] Alassaf, N., & Gutub, A. (2019). Simulating light-weight-cryptography implementation for IoT healthcare data security applications. *International Journal of E-Health and Medical Communications (IJEHMC)*, 10(4), 1-15.
- [17] Li, H., Yu, K., Liu, B., Feng, C., Qin, Z., & Srivastava, G. (2021). An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things. *IEEE Journal of Biomedical and Health Informatics*, 26(5), 1949-1960.
- [18] Lu, Z., Guo, Y., Li, J., Jia, W., Lv, L., & Shen, J. (2022). Novel searchable attribute-based encryption for the internet of things. *Wireless Communications and Mobile Computing*, 2022.
- [19] Hu, S., Wang, X., He, H., & Zhong, T. (2022). Complex and flexible data access policy in attribute-based encryption. *The Journal of Supercomputing*, 78(1), 1010–1029. <https://doi.org/10.1007/s11227-021-03867-5>
- [20] El Hadj Youssef, W., Abdelli, A., Dridi, F., Brahim, R., & Machhout, M. (2022). Efficient lightweight cryptographic instructions set extension for IoT device security. *Security and Communication Networks*, 2022.
- [21] Chai, N., Mao, C., Ren, M., Zhang, W., Poovendran, P., & Balamurugan, P. (2021). Role of BIC (Big Data, IoT, and Cloud) for Smart Cities. *Arabian Journal of Science and Engineering*. Article in Press.
- [22] Rubí, J.N.S., & Gondim, P.R.D.L. (2021). IoT-based platform for environment data sharing in smart cities. *International Journal of Communication Systems*, 34, 4515. <https://doi.org/10.1002/dac.4515>
- [23] Kaur, M.J., & Maheshwari, P. (2016). Building smart cities applications using IoT and cloud-based architectures. In *Proceedings of the 2016 International Conference on Industrial Informatics and Computer Systems (CIICS)* (pp. 1–5). IEEE, Sharjah, United Arab Emirates, 13–15 March 2016.
- [24] Saleem, S.I., Zeebaree, S., Zeebaree, D.Q., & Abdulazeez, A.M. (2020). Building smart cities applications based on IoT technologies: A review. *Technology Reports of Kansai University*, 62, 1083–1092.

- [25] Dlodlo, N., Gcaba, O., & Smith, A. (2016). Internet of things technologies in smart cities. In *Proceedings of the 2016 IST-Africa Week Conference* (pp. 1–7). IEEE, Durban, South Africa, 11–13 May 2016.
- [26] Hyman, B.T., Alisha, Z., & Gordon, S. (2019). Secure Controls for Smart Cities: Applications in Intelligent Transportation Systems and Smart Buildings. *International Journal of Science and Engineering Applications*, 8, 167–171. <https://doi.org/10.7753/IJSEA0804.1003>
- [27] Curry, E., Dustdar, S., Sheng, Q.Z., & Sheth, A. (2016). Smart cities—Enabling services and applications. *Journal of Internet Services and Applications*, 7(1), 1–3. <https://doi.org/10.1186/s13174-016-0048-6>
- [28] González-Zamar, M.-D., Abad-Segura, E., Vázquez-Cano, E., & López-Meneses, E. (2020). IoT Technology Applications-Based Smart Cities: Research Analysis. *Electronics*, 9(1246). <https://doi.org/10.3390/electronics9081246>
- [29] Saravanan, K., Julie, E.G., & Robinson, Y.H. (2019). Smart cities & IoT: Evolution of applications, architectures & technologies, present scenarios & future dream. In *Internet of Things and Big Data Analytics for Smart Generation* (pp. 135–151). Springer, Cham.
- [30] Shamsir, S., Mahbub, I., Islam, S.K., & Rahman, A. (2017). Applications of sensing technology for smart cities. In *Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)* (pp. 1150–1153), Boston, MA, USA, 6–9 August 2017.
- [31] Lanza, J., Sánchez, L., Gutiérrez, V., Galache, J.A., Santana, J.R., Sotres, P., & Muñoz, L. (2016). Smart city services over a future Internet platform based on IoT and cloud: The smart parking case. *Energies*, 9(719). <https://doi.org/10.3390/en9070719>
- [32] Syed, A., Sierra-Sosa, D., Kumar, A., & Elmaghraby, A. (2021). IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities*, 4, 429–475. <https://doi.org/10.3390/smartcities4020022>

- [33] Almalki, F.A., Alsamhi, S.H., Sahal, R., Hassan, J., Hawbani, A., Rajput, N.S., Saif, A., Morgan, J., & Breslin, J. (2021). Green IoT for Eco-Friendly and Sustainable Smart Cities: Future Directions and Opportunities. *Mobile Networks and Applications*. Advance online publication. <https://doi.org/10.1007/s11036-021-01773-z>
- [34] Xively. (n.d.). Retrieved from <https://www.nettrotter.io/en/xively-platform>
- [35] Adedeji, K.B., & Ponnle, A. (2014). A New Hybrid Data Encryption and Decryption Technique to Enhance Data Security in Communication Networks: Algorithm Development. *International Journal of Scientific & Engineering Research*, 5(10).
- [36] Chandra, S., Mandal, B., Alam, S.S., & Bhattacharyya, S. (2015). Content based double encryption algorithm using symmetric key cryptography. In *International Conference on Recent Trends in Computing*.
- [37] Iyer, S.C., Sedamkar, R.R., & Gupta, S. (2016). A Novel Idea on Multimedia Encryption using Hybrid Crypto Approach. In *7th International Conference on Communication, Computing and Virtualization*.
- [38] Khan, M.A., Mishra, K.K., Santhi, N., & Jayakumari, J. (2015). A New Hybrid Technique for Data Encryption. In *Proceedings of 2015 Global Conference on Communication Technologies*.
- [39] WU, X. (2010). Research of the Database Encryption Technique Based on Hybrid Cryptography. In *International Symposium on Computational Intelligence and Design*.
- [40] Singh, K. (2010). Selective encryption technique in RSA based singular cubic curve with AVK for text based documents: Enhancement of Koyama approach. In *2010 International Conference on Networking and Information Technology (ICNIT)*. DOI:10.1109/ICNIT.2010.5508497
- [41] Mohamed, N. N., Hashim, H., & Yussoff, Y. M. Y. (2014, April). Compression and Encryption Technique on Securing TFTP Packet. In *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Penang, Malaysia.

- [42] Sharbaf, M. S. (2009). Quantum Cryptography: A New Generation of Information Technology Security System. Retrieved from <http://ieeexplore.ieee.org/xpl/freeabsall?arnumber=5070885>
- [43] Kharat, P. H. (2015, May). A secured Transmission of data using 3D chaotic map encryption and data hiding technique. In International Conference on Industrial Instrumentation and Control (ICIC), College of Engineering Pune, India.
- [44] El Hennawy, H. M. S., Omar, A. E. A., & Kholaf, S. M. A. (2014). LEA: Link Encryption Algorithm Proposed Stream Cipher Algorithm. Production and hosting by Elsevier B.V. on behalf of Ain Shams University.
- [45] Katz, J., Sahai, A., & Waters, B. (2013). Predicate encryption supporting disjunctions, polynomial equations, and inner products. *Journal of Cryptology*, 26(2), 191–224.
- [46] Patil, P. A., & Joshi, S. (2017). Hidden CP-ABE to Enhance Patient Data Privacy in Smart Healthcare Systems. *International Journal of Applied Engineering Research*, 12, 3950–3960.
- [47] Phuong, T. V. X., Yang, G., & Susilo, W. (2016). Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions. *IEEE Transactions on Information Forensics and Security*, 11(1), 35–45.
- [48] Yao, X., Liu, H., Ning, H., Yang, L. T., & Xiang, Y. (2015). Anonymous Credential-Based Access Control Scheme for Clouds. *IEEE Cloud Computing*, 2(4), 34–43.
- [49] Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-Policy Attribute-Based Encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP '07)*, Berkeley, CA, USA (pp. 321–334).
- [50] Doshi, D., & Jinwala, D. C. (2014). Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption. *Security and Communication Networks*, 7, 1988–2002.
- [51] Gorasia, N., Srikanth, R. R., Doshi, N., & Rupareliya, J. (2016). Improving Security in Multi Authority Attribute Based Encryption with Fast Decryption. *Procedia Computer Science*, 79, 632–639.

- [52] Miao, Y., Liu, X., Choo, K. K. R., Deng, R. H., Li, J., Li, H., & Ma, J. (2021). Privacy-Preserving Attribute-Based Keyword Search in Shared Multi-owner Setting. *IEEE Transactions on Dependable and Secure Computing*, 18, 1080–1094.
- [53] Helil, N., & Rahman, K. (2017). CP-ABE Access Control Scheme for Sensitive Data Set Constraint with Hidden Access Policy and Constraint Policy. *Security and Communication Networks*, 2017, Article 2713595.
- [54] Sabitha, S., & Rajasree, M. S. (2017). Access control based privacy preserving secure data sharing with hidden access policies in cloud. *Journal of Systems Architecture*, 75, 50–58.
- [55] Liu, L., Lai, J., Deng, R. H., & Li, Y. (2016). Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment. *Security and Communication Networks*, 9, 4897–4913.
- [56] Balu, A., & Kuppusamy, K. (2010). Ciphertext policy Attribute based Encryption with anonymous access policy. *International Journal of Peer to Peer Networks*, 1(1), 1–8.
- [57] Yadav, U. C., & Ali, S. T. (2015, August). Ciphertext policy-hiding attribute-based encryption. In *Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India (pp. 2067–2071).
- [58] Zhong, H., Zhu, W., Xu, Y., & Cui, J. (2018). Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage. *Soft Computing*, 22, 243–251.
- [59] Yang, K., Han, Q., Li, H., Zheng, K., Su, Z., & Shen, X. (2017). An Efficient and Fine-Grained Big Data Access Control Scheme with Privacy-Preserving Policy. *IEEE Internet of Things Journal*, 4(2), 563–571.
- [60] Appleby, A. (2011). MurmurHash. Retrieved from <https://sites.google.com/site/murmurhash/>

- [61] Zhang, Y., Li, J., Chen, X., Li, H. (2016). Anonymous attribute-based proxy re-encryption for access control in cloud computing. *Security and Communication Networks*, 9, 2397–2411.
- [62] Zhang, Y., Zheng, D., Deng, R. H. (2018). Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control. *IEEE Internet of Things Journal*, 5(5), 2130–2145.
- [63] Zhang, Y., Chen, X., Li, J., Wong, D. S., Li, H., You, I. (2017). Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Information Sciences*, 379, 42–61.
- [64] Gupta, M., Benson, J., Patwa, F., & Sandhu, R. (2019, March). Dynamic Groups and Attribute-Based Access Control for Next-Generation Smart Cars. In *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, Richardson, TX, USA (pp. 61–72).
- [65] Chinnasamy, P., Deepalakshmi, P., Dutta, A. K., You, J., & Joshi, G. P. (2022). Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System. *Mathematics*, 10(1), 68. <https://doi.org/10.3390/math10010068>
- [66] Mehmood, M. R., Shahid, A., Jamil, A., Ashraf, R., Mahmood, T., & Mehmood, A. (2019). A Comprehensive Literature Review of Data Encryption Techniques in Cloud Computing and IoT Environment. In *2019 8th International Conference on Information and Communication Technologies (ICICT)* (pp. 54-59), Karachi, Pakistan. <https://doi.org/10.1109/ICICT47744.2019.9001945>
- [67] VENKATESAN, S., RAJA RAJESHWARI, K., RAMAKRISHNAN, M., et al. (2022). A Lightweight Searchable Encryption and Delegation Mechanism with Forward Privacy for Improving the Security of Industrial Internet of Things-Cloud Systems, 23 February 2022, PREPRINT (Version 1) available at Research Square. <https://doi.org/10.21203/rs.3.rs-1384533/v1>

- [68] Suthar, K., Kumar, P., Gupta, H., & Patel, H. (2012). Analytical comparison of symmetric encryption and encoding techniques for cloud environment. *International Journal of Computer Applications*, 60(19), 16–19.
- [69] Soofi, S. S., Khan, M. I., & Amin, F. (2014). Encryption techniques for cloud data confidentiality. *International Journal of Grid and Distributed Computing*, 17(4), 11–20.
- [70] Song, C., Park, Y., Gao, J., Nanduri, S. K., & Zegers, W. (2015). Favored encryption techniques for cloud storage. In *IEEE First International Conference on Big Data Computing Service and Applications* (pp. 267-274).
- [71] Sikka, P. (2022). Hashed Access Policy-Based Secure Data Transmission Using SDHAK-ECC. *International Journal of Cooperative Information Systems*, 31(03n04), 2350001. <https://doi.org/10.1142/S0218843023500013>
- [72] Ranjith, & Mahantesh, K. (2019). Privacy and Security issues in Smart Health Care. In *2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)* (pp. 378-383), Mysuru, India. <https://doi.org/10.1109/ICEECCOT46775.2019.9114681>
- [73] RAJESHWARI, K. R., & RAMAKRISHNAN, M. (2020). A Lightweight Searchable Encryption and Delegation Mechanism with Forward Privacy for Improving the Security of Industrial Internet of Things-Cloud Systems. <https://doi.org/10.21203/rs.3.rs-1384533/v1>
- [74] Prabavathi, D., & Prabakaran, M. (2020). A SURVEY OF DATA SECURITY MECHANISMS IN CLOUD COMPUTING.
- [75] Narayanan, U., Paul, V., & Joseph, S. (2022). A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment. *Journal of King Saud University-Computer and Information Sciences*, 34(6), 3121-3135.
- [76] Namasudra, S., & Roy, P. (2016). Secure and efficient data access control in cloud computing environment: a survey. *Multi-agent and Grid Systems - An International Journal*, 12(2), 69-90.

- [77] Mehmood, M. S., Shahid, M. R., Jamil, A., Ashraf, R., Mahmood, T., & Mehmood, A. (2019). A comprehensive literature review of data encryption techniques in cloud computing and IoT environment. In 8th International Conference on Information and Communication Technologies (ICICT) (pp. 54–59).
- [78] Karumanchi, M. D., Sheeba, J. I., & Devaneyan, S. P. (2022). Integrated Internet of Things with cloud developed for data integrity problems on supply chain management. *Measurement: Sensors*, 24, 100445.
- [79] Ijaw, D. Unal, R., Hamila, A., Gastli, A., & Ellabban, O. (2021). Lightweight KPABE Architecture Enabled in Mesh Networked Resource-Constrained IoT Devices. *IEEE Access*, 9, 5640-5650. <https://doi.org/10.1109/ACCESS.2020.3048192>
- [80] Chang, V., Kuo, Y. H., & Ramachandran, M. (2016). Cloud computing adoption framework: A security framework for business clouds. *Future Generation Computer Systems*, 57, 24–41.
- [81] Brandao, P. R. (2018). The importance of authentication and encryption in cloud computing framework security. *International Journal of Data Science and Technology*, 4(1), 1–5.
- [82] Balusamy, B., Krishna, P. V., Arasi, G. T., & Chang, V. (2017). A Secured Access Control Technique for Cloud Computing Environment Using Attribute-Based Hierarchical Structure and Token Granting System. *International Journal of Network Security*, 19(4), 559-572. [https://doi.org/10.6633/IJNS.201707.19\(4\).09](https://doi.org/10.6633/IJNS.201707.19(4).09)
- [83] C. Thirumalai, S. Mohan, and G. Srivastava, "An Efficient Public Key Secure Scheme for Cloud and IoT Security," **Computer Communications**, vol. 150, pp. 634-643, 2020.
- [84] Y. Chandu, K. R. Kumar, N. V. Prabhukhanolkar, A. N. Anish, and S. Rawal, "Design and Implementation of Hybrid Encryption for Security of IoT Data," in **2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon)**, 2017, pp. 1228-1231.

- [85] S. Namasudra, "An Improved Attribute-Based Encryption Technique Towards the Data Security in Cloud Computing," **Concurrency and Computation: Practice and Experience**, vol. 31, no. 3, p. e4364, 2019.
- [86] M. Vashishtha, P. Chouksey, D. S. Rajput, S. R. Reddy, M. P. K. Reddy, G. T. Reddy, and H. Patel, "Security and Detection Mechanism in IoT-Based Cloud Computing Using Hybrid Approach," **International Journal of Internet Technology and Secured Transactions**, vol. 11, no. 5-6, pp. 436-451, 2021.
- [87] T. Wang, Q. Yang, X. Shen, T. R. Gadekallu, W. Wang, and K. Dev, "A Privacy-Enhanced Retrieval Technology for the Cloud-Assisted Internet of Things," **IEEE Transactions on Industrial Informatics**, vol. 18, no. 7, pp. 4981-4989, 2021.
- [88] C. Wang, D. Wang, Y. Duan, and X. Tao, "Secure and Lightweight User Authentication Scheme for Cloud-Assisted Internet of Things," **IEEE Transactions on Information Forensics and Security**, 2023.
- [89] S. Halder and T. Newe, "Enabling Secure Time-Series Data Sharing via Homomorphic Encryption in Cloud-Assisted IIoT," **Future Generation Computer Systems**, vol. 133, pp. 351-363, 2022.
- [90] P. Chinnasamy, P. Deepalakshmi, A. K. Dutta, J. You, and G. P. Joshi, "Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System," **Mathematics**, vol. 10, no. 1, p. 68, 2021.
- [91] A. Saidi, O. Nouali, and A. Amira, "SHARE-ABE: An Efficient and Secure Data Sharing Framework Based on Ciphertext-Policy Attribute-Based Encryption and Fog Computing," **Cluster Computing**, vol. 25, no. 1, pp. 167-185, 2022.
- [92] V. R. Catherine and A. S. Nargunam, "Multi-Authority Ciphertext-Policy Attribute-Based Encryption for Security Enhancement in Cloud Storage Unit," **Sustainable Energy Technologies and Assessments**, vol. 53, p. 102556, 2022.
- [93] Hasegawa, K., Kanayama, N., Nishide, T., & Okamoto, E. (2016). Software library for ciphertext/key-policy functional encryption with simple usability. *Journal of Information Processing*, 24(5), 764-771.

- [94] Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security (pp. 89-98).
- [95] Shamir, A. (1985). Identity-based cryptosystems and signature schemes. In Advances in Cryptology: Proceedings of CRYPTO '84 (pp. 47-53). Springer Berlin Heidelberg.
- [96] Boneh, D., & Franklin, M. (2001, August). Identity-based encryption from the Weil pairing. In Annual International Cryptology Conference (pp. 213-229). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [97] Cocks, C. (2001). An identity-based encryption scheme based on quadratic residues. In Cryptography and Coding: 8th IMA International Conference, Cirencester, UK, December 17–19, 2001 Proceedings (pp. 360-363). Springer Berlin Heidelberg.
- [98] Rauscher, J., & Bauer, B. (2018, September). Safety and security architecture analyses framework for the internet of things of medical devices. In 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom) (pp. 1-3). IEEE.
- [99] Li, C., & Yang, C. (2018, July). (WIP) authenticated key management protocols for internet of things. In 2018 IEEE International Congress on Internet of Things (ICIOT) (pp. 126-129). IEEE.
- [100] Rajashree, S., Shah, P. G., & Murali, S. (2018, July). Security model for internet of things end devices. In 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 219-221). IEEE.
- [101] Deshmukh, S., & Sonavane, S. S. (2017, March). Security protocols for Internet of Things: A survey. In 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2) (pp. 71-74). IEEE.
- [102] Lu, X., Pan, Z., & Xian, H. (2020). An integrity verification scheme of cloud storage for internet-of-things mobile terminal devices. Computers & Security, 92, 101686.

- [103] Shukla, S., Hassan, M. F., Tran, D. C., Akbar, R., Paputungan, I. V., & Khan, M. K. (2021). Improving latency in Internet-of-Things and cloud computing for real- time data transmission: a systematic literature review (SLR). *Cluster Computing*, 26(5), 1–24. <https://doi.org/10.1007/s10586-021-03279-3>
- [104] Iqbal, U., Tandon, A., Gupta, S., Yadav, A. R., Neware, R., & Gelana, F. W. (2022). A Novel Secure Authentication Protocol for IoT and Cloud Servers. *Wireless Communications and Mobile Computing*, 2022, 1–17. <https://doi.org/10.1155/2022/7707543>
- [105] Albshaier, L., Budokhi, A., & Aljughaiman, A. (2024). A Review of Security Issues When Integrating IoT With Cloud Computing and Blockchain. *IEEE Access*, 12, 109560–109595. <https://doi.org/10.1109/ACCESS.2024.3435845>
- [106] Ullah, A., Azeem, M., Ashraf, H., Alaboudi, A. A., Humayun, M., & Jhanjhi, N. Z. (2021). Secure Healthcare Data Aggregation and Transmission in IoT - A Survey. *IEEE Access*, 9, 16849–16865. <https://doi.org/10.1109/ACCESS.2021.3052850>
- [107] Stergiou, C. L., Koidou, M. P., & Psannis, K. E. (2023). IoT-Based Big Data Secure Transmission and Management over Cloud System: A Healthcare Digital Twin Scenario. *Applied Sciences (Switzerland)*, 13(16), 1–28. <https://doi.org/10.3390/app13169165>
- [108] Irshad, R. R., Hussain, S., Hussain, I., Nasir, J. A., Zeb, A., Alalayah, K. M., Alattab, A. A., Yousif, A., & Alwayle, I. M. (2023). IoT-Enabled Secure and Scalable Cloud Architecture for Multi-User Systems: A Hybrid Post-Quantum Cryptographic and Blockchain-Based Approach Toward a Trustworthy Cloud Computing. *IEEE Access*, 11, 105479–105498. <https://doi.org/10.1109/ACCESS.2023.3318755>
- [109] Panda, P. K., & Chattopadhyay, S. (2021). An enhanced mutual authentication and security protocol for IoT and cloud server. *Information Security Journal*, 31(2), 144–156. <https://doi.org/10.1080/19393555.2020.1871534>
- [110] Bao, Y., Qiu, W., & Cheng, X. (2022). Secure and Lightweight Fine-Grained Searchable Data Sharing for IoT-Oriented and Cloud-Assisted Smart Healthcare

System. IEEE Internet of Things Journal, 9(4), 2513–2526.
<https://doi.org/10.1109/JIOT.2021.3063846>

- [111] Vengala, D. V. K., Kavitha, D., & Kumar, A. P. S. (2023). Three factor authentication system with modified ECC based secured data transfer: untrusted cloud environment. *Complex and Intelligent Systems*, 9(3), 2915–2928. <https://doi.org/10.1007/s40747-021-00305-0>
- [112] Zargar, S., Shahidinejad, A., & Ghobaei-Arani, M. (2021). A lightweight authentication protocol for IoT-based cloud environment. *International Journal of Communication Systems*, 34(11), 1–17. <https://doi.org/10.1002/dac.4849>
- [113] Hussien, Z. A., Abdulmalik, H. A., Hussain, M. A., Nyangaresi, V. O., Ma, J., Abduljabbar, Z. A., & Abduljaleel, I. Q. (2023). Lightweight Integrity Preserving Scheme for Secure Data Exchange in Cloud-Based IoT Systems. *Applied Sciences (Switzerland)*, 13(2), 1–20. <https://doi.org/10.3390/app13020691>
- [114] Refaee, E., Parveen, S., Begum, K. M. J., Parveen, F., Raja, M. C., Gupta, S. K., & Krishnan, S. (2022). Secure and Scalable Healthcare Data Transmission in IoT Based on Optimized Routing Protocols for Mobile Computing Applications. *Wireless Communications and Mobile Computing*, 2022, 1–12. <https://doi.org/10.1155/2022/5665408>
- [115] Karati, A., Amin, R., Mohit, P., Sureshkumar, V., & Biswas, G. P. (2021). Design of a secure file storage and access protocol for cloud-enabled Internet of Things environment. *Computers and Electrical Engineering*, 94, 1–15. <https://doi.org/10.1016/j.compeleceng.2021.107298>
- [116] Kaur, M., Alzubi, A. A., Walia, T. S., Yadav, V., Kumar, N., Singh, D., & Lee, H. N. (2023). EGCrypto: A Low-Complexity Elliptic Galois Cryptography Model for Secure Data Transmission in IoT. *IEEE Access*, 11, 90739–90748. <https://doi.org/10.1109/ACCESS.2023.3305271>
- [117] El-Latif, A. A. A., Iliyasu, A. M., & Abd-El-atty, B. (2021). An efficient visually meaningful quantum walks-based encryption scheme for secure data transmission on

IoT and smart applications. *Mathematics*, 9(23), 1–20.
<https://doi.org/10.3390/math9233131>

- [118] Chinnasamy, P., Deepalakshmi, P., Dutta, A. K., You, J., & Joshi, G. P. (2022). Ciphertext-policy attribute-based encryption for cloud storage: Toward data privacy and authentication in ai-enabled iot system. *Mathematics*, 10(1), 1–24.
<https://doi.org/10.3390/math10010068>
- [119] Premkumar, R., & Sathya Priya, S. (2022). Service Constraint NCBQ trust orient secure transmission with IoT devices for improved data security in cloud using blockchain. *Measurement: Sensors*, 24, 1–9.
<https://doi.org/10.1016/j.measen.2022.100486>
- [120] Pampapathi, B. M., Nageswara Guptha, M., & Hema, M. S. (2021). Data distribution and secure data transmission using IANFIS and MECC in IoT. *Journal of Ambient Intelligence and Humanized Computing*, 13(3), 1471–1484.
<https://doi.org/10.1007/s12652-020-02792-4>
- [121] Duraisamy, A., & Subramaniam, M. (2021). Attack Detection on IoT Based Smart Cities using IDS Based MANFIS Classifier and Secure Data Transmission Using IRSA Encryption. *Wireless Personal Communications*, 119(2), 1913–1934.
<https://doi.org/10.1007/s11277-021-08362-x>
- [122] Sajid, F., Hassan, M. A., Khan, A. A., Rizwan, M., Kryvinska, N., Vincent, K., & Khan, I. U. (2022). Secure and Efficient Data Storage Operations by Using Intelligent Classification Technique and RSA Algorithm in IoT-Based Cloud Computing. *Scientific Programming*, 2022, 1–10. <https://doi.org/10.1155/2022/2195646>
- [123] Jiang, L., & Mu, C. (2025). Secure identity access and data transmission scheme of cloud-assisted intelligent gymnasium. *Alexandria Engineering Journal*, 115, 469–478.
<https://doi.org/10.1016/j.aej.2024.12.001>
- [124] Alwazzeh, M., Karaman, S., & Shamma, M. N. (2020). Man in the middle attacks against SSL/TLS: Mitigation and defeat. *Journal of Cyber Security and Mobility*, 449–468. <http://dx.doi.org/10.13052/jcsm2245-1439.933>

- [125]Koo, D., Shin, Y., Yun, J., & Hur, J. (2018). Improving security and reliability in Merkle tree-based online data authentication with leakage resilience. *Applied Sciences*, 8(12), 2532. <http://dx.doi.org/10.3390/app8122532>
- [126]Poch, M, (1993). Water Treatment Plant. UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/106/water+treatment+plant>
- [127]Vergara, A, (2012). Gas Sensor Array Drift Dataset. UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/224/gas+sensor+array+drift+dataset>
- [128]Oladipupo, E. T., Abikoye, O. C., & Awotunde, J. B. (2024). A lightweight image cryptosystem for cloud-assisted internet of things. *Applied Sciences*, 14(7), 1-27. <https://doi.org/10.3390/app14072808>
- [129]Jain, K., Titus, B., Krishnan, P., Sudevan, S., Prabu, P., & Alluhaidan, A. S. (2024). A lightweight multi-chaos-based image encryption Scheme for IoT Networks. *IEEE access*, 12, 1-31. <https://10.1109/ACCESS.2024.3377665>
- [130]Jamil, M. N., Schelén, O., Monrat, A. A., & Andersson, K. (2024). Enabling Industrial Internet of Things by Leveraging Distributed Edge-to-Cloud Computing: Challenges and Opportunities. *IEEE Access*, 12, 1-15. <https://10.1109/ACCESS.2024.3454812>
- [131]Kathole, A. B., Vhatkar, K. N., Goyal, A., Kaushik, S., Mirge, A. S., Jain, P., ... & Islam, M. T. (2024). Secure Federated Cloud Storage Protection Strategy Using Hybrid Heuristic Attribute-Based Encryption with Permissioned Blockchain. *IEEE Access*, 12, 1-16. <https://10.1109/ACCESS.2024.3447829>
- [132]Shingari, N., & Mago, B. (2024). A framework for application-centric Internet of Things authentication. *Results in Engineering*, 22, 1-10. <https://doi.org/10.1016/j.rineng.2024.102109>
- [133]Sasikumar, K., & Nagarajan, S. (2024). Comprehensive review and analysis of cryptography techniques in cloud computing. *IEEE Access*, 12, 1-27. <https://10.1109/ACCESS.2024.3385449>

- [134]Kokila, M., & Reddy, S. (2024). Authentication, access control and scalability models in internet of things security-A review. *Cyber Security and Applications*, 3, 1-18. <https://doi.org/10.1016/j.csa.2024.100057>
- [135]Bojič Burgos, J., & Pustišek, M. (2024). Decentralized iot data authentication with signature aggregation. *Sensors*, 24(3), 1-25. <https://doi.org/10.3390/s24031037>
- [136]Iqbal, U., Tandon, A., Gupta, S., Yadav, A. R., Neware, R., & Gelana, F. W. (2022). A novel secure authentication protocol for IoT and cloud servers. *Wireless Communications and Mobile Computing*, 2022(1), 1-17. <https://doi.org/10.1155/2022/7707543>
- [137]Saravanan, V., Madijagan, M., Rafee, S. M., Sanju, P., Rehman, T. B., & Pattanaik, B. (2023). IoT-based blockchain intrusion detection using optimized recurrent neural network. *Multimedia Tools and Applications*, 83(11), 1-22. <https://doi.org/10.1007/s11042-023-16662-6>
- [138]Corthis, P. B., Ramesh, G. P., García-Torres, M., & Ruíz, R. (2024). Effective Identification and Authentication of Healthcare IoT Using Fog Computing with Hybrid Cryptographic Algorithm. *Symmetry*, 16(6), 1-23. <https://doi.org/10.3390/sym16060726>
- [139]Varri, U. S., Mallick, D., Das, A. K., Hossain, M. S., Park, Y., & Rodrigues, J. J. (2024). TL-ABKS: Traceable and lightweight attribute-based keyword search in edge–cloud assisted IoT environment. *Alexandria Engineering Journal*, 107, 1-13. <https://doi.org/10.1016/j.aej.2024.09.030>
- [140]Rana, P., Batra, I., Malik, A., Ra, I. H., Lee, O. S., & Hosen, A. S. (2024). Efficacious Novel Intrusion Detection System for Cloud Computing Environment. *IEEE Access*, 12, 1-17. <https://10.1109/ACCESS.2024.3424528>
- [141]Chahal, R. K., Kumar, N., & Batra, S. (2020). Trust management in social Internet of Things: A taxonomy, open issues, and challenges. *Computer Communications*, 150, 13-46.

- [142]Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-policy attribute-based encryption. In 2007 IEEE Symposium on Security and Privacy (SP'07) (pp. 321-334). IEEE.
- [143]Odelu, V., Das, A. K., Khan, M. K., Choo, K. K. R., & Jo, M. (2017). Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts. *IEEE Access*, 5, 3273-3283.
- [144]Li, X., Wang, Q., Lan, X., Chen, X., Zhang, N., & Chen, D. (2019). Enhancing cloud-based IoT security through trustworthy cloud service: An integration of security and reputation approach. *IEEE Access*, 7, 9368-9383.
- [145]Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006, October). Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security* (pp. 89-98).
- [146]Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-policy attribute-based encryption. In 2007 IEEE symposium on security and privacy (SP'07) (pp. 321-334). IEEE.
- [147]Wan, Z., & Deng, R. H. (2011). HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Transactions on Information Forensics and Security*, 7(2), 743-754.
- [148]Wang, G., Liu, Q., & Wu, J. (2010, October). Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In *Proceedings of the 17th ACM conference on Computer and communications security* (pp. 735-737).
- [149]Beimel, A. (1996). Secure schemes for secret sharing and key distribution.
- [150]Mohamed, D., & Ismael, O. (2023). Enhancement of an IoT hybrid intrusion detection system based on fog-to-cloud computing. *Journal of Cloud Computing*, 12(1), 1-13. <https://doi.org/10.1186/s13677-023-00420-y>
- [151]Bakro, M., Kumar, R. R., Alabrah, A., Ashraf, Z., Ahmed, M. N., Shameem, M., & Abdelsalam, A. (2023). An improved design for a cloud intrusion detection system

using hybrid features selection approach with ML classifier. IEEE Access, 11, 1-20.
<https://doi.org/10.1109/ACCESS.2023.3289405>

- [152] Fatani, A., Dahou, A., Abd Elaziz, M., Al-Qaness, M. A., Lu, S., Alfadhli, S. A., & Alresheedi, S. S. (2023). Enhancing intrusion detection systems for IoT and cloud environments using a growth optimizer algorithm and conventional neural networks. *Sensors*, 23(9), 1-14. <https://doi.org/10.3390/s23094430>
- [153] Alzughaibi, S., & El Khediri, S. (2023). A cloud intrusion detection systems based on dnn using backpropagation and pso on the cse-cic-ids2018 dataset. *Applied Sciences*, 13(4), 1-21. <https://doi.org/10.3390/app13042276>
- [154] Aljuaid, W. A. H., & Alshamrani, S. S. (2024). A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences*, 14(13), 1-19. <https://doi.org/10.3390/app14135381>
- [155] Wu, A., Zheng, D., Zhang, Y., & Yang, M. (2018). Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing. *Sensors*, 18(7), 2158.
- [156] Pradhan, B., Singh, B., Bhorla, A., & Singh, A. K. (2021). A comparative study on cipher text policy attribute based encryption schemes. *International Journal of Engineering Research & Technology*, 10(5), 786-790.
- [157] Li, X., Gu, D., Ren, Y., Ding, N., & Yuan, K. (2012). Efficient ciphertext-policy attribute based encryption with hidden policy. In *Internet and Distributed Computing Systems: 5th International Conference, IDCS 2012, Wuyishan, Fujian, China, November 21-23, 2012. Proceedings 5* (pp. 146-159). Springer Berlin Heidelberg.
- [158] Al-Dahhan, R. R., Shi, Q., Lee, G. M., & Kifayat, K. (2019). Survey on revocation in ciphertext-policy attribute-based encryption. *Sensors*, 19(7), 1695.
- [159] Pan, F., Zhang, B., Zhao, X., Shuai, L., Chen, P., & Duan, X. (2023). A Lightweight, Secure Authentication Model for the Smart Agricultural Internet of Things. *Agronomy*, 13(9), 2257..

- [160] Tan, S. Y., Yeow, K. W., & Hwang, S. O. (2019). Enhancement of a lightweight attribute-based encryption scheme for the Internet of Things. *IEEE Internet of Things Journal*, 6(4), 6384-6395.
- [161] Moffat, S., Hammoudeh, M., & Hegarty, R. (2017, July). A survey on ciphertext-policy attribute-based encryption (CP-ABE) approaches to data security on mobile devices and its application to IoT. In *Proceedings of the international conference on future networks and distributed systems*.
- [162] Wan, Z., & Deng, R. H. (2011). HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE transactions on information forensics and security*, 7(2), 743-754.
- [163] Yang, W., Wang, R., Guan, Z., Wu, L., Du, X., & Guizani, M. (2020, June). A lightweight attribute-based encryption scheme with constant size ciphertext for internet of things. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.
- [164] Wu, A., Zheng, D., Zhang, Y., & Yang, M. (2018). Hidden policy attribute-based data sharing with direct revocation and keyword search in cloud computing. *Sensors*, 18(7), 2158.
- [165] Hohenberger, S., Lu, G., Waters, B., & Wu, D. J. (2023, April). Registered attribute-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 511-542). Cham: Springer Nature Switzerland.
- [166] Phuong, T. V. X., Yang, G., & Susilo, W. (2014, November). Poster: Efficient ciphertext policy attribute-based encryption under decisional linear assumption. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1490-1492).
- [167] Yao, X., Chen, Z., & Tian, Y. (2015). A lightweight attribute-based encryption scheme for the Internet of Things. *Future Generation Computer Systems*, 49, 104-112.

- [168]Li, X., Wang, Q., Lan, X., Chen, X., Zhang, N., & Chen, D. (2019). Enhancing cloud-based IoT security through trustworthy cloud service: An integration of security and reputation approach. *IEEE access*, 7, 9368-9383.

