

Security, Trust and Privacy for Cloud, Fog and Internet of Things

Lead Guest Editor: Chien-Ming Chen

Guest Editors: Shehzad Ashraf Chaudhry, Kuo-Hui Yeh, and Muhammad Naveed Aman





Security, Trust and Privacy for Cloud, Fog and Internet of Things

Security and Communication Networks

**Security, Trust and Privacy for Cloud,
Fog and Internet of Things**

Lead Guest Editor: Chien-Ming Chen

Guest Editors: Shehzad Ashraf Chaudhry, Kuo-Hui
Yeh, and Muhammad Naveed Aman



Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in "Security and Communication Networks." All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Chief Editor

Roberto Di Pietro, Qatar

Editorial Board

Ahmed A. Abd El-Latif, Egypt
Mamoun Alazab, Australia
Cristina Alcaraz, Spain
Saud Althunibat, Jordan
Ruhul Amin, India
Maria Azees, India
Benjamin Aziz, United Kingdom
Shahram Babaie, Iran
Taimur Bakhshi, United Kingdom
Spiridon Bakiras, Qatar
Pablo Garcia Bringas, Spain
William Buchanan, United Kingdom
Michele Bugliesi, Italy
Jin Wook Byun, Republic of Korea
Pino Caballero-Gil, Spain
Bruno Carpentieri, Italy
Luigi Catuogno, Italy
Shehzad Ashraf Chaudhry, Turkey
Ricardo Chaves, Portugal
Rongmao Chen, China
Chien-Ming Chen, China
Chin-Ling Chen, Taiwan
Tom Chen, United Kingdom
Stelvio Cimato, Italy
Vincenzo Conti, Italy
Luigi Coppolino, Italy
Juhriyansyah Dalle, Indonesia
Salvatore D'Antonio, Italy
Alfredo De Santis, Italy
Angel M. Del Rey, Spain
Roberto Di Pietro, France
Jesús Díaz-Verdejo, Spain
Wenxiu Ding, China
Nicola Dragoni, Denmark
Wei Feng, China
Carmen Fernandez-Gago, Spain
Mohamed Amine Ferrag, Algeria
AnMin Fu, China
Clemente Galdi, Italy
Dimitrios Geneiatakis, Italy
Bela Genge, Romania
Anwar Ghani, Pakistan
Debasis Giri, India
Muhammad A. Gondal, Oman

Prosanta Gope, United Kingdom
Francesco Gringoli, Italy
Biao Han, China
Jinguang Han, United Kingdom
Weili Han, China
Khizar Hayat, Oman
Jiankun Hu, Australia
Iqtadar Hussain, Qatar
Azeem Irshad, Pakistan
M.A. Jabbar, India
Mian Ahmad Jan, Pakistan
Rutvij Jhaveri, India
Tao Jiang, China
Xuyang Jing, China
Minho Jo, Republic of Korea
Bruce M. Kapron, Canada
Arijit Karati, Taiwan
Marimuthu Karuppiah, India
ASM Kayes, Australia
Habib Ullah Khan, Qatar
Fazlullah Khan, Pakistan
Kiseon Kim, Republic of Korea
Sanjeev Kumar, USA
Maryline Laurent, France
Huaizhi Li, USA
Wenjuan Li, Hong Kong
Kaitai Liang, United Kingdom
Xueqin Liang, Finland
Zhe Liu, Canada
Guangchi Liu, USA
Flavio Lombardi, Italy
Pascal Lorenz, France
Yang Lu, China
Leandros Maglaras, United Kingdom
Emanuele Maiorana, Italy
Vincente Martin, Spain
Barbara Masucci, Italy
David Megias, Spain
Weizhi Meng, Denmark
Laura Mongioi, Italy
Raul Monroy, Mexico
Rebecca Montanari, Italy
Leonardo Mostarda, Italy
Mohamed Nassar, Lebanon






Shah Nazir, Pakistan
Qiang Ni, United Kingdom
Mahmood Niazi, Saudi Arabia
Petros Nicopolitidis, Greece
Vijayakumar Pandi, India
A. Peinado, Spain
Gerardo Pelosi, Italy
Gregorio Martinez Perez, Spain
Pedro Peris-Lopez, Spain
Carla Ràfols, Germany
Francesco Regazzoni, Switzerland
Abdaloussein Rezai, Iran
Helena Rifà-Pous, Spain
Arun Kumar Sangaiah, India
Neetesh Saxena, United Kingdom
Savio Sciancalepore, The Netherlands
Young-Ho Seo, Republic of Korea
De Rosal Ignatius Moses Setiadi, Indonesia
Wenbo Shi, China
Ghanshyam Singh, South Africa
Daniel Slamanig, Austria
Salvatore Sorce, Italy
Abdulhamit Subasi, Saudi Arabia
Zhiyuan Tan, United Kingdom
Farhan Ullah, China
Fulvio Valenza, Italy
Sitalakshmi Venkatraman, Australia
Jinwei Wang, China
Qichun Wang, China
Guojun Wang, China
Hu Xiong, China
Xuehu Yan, China
Zheng Yan, China
Anjia Yang, China
Qing Yang, USA
Yu Yao, China
Yinghui Ye, China
Kuo-Hui Yeh, Taiwan
Yong Yu, China
Xiaohui Yuan, USA
Sherali Zeadally, USA
Tao Zhang, China
Leo Y. Zhang, Australia
Zhili Zhou, China
Youwen Zhu, China

Contents



Security, Trust and Privacy for Cloud, Fog and Internet of Things

Chien-Ming Chen , Shehzad Ashraf Chaudhry , Kuo-Hui Yeh , and Muhammad Naveed Aman 
Editorial (2 pages), Article ID 9841709, Volume 2022 (2022)






DAWM: Cost-Aware Asset Claim Analysis Approach on Big Data Analytic Computation Model for Cloud Data Centre

M. S. Mekala , Rizwan Patan , SK Hafizul Islam , Debabrata Samanta , Ghulam Ali Mallah ,
and Shehzad Ashraf Chaudhry 
Research Article (16 pages), Article ID 6688162, Volume 2021 (2021)


Fully Constant-Size CP-ABE with Privacy-Preserving Outsourced Decryption for Lightweight Devices in Cloud-Assisted IoT

Zhishuo Zhang , Wei Zhang, and Zhiguang Qin 
Research Article (16 pages), Article ID 6676862, Volume 2021 (2021)

A Lightweight Intelligent Intrusion Detection Model for Wireless Sensor Networks

Jeng-Shyang Pan , Fang Fan , Shu-Chuan Chu , Hui-Qi Zhao , and Gao-Yuan Liu 
Research Article (15 pages), Article ID 5540895, Volume 2021 (2021)




A Bibliometric Analysis of Edge Computing for Internet of Things

Yiou Wang, Fuquan Zhang , Junfeng Wang, Laiyang Liu, and Bo Wang
Review Article (10 pages), Article ID 5563868, Volume 2021 (2021)





Task Priority-Based Cached-Data Prefetching and Eviction Mechanisms for Performance Optimization of Edge Computing Clusters

Ihsan Ullah , Muhammad Sajjad Khan , Marc St-Hilaire , Mohammad Faisal , Junsu Kim , and
Su Min Kim 
Research Article (10 pages), Article ID 5541974, Volume 2021 (2021)




S-DPS: An SDN-Based DDoS Protection System for Smart Grids

Hassan Mahmood, Danish Mahmood , Qaisar Shaheen , Rizwan Akhtar, and Wang Changda 
Research Article (19 pages), Article ID 6629098, Volume 2021 (2021)

Multiauthority Attribute-Based Encryption with Traceable and Dynamic Policy Updating

Jie Ling , Junwei Chen , Jiahui Chen , and Wensheng Gan 
Research Article (13 pages), Article ID 6661450, Volume 2021 (2021)

Assessing Security of Software Components for Internet of Things: A Systematic Review and Future Directions







Zitian Liao , Shah Nazir , Habib Ullah Khan , and Muhammad Shafiq
Review Article (22 pages), Article ID 6677867, Volume 2021 (2021)

Private Predicate Encryption for Inner Product from Key-Homomorphic Pseudorandom Function

Yi-Fan Tseng, Zi-Yuan Liu , Jen-Chieh Hsu, and Raylin Tso
Research Article (12 pages), Article ID 6678194, Volume 2021 (2021)



Improved Authenticated Key Agreement Scheme for Fog-Driven IoT Healthcare System

Tsu-Yang Wu , Tao Wang , Yu-Qi Lee , Weimin Zheng , Saru Kumari , and Sachin Kumar 

Research Article (16 pages), Article ID 6658041, Volume 2021 (2021)

Editorial

Security, Trust and Privacy for Cloud, Fog and Internet of Things

Chien-Ming Chen ¹, **Shehzad Ashraf Chaudhry** ², **Kuo-Hui Yeh** ³,
and Muhammad Naveed Aman ⁴

¹Shandong University of Science and Technology, Qingdao, China

²Istanbul Gelisim University, Istanbul, Turkey

³National Dong Hwa University, Hualien, Taiwan

⁴University of Nebraska-Lincoln, Lincoln, NE, USA

Correspondence should be addressed to Chien-Ming Chen; chienmingchen@ieee.org

Received 5 January 2022; Accepted 5 January 2022; Published 28 January 2022

Copyright © 2022 Chien-Ming Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet of Things (IoT) is a promising networking scenario in the cyber world, bridging physical devices and virtual objects. By considering the limited capacity of smart things, cloud computing is generally applied to store and process the massive data collected by the IoT. Furthermore, fog computing is described as an extension and a complement to cloud computing. It utilizes fog nodes to perform storage, computation, and communication locally. The merging of cloud/fog computing and IoT can be seen as the best of two worlds by concurrently offering ubiquitous sensing services and powerful processing capabilities.

Despite the advantages of cloud/fog-assisted IoT, it is unwise to neglect the significance of security and privacy in this highly heterogeneous and interconnected system. Various solutions have recently been put forward independently for cloud, fog, or IoT environments to deal with security threats to IoT devices and sensitive data. However, a few crucial features, such as heterogeneity and scalability, have not been appropriately considered in these solutions.

This Special Issue aims to compile recent research efforts dedicated to studying the security and privacy of rapidly increasing cloud/fog-assisted IoT applications. A summary of all the accepted papers is provided as follows.

The paper by Mekala et al. designed a data analytic weight measurement (DAWM) model and multiobjective heuristic user service demand (MHUSD) approach for profit maximization and adequate service reliability. The DAWM model concentrates on instances or machine size with elastic service of generic lambda function to scale up and scale down the instance size as per demand request by considering

instance computation status and its service execution rate and energy consumption. The MHUSD approach measures the CPS profit rate and USD rate before sharing the resources to the instances. The fundamental logic is if the instance DAWM rate is not above moderate or moderate, then the CSP does not share the resources as per demand; otherwise, the CSP shares the resources. In addition, the CSP scales up and down the cost of the resources as per the USD rate to maximize the profit (a business model).

In the paper by Zhang et al., a constant-size CP-ABE scheme with outsourced decryption for the cloud-assisted IoT is proposed. In their scheme, the ciphertexts and the attribute-based private keys for users are both of constant size, which can alleviate the transmission overhead and reduce the occupied storage space. And, the outsourced decryption algorithm in their work is privacy-protective, which means the proxy server cannot know anything about the access policy of the ciphertext and the attribute set of the user while performing the online partial decryption algorithm. This scheme can prevent privacy from leaking out to the proxy server. And, they have rigorously proved that their scheme is selectively indistinguishably secure under the chosen-ciphertext attacks (IND-CCAs) in the random oracle model (ROM). Finally, the authors evaluate and implement their scheme and other CP-ABE schemes in terms of space and time complexity to confirm that their scheme is more suitable and applicable for cloud-assisted IoT.

The paper by Pan et al. proposed an intrusion detection model. The model can be deployed in the architecture based on cloud computing and fog computing to play its role

better. The designed intrusion detection algorithm combines kNN and sine cosine algorithm (SCA). Specifically, SCA is used to optimize the hyperparameters of kNN, thereby improving the classification accuracy of kNN. This algorithm can significantly improve the accuracy of intrusion detection and reduce the false alarm rate. In the benchmark function test, the proposed algorithm shows good optimization efficiency.

In the paper by Wang et al., a bibliometric analysis of edge computing for the Internet of things was performed using the Web of Science (WoS) Core Collection dataset. The relevant literature published in this field was quantitatively analyzed based on a bibliometric analysis method combined with VOSviewer software. The development history, research hotspots, and future directions of this field were also studied. The research results show that the number of literature studies published in edge computing for the Internet of things is on the rise over time, especially after 2017, and the growth rate is accelerating.

The paper by Ullah et al. proposed a scheme named task priority-based data-prefetching scheduler (TPDS), which tries to improve the data locality through available cached and prefetching data for offloading tasks to the edge computing nodes. The proposed TPDS prioritizes the tasks in the queue based on the available cached data in the edge computing nodes. Consequently, it increases the utilization of cached data and reduces the overhead caused by data eviction. The simulation results show that the proposed TPDS can be effective in terms of task scheduling and data locality.

In the paper by Mahmood et al., a Software Defined Networking (SDN)-based DDoS Protection System named S-DPS is proposed. It provides an early detection mechanism with mitigation of anomaly in real time. The approach offers the best deployment location of defense mechanism due to the centralized control of the network. S-DPS has demonstrated its effectiveness and efficiency in terms of Detection Rate and minimal CPU/RAM utilization, considering DDoS protection focusing on smurf attacks, socket stress attacks, and SYN flood attacks.

The paper by Ling et al. proposed multiauthority attribute-based encryption with traceable and dynamic policy updating. The proposed T-DPU-MCP-ABE is used to protect user's data privacy and solve the problem that the single authorization center load is too large, the user key leakage cannot be traced, and the data owner frequently changes the access policy in cloud storage CP-ABE access control for IoT. The scheme is constructed on prime order groups over a large attribute universe. Therefore, it is more suitable for multiuser scenarios. The authors prove that the designed scheme is static, secure, and traceable based on state-of-the-art security models. Finally, through theoretical comparison and extensive experimental comparisons, the authors show that the proposed algorithm can be better than the baseline algorithms.

In the paper by Liao et al., a systematic literature review of the current solutions and approaches available for assessing the security of software components to protect software systems for the Internet of Things is presented. This

paper searches the literature in the popular and well-known libraries, filters the relevant literature, organizes the filter papers, and extracts derivations from the selected studies based on different perspectives.

The paper by Tseng et al. proposed a generic construction of inner product predicate encryption under symmetric-key setting, called private inner product predicate encryption, from a specific key-homomorphic pseudorandom function. In addition, they show that the proposed construction is also payload-hiding, attribute-hiding, and predicate-hiding secure. With the advantage of the generic construction, if the underlying pseudorandom function can resist quantum attacks, then through the proposed generic construction, a quantum-resistant private inner product predicate encryption can be obtained. Hence, compared with other private inner product predicate encryption schemes, our scheme enjoys more robust security.

In the paper by Wu et al., a secure authentication and key agreement scheme is proposed. This scheme compensates for the imperfections of the previously proposed schemes. For a security evaluation of the proposed authentication scheme, informal security analysis, and the Burrows-Abadi-Needham (BAN) logic analysis are implemented. In addition, the ProVerif tool is used to normalize the security verification of the scheme. Finally, the performance comparisons with the former schemes show that the proposed scheme is more applicable and secure.

Conflicts of Interest

The Guest Editors declare that there are no conflicts of interest regarding the publication of the Special Issue.

*Chien-Ming Chen
Shehzad Ashraf Chaudhry
Kuo-Hui Yeh
Muhammad Naveed Aman*

Research Article

DAWM: Cost-Aware Asset Claim Analysis Approach on Big Data Analytic Computation Model for Cloud Data Centre

M. S. Mekala ^{1,2}, Rizwan Patan ³, SK Hafizul Islam ⁴, Debabrata Samanta ⁵,
Ghulam Ali Mallah ⁶, and Shehzad Ashraf Chaudhry ⁷

¹Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38544, Republic of Korea

²RLRC for Autonomous Vehicle Parts and Materials Innovation, Yeungnam University, Gyeongsan 38544, Republic of Korea

³Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

⁴Department of Computer Science and Engineering, Indian Institute of Information Technology, Kalyani, West Bengal 741235, India

⁵Department of Computer Science, CHRIST University, Bengaluru 560029, India

⁶Department of Computer Science, Shah Abdul Latif University, Khairpur, Sindh 66020, Pakistan

⁷Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Avcilar, Istanbul 34310, Turkey

Correspondence should be addressed to Shehzad Ashraf Chaudhry; sashraf@gelisim.edu.tr

Received 25 December 2020; Revised 23 February 2021; Accepted 8 May 2021; Published 29 May 2021

Academic Editor: Zhiyuan Tan

Copyright © 2021 M. S. Mekala et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The heterogeneous resource-required application tasks increase the cloud service provider (CSP) energy cost and revenue by providing demand resources. Enhancing CSP profit and preserving energy cost is a challenging task. Most of the existing approaches consider task deadline violation rate rather than performance cost and server size ratio during profit estimation, which impacts CSP revenue and causes high service cost. To address this issue, we develop two algorithms for profit maximization and adequate service reliability. First, a belief propagation-influenced cost-aware asset scheduling approach is derived based on the data analytic weight measurement (DAWM) model for effective performance and server size optimization. Second, the multiobjective heuristic user service demand (MHUSD) approach is formulated based on the CPS profit estimation model and the user service demand (USD) model with dynamic acyclic graph (DAG) phenomena for adequate service reliability. The DAWM model classifies prominent servers to preserve the server resource usage and cost during an effective resource slicing process by considering each machine execution factor (remaining energy, energy and service cost, workload execution rate, service deadline violation rate, cloud server configuration (CSC), service requirement rate, and service level agreement violation (SLAV) penalty rate). The MHUSD algorithm measures the user demand service rate and cost based on the USD and CSP profit estimation models by considering service demand weight, tenant cost, and energy cost. The simulation results show that the proposed system has accomplished the average revenue gain of 35%, cost of 51%, and profit of 39% than the state-of-the-art approaches.

1. Introduction

Nowadays, cloud computing has become a backbone for government enterprises and education sectors because of providing continuous resource (memory, CPU, and bandwidth) allocation service to ensure their application service reliability. The cloud service supplier shares the resources among end-users based on cost function's value (CF) to meet

the demand of system performance. Many service suppliers estimate the server cost based on bandwidth usage rate (BUR) and energy usage rate (EUR). As per the Gartner report, the cloud service provider (CSP) market would grow approximately 331.2 billion dollars in 2022 [1]. The cloud global report [2] confines 623.3-billion-dollar market growth rate in 2023 for data computation. The statistical analysis states that cloud computing has a notable impact on

the Internet of Things (IoT), blockchain, and soft computing measurement systems with artificial intelligence models. The tasks are divided into subtasks with relative attribute definitions through DAG theory. The DAG approach shows a prominent impact while dealing with complex workflow applications such as systematic mathematical applications [3–5]. Data analytic languages such as Hive and Pig [6–8] platforms handle the MapReduce model queries. Thus, the DAG theory’s importance tremendously changed over the past decade since it influences the service execution time and resource usage. Therefore, this issue is formulated as NP-hard [9], and many heuristic approaches resolved the same issue through resource usage consolidation [10–12].

Each machine enables a list of resource attributes (e.g., CPU, RAM size, and hard disc space) provided by CSP. In our solution, the cloud resource cost is optimized by estimating user service demands (such as CPU, IOPS, memory, and storage). For instance, an online incremental learning method has been designed in [13–15] to estimate service completion time based on heuristic algorithms by allocating the arrived service requests to the correct VM. However, these approaches have not considered server size and machine resource usage rates which causes performance delay. Therefore, in our approach, we consider CSC size, effective resource management of machines, and resource autoscaling methods; these are not present in state-of-the-art approaches. Several examinations were carried out for designing effective resource allocation methods to reduce allocation cost by satisfying service request requirements. Most current studies [16] have not considered the pricing models and data analysis models; some on-demand pricing models are considered with an inadequate measurement index. Several recent studies [17] recognize the importance of both on-demand data analytical models and reserved pricing models to minimize resource allocation costs. However, our solution assesses the server resource capacity rate, profit, and cost based on the data analysis model. The user service demand measurement algorithm is essential for profit maximization by autoscaling the resource allocation certainty.

Our research work aim is to design a novel profit optimization model for CSPs to enhance their revenue maximization (RM) by maintaining reliable quality of service (QoS). The profit optimization model must impact active server count, cost, and speed to meet the end-user satisfaction, influencing their service continuity. If there is no precise profit optimization model, then the profit and service quality and revenue generation factors will be affected. However, CSP revenue maximization has become a billion-dollar question in the competitive service computing market because of heterogeneous resource-required application tasks.

To address the listed issues, we develop two algorithms for profit maximization and adequate service reliability. First, a belief propagation-influenced cost-aware asset scheduling approach is derived based on the data analytic weight measurement (DAWM) model for effective performance and server size optimization. Second, the multi-objective heuristic user service demand (MHUSD) approach is formulated based on the CPS profit estimation model and the user service demand (USD) model with dynamic acyclic

graph (DAG) phenomena for adequate service reliability. The DAWM model classifies prominent servers to preserve the server resource usage and cost during an effective resource slicing process by considering each machine execution factor (remaining energy, energy and service cost, workload execution rate, service deadline violation rate, cloud server configuration (CSC), service requirement rate, and service level agreement violation (SLAV) penalty rate). The MHUSD algorithm measures the user demand service rate and cost based on the USD and CSP profit estimation models by considering service demand weight, service tenant cost, and machine energy cost.

1.1. Key Contributions. The trade-off between cost optimization and revenue maximization models is extensively examined in Section 2. Our manuscript’s key contributions are summarized as follows:

- (1) Develop a data analytic weight measurement (DAWM) approach to optimize service quality and price of CSP during an effective resource slicing process by considering each machine cost and revenue, and profit.
- (2) Develop a multiobjective heuristic user service demand (MHUSD) based on the CPS profit estimation model and the user service demand (USD) model to measure the user demand service rate cost by considering service demand weight, service tenant cost, and machine energy cost. Subsequently, the MHUSD algorithm also considers maximum baring wait-time of end-user to maximize CSP revenue and optimize operational energy cost.
- (3) Simulation results confirm the advantage of the proposed approaches, enhancement rate of revenue, and the CSP’s profit attributes. The impacts of mathematical key factors are being analyzed theoretically and practically.

The manuscript’s respite is designed as Section 2 briefly explains research gaps and problem statements of extant approaches. Section 3 describes the proposed system and its mathematical models with an algorithm in detail. Section 4 evaluates the investigation outcomes, and Section 5 concludes the manuscripts.

2. Related Work

This section describes the examination of related research work, which is classified into 3 steps, such as profit maximization, green data center, and graph theory-based task consolidation approaches.

2.1. Profit Maximization. Several profit maximization methods are proposed for the sustainability of green computing. We can observe the current scenario and requirement analysis of revenue in Figure 1. In [18], the broker management system has been designed to maximize the VM cost and minimize user cost. The author formulates

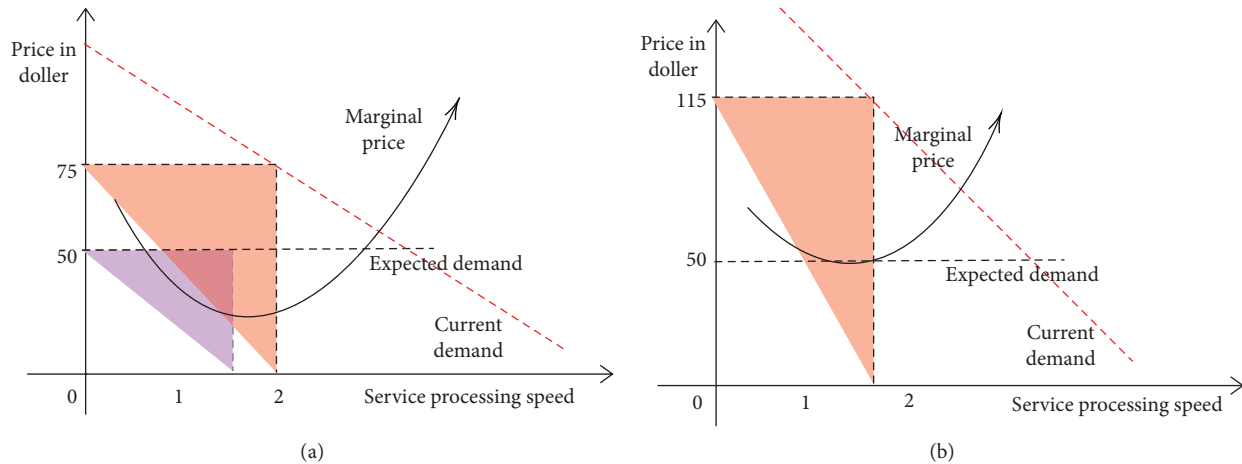


FIGURE 1: Formulation of revenue maximization. (a) Haphazard cost impact. (b) Expected cost importance.

multiserver configuration cost as a profit maximization issue, and a heuristic method has been designed to solve this issue. The delay-sensitive workload dimensionality has been examined based on a novel online heuristic approach to optimize the system's cost and profit [19]. Subsequently, the offline issue is formulated as NP-hard, and it has been resolved by a linear programming concept. In [20], a dynamic cost charging method has been designed to fix specific prices to servers as per the resource demand. A pricing approach has been designed to regulate the prices dynamically as per the demand of a kind. In [21, 22], the service penalty has diminished and enhances the profit by VM replacement approach through a mixed-integer nonlinear program called NP-hard; subsequently, a novel heuristic method has been designed to optimize the penalties and profits.

CPS profit maximization approaches have been extensively examined in this literature survey. In [23], the authors designed a stochastic programming scheme for the subscription of computing resources to maximize service providers' profit during user request uncertainty. In [24], a profit control policy has been designed to assess machine computing capacity, which decides to maximize the service provider profit. In [25–27], an SLA-based resource allocation issue has formulated with profit maximization objective with the consideration of 3 dimensions (processing, storage, and communication). In [28], a service request (SR) distribution approach is designed to enhance the profit with quality of service rate as per the service demand. In [29], the author has addressed the service provider revenue maximization issue by consolidating the service tenant cost and power consumption cost. A joint optimization scheduling model has been designed to manage delay-tolerant batch services based on pricing decisions to maximize service provider revenue [30]. In [31], the authors designed a model to maximize the service provider revenue based on the machine's tenant cost, resource demand size, and the application workload. A suitable online algorithm has been designed for the geo-distributed cloud with an adaptive VM resource cost scheme to maximize the service provider revenue [32]. The relationship between load

balance, revenue, and the cost has concentrated on maximizing the service provider revenue than state-of-the-art approaches [33]. In [34, 35], a virtual resource rental strategy has been designed based on tenant cost, task urgency, and task uncertainty to enhance provider profit.

A hill-climbing algorithm has been designed to estimate customer service satisfaction by analyzing demand mark and profit fluctuations [36]. It assesses the customer satisfaction from economic growth ratio by leveraging the cloud server configuration (CSC), task arrival rate, and profit up-downs. Therefore, the CSC directly impacts the cloud user service satisfaction rate and the inadequate customer satisfaction also has a direct impact on service request arrival rate. However, there is a lack of an accurate decision-making system and data analysis system that affects the server's profit and performance cost. A profit estimation model has been designed by considering CSC, service requirement rate, SLA, SLAV penalty rate, energy cost, tenant cost, and current CSP margin profit [37]. A server task execution speed-based power usage model is also designed to assess the CSP profit.

2.2. Green Data Centre. In [38], a mixed-integer linear program has been designed for resource allocation to optimize the data center cost and energy consumption. Green computing accomplishes the proficient process and usage of assets by limiting the vitality utilization. An enhanced ant colony approach for optimal VM execution has been developed to enhance vitality utilization and to optimize the cost of cloud environment [39–42]. The practical swarm optimization (PSO) approach resolves the task allocation issue by consolidating data center count and task demand. In distributed computing, the assets have to schedule effectively to achieve a high-performance rate. Accordingly, the multi-target PSO approach remains preferable to enhance the resource usage rates. Therefore, this approach effectively increases the usage of assets and lessens energy and makespan. The outcomes delineated that the proposed strategy multiobjective practical swarm optimization

(MOPSO) performance is quite beneficial than concerned existing models. A VM scheduling approach has been designed based on multidimensional resource imperatives, for example, link capacity, to diminish the quantities of dynamic PMs to preserve energy utilization. The 2-step heuristic approach resolves the VM scheduling through migration and VM positioning models [43, 44]. The designed method has consolidated the execution time than extant systems in a simulation platform. Asset overburdening is still an issue, and live relocation does not uphold the change of VM performance. In [45], the energy-aware asset allocation approach has been investigated to improve the energy productivity of a server farm without SLA negotiations. An asset scheduling strategy with a hereditary method has been proposed to improve the usage of assets and save the expense of energy in distributed computing [46, 47]. It utilizes a migration approach dependent on 3 load degrees (CPU usage, the throughput of organization, and pace of circle I/O). The calculation succeeds in improving the usage of assets, and saving energy by run-time asset scheduling is high. An energy preservation system is classified by assorting the asset into four distinct classifications (CPU, memory, storage, and networks). Additionally, the author designed a unique asset scheduling system dependent on cloud assets' energy streamlining with assessment technique [48]. The study [49] evaluates every machine's fitness value, which helps assess the machine rank based on the performance and resource usage rate. However, the machine rank evolution process consumes more time which influences the performance, and task scheduling policy leads to high-performance cost. The complexity rate is high over large-scale frameworks.

2.3. Graph Theory-Based Resource/Task Scheduling. Dynamic acyclic graph (DAG) has been used for task scheduling by considering PM capacity and task resource weight to formulate the issue [50]. Here, $X[i, j]$ matrix identifies the errand evolution time of all VMs under different instances. To address all these issues, we design a data analytic weight measurement (DAWM) approach to optimize a cloud service provider's quality and price during an effective resource slicing process by considering each machine's cost and revenue, and profit. The entire cost does not iteratively consider traditional DAG-based models during the measurement of data analysis. Subsequently, we design a multiobjective heuristic user service demand (MHUSD) algorithm based on the CPS profit estimation model and the user service demand (USD) model to measure the user demand service rate and cost by considering service demand weight, service tenant cost, and machine energy cost.

3. DAWM System Model

A belief propagation-influenced data analysis model is designed for CSP profit maximization by formulating DAG task and resource scheduling policy, as shown in Figure 2. The CSP receives a service request from the cloud

user, and by default, the CSP has three service modes: on-demand, advanced reservation, and spot resource allocation, which helps to slice the resources as per resource demand. As per the received service request, the CSP assesses its demand, cost, performance, profit, and required server size factors. The CSP consolidates the overprovisioning machines by optimizing the service execution cost and machine asset usage. Cloud service suppliers drive the data utility analytic method on machines to classify the high- and low-resource usage rate machines, preserve CDC usage and performance cost, and avoid instant repudiations/migrations.

It classifies adaptive servers after the first iteration by concocting an exact data analytic weight measurement (DAWM) model. First, a belief propagation influences a cost-aware asset scheduling approach based on the data analytic weight measurement (DAWM) model, which effectively optimizes the performance cost and server size. The DAWM model classifies prominent servers to preserve the server resource usage and cost during an effective resource slicing process by considering each machine execution factor (remaining energy, energy and service price, workload execution rate, service deadline violation rate, cloud server configuration (CSC), service requirement rate, and service level agreement violation (SLAV) penalty rate). Second, the multiobjective heuristic user service demand (MHUSD) approach is processed based on the CPS profit estimation model and the user service demand (USD) model with dynamic acyclic graph (DAG) phenomena for adequate service reliability. The MHUSD algorithm prognosticates the user demand service rate and cost based on the USD and CSP profit estimation models by considering service demand weight, service tenant cost, and machine energy cost. The USD model estimates the resource service demand to estimate the profit and revenue gain and the system's performance cost. The CSP profit estimation model helps assess the service profit by forecasting the server's performance cost, energy usage, and resource tenant cost. Each subsection describes a sub-component of the framework mathematically and theoretically.

3.1. Cloud Service Provider Model. The CSP offers various services to cloud end-users. For instance, infrastructure is a service, where the resources are being offered as VMs to meet the end-user satisfaction by running their applications. The user service request (USR) is submitted to the service provider, which runs on a multiserver system to deliver the response for the received service requests. Consider a multiserver system (MSS) enables N homogeneous servers with m speed, and these are modeled based on the $(M/M/M)$ queuing system. Assume that the MSS framework receives a number of user service requests with a rate of u . The service time $v = (x/m)$, where x refers to required instruction count to execute the USR and mean $\bar{v} = (\bar{x}/m)$. The service rate of the USR is denoted as $q = (1/\bar{v}) = (m/\bar{x})$. The server utilization rate is estimated with equation (1), and it is denoted with Z :

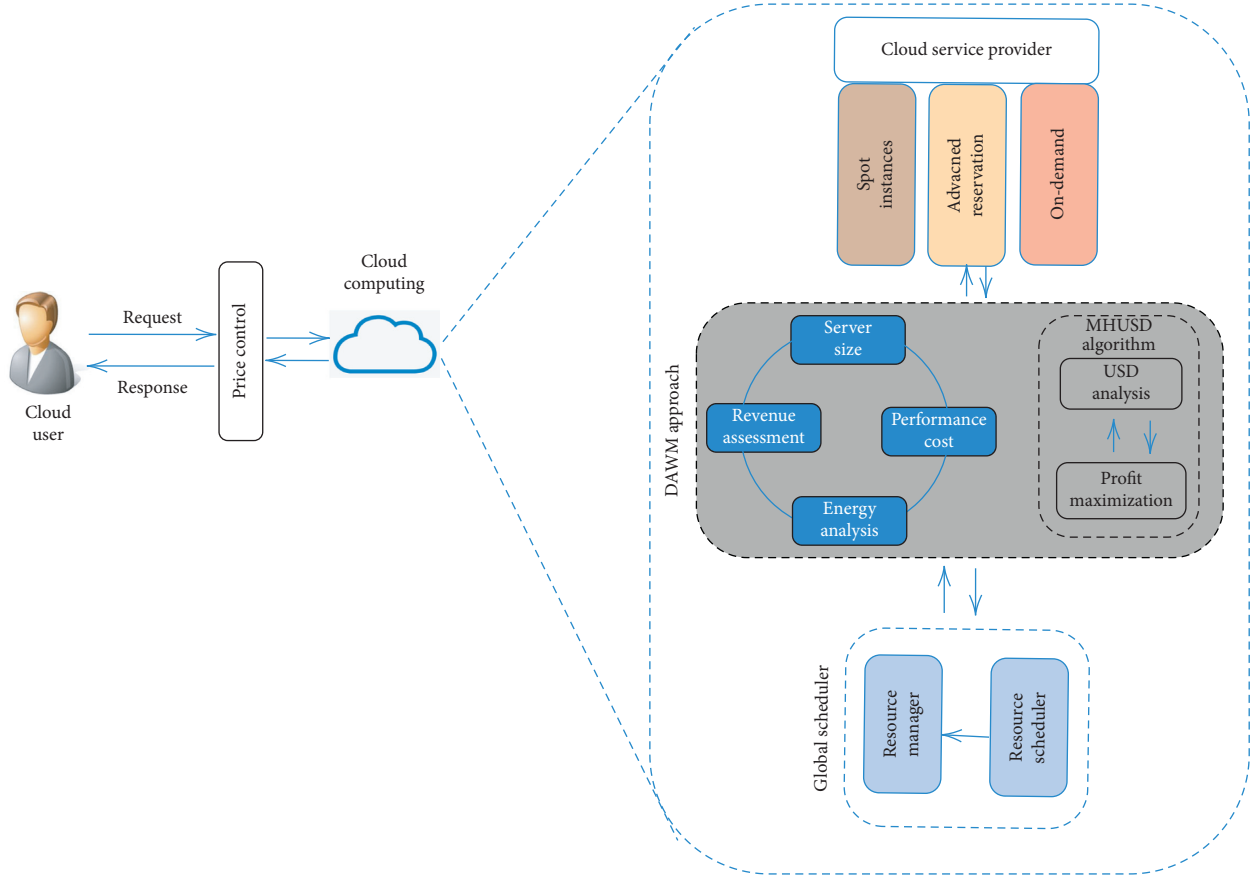


FIGURE 2: DAWM system model.

$$Z = \frac{u}{N \cdot q} = \frac{u}{N \cdot (m/\bar{x})} = \frac{u \cdot \bar{x}}{N \cdot m}, \quad (1)$$

$$\rho_r = \begin{cases} \rho_0 \frac{(N \cdot \rho)^r}{r!}, & r \leq N, \\ \rho_0 \frac{(N^N \cdot \rho^r)^r}{N!}, & r > N, \end{cases} \quad (2)$$

where ρ_r refers to probability of r service requests which are executing at a server. In case if there are no tasks/service requests, then the probability of zero service request is

$$\rho_0 = \left(\sum_{r=0}^{N-1} \frac{(N \cdot \rho)^r}{r!} + \frac{(N \cdot \rho)^N}{N!} \cdot \frac{1}{1-Z} \right). \quad (3)$$

Subsequently, ρ_b is the probability of new arrived SRs, which should wait when the server system is busy executing assigned tasks where ρ_N refers to probability of all

N SRs. The probability density function is defined with equation (5), and d refers to service waiting time:

$$\rho_a = \sum_{r=N}^{\infty} \rho_r = \frac{\rho_N}{1-N}, \quad (4)$$

$$\rho_d en(t) = (1 - \rho_a) \cdot d + N \cdot q \cdot \rho_N \cdot e^{-(1-\rho)N \cdot q(t)}. \quad (5)$$

Figure 3 illustrates the DAG task classification and scheduling scheme that accomplishes by evaluating cost price/unit of the machine, which is magnified with ample of time required for task completion. Therefore, for instance, n is the number of VMs of $F[i]$ type with weight $W[r[i]]$, $\forall 1 \leq i \leq n$. Let τ be the required time to finish all the errands on a set of VMs through the DAG-based approach. The collected value/unit time is $\sum_{1 \leq i \leq n} W[r[i]]$. Appropriately, the complete performance weight is $(\varphi, \vartheta(t), \omega(t))$, and it is characterized as

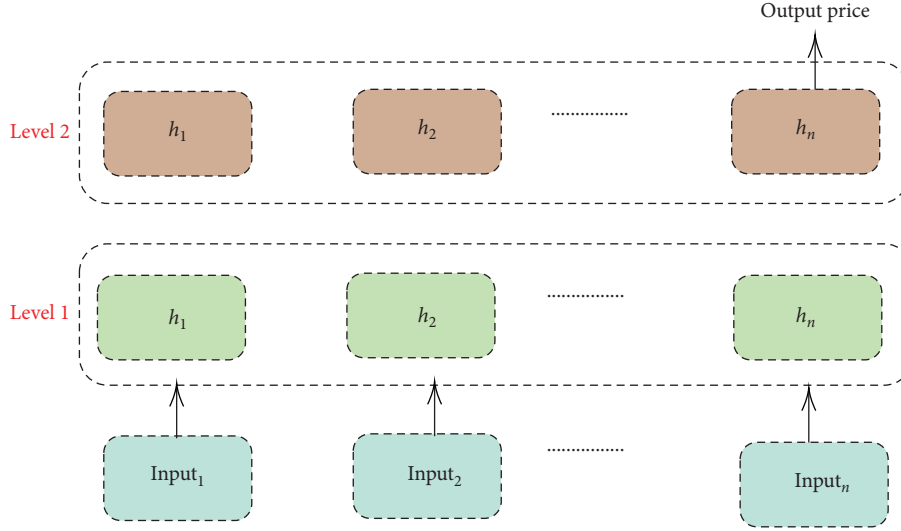


FIGURE 3: Belief propagation-influenced MRS cost assessment submodel.

$$\text{Weight}(\varphi, \vartheta(t), \omega(t)) = \sum_{1 \leq i \leq n} W[r[i]] \times \tau. \quad (6)$$

3.2. *Service Level Agreement Model.* The SLA is a method which maintains a trade-off between price and service quality between end-user and CSP. Here, the required service attribute x is executed within the response time T , to meet the application deadline:

$$S(x, T) = \begin{cases} cx, & \text{if } 0 \leq d \leq \left(\frac{b}{m_0} - \frac{1}{m}\right) \cdot x, \\ \left(c + \frac{b \cdot p}{m_0} - \frac{p}{m}\right)x - p \cdot d, & \text{if } \left(\frac{b}{m_0} - \frac{1}{m}\right) \cdot x < d \leq \left(\frac{c}{p} + \frac{b}{m_0} - \frac{1}{m}\right) \cdot x, \\ 0, & \text{if } d > \left(\frac{c}{p} + \frac{b}{m_0} - \frac{1}{m}\right) \cdot x, \end{cases} \quad (7)$$

where a is the service cost/unit, d is the penalty cost if any SLA violation, b is the constant weight of SLA, and m_0 is the expected service processing speed. There are three conditions listed even the service request has under waiting time. Therefore, $T = ((d + x)/m)$:

- (1) If d has low value than $bc \times m_0$, it provides high-quality, reliable service
- (2) If d is in-between the $((b/m_0) - (1/m)) \cdot x < d \leq ((c/p) + (b/m_0) - (1/m)) \cdot x$, time interval leads to moderate service quality
- (3) If d is longer than $((c/p) + (b/m_0) - (1/m)) \cdot x$, then the service is free because the service request waited long time in queue

Equation (7) is used to assess the prognosticated service charge of the CSP based on 5 parameters:

c , p , b , d , and m . Here, c refers to service cost/unit, p refers to SLAV penalty cost, m_0 refers to expected service speed, b refers to SLA constant weight, and d is the average service waiting time.

3.3. *User Service Satisfaction Model.* User service satisfaction (USS) is estimated in two ways: quality of service (QoS) and price of service (PoS). QoS describes the discrepancy between users' expectations (how to server SR) and users' perceptions (how to perform service). The user's quality of service ($\eta_i^{sq}(x, T)$) is evaluated with

$$\eta_i^{sq}(x, T) = \begin{cases} 1, & \text{if } J_{ac} \geq J_{ex}, \\ e^{-|(J_{ac} - J_{ex})/J_{ex}|}, & \text{if } J_{ac} < J_{ex}. \end{cases} \quad (8)$$

The $\eta_i^{tc} = e^{((S_{ex}-S_{ac})/S_{ex})}$ is a fundamental expression to assess the price of service (PoS) with equation (9). Here, S_{ex} and S_{ac} refer to expected cost and actual cost, respectively:

(1) If $S_{ex} = S_{ac}$, then $\eta_i^{tc} = 1$, shows there is not impact on user satisfaction

(2) If $S_{ex} > S_{ac}$, then it leads to the higher service cost ($\eta_i^{tc} < 1$), and it decreases by increasing the actual price

(3) If $S_{ex} < S_{ac}$, then it leads to the lower service cost ($\eta_i^{tc} > 1$), and it increases by decreasing the actual price

$$\eta_i^{tc}(x, T) = \begin{cases} 1, & \text{if } 0 \leq d \leq \left(\frac{b}{m_0} - \frac{1}{m}\right) \cdot x \\ e^{((1/m)+(d/x)-(b/m_0)) \cdot (p/c)}, & \text{if } \left(\frac{b}{m_0} - \frac{1}{m}\right) \cdot x < d \leq \left(\frac{c}{p} + \frac{b}{m_0} - \frac{1}{m}\right), \\ e, & \text{if } d > \left(\frac{c}{p} + \frac{b}{m_0} - \frac{1}{m}\right) \cdot x. \end{cases} \quad (9)$$

The USS (η_i^{sa}) is defined as product of service price and quality of service ($\eta_i^{sa} = \eta_i^{sd}(x, T) + \eta_i^{tc}(x, T)$) with (10).

Such that,

$$\eta_i^{sa} = \begin{cases} 1, & \text{if } 0 \leq d \leq \left(\frac{b}{m_0} - \frac{1}{m}\right) \cdot x, \\ e^{((1+(1/m)-(b/m_0)) \cdot (p/c)+d) / (x \cdot ((p/c)-(1/((b/m_0)-(1/m)))))}, & \text{if } \left(\frac{b}{m_0} - \frac{1}{m}\right) \cdot x < d \leq \left(\frac{c}{p} + \frac{b}{m_0} - \frac{1}{m}\right), \\ e^{2-d/((b/m_0)-(1/m)) \cdot x}, & \text{if } d > \left(\frac{c}{p} + \frac{b}{m_0} - \frac{1}{m}\right) \cdot x. \end{cases} \quad (10)$$

The product of sum is calculated with equation (8) and equation (9).

3.4. User Demand Service Estimation Model and Algorithm.

The user service demand weight factor ($\eta_{i,k}^{\text{exp ec}}$) assessment plays an essential role to optimize the cost of cloud service provider, and it is estimated with equation (11):

$$\eta_{i,k}^{\text{exp ec}} = \sum_{k=1}^x K_i (\chi_k - \gamma_k), \quad (11)$$

where x refers to a list of service attributes, K_i refers to the service attribute weight, χ_k refers to the attribute perception, and γ_k refers to the attribute expectation.

The service demand is formulated as the product of potential demand and user service demand weight factor. It is defined as

$$\eta_{i,k}^{\text{dema}} = 0.25 \times (\alpha + \beta \times \eta_{i,k}^{\text{exp ec}}), \quad (\text{where } \alpha, \beta > 0), \quad (12)$$

where α and β refer to constant basic demand and constant potential demand. Subsequently, both values must be greater than >0 , such as $\alpha, \beta > 0$.

MHUDS algorithm 1 assesses the user service demand adequately. Lines 1-2 define the entail parameters and attributes for estimation of the user service demand. Line 4 assesses all the service attributes of the cloud service provider and also checks the CPS set. Line 5 helps assess the lower and upper bound value that should not be less than $<\mathbb{R}$. Line 6 estimates the median value of the service attribute demand. Line 7 assesses the $\eta_{i,x}^{\text{dema}}(u_m^k)$ which should not be less than 0. $\eta_{i,x}^{\text{dema}}(u_m^k)$ refers to user service demand of attribute k with middle-range value. Similarly, the rest of the two variables refer to higher and lower values of the user service demand rate. Lines 12-15 are used to update the concerned value at each iteration of time.

3.5. CPS Profit Estimation Model. The CSP profit is assessed based on the gap between the profits gained by acquiring services to users and the monetary cost of processing user SRs. Equation (13) is defined with function number and

$$S = -0.25 \times \left(\frac{\rho_a \cdot c \cdot \bar{x}}{\rho_a \cdot c \cdot \bar{x} ((N \cdot m - u\bar{x}) \times ((b/m_0) - (1/m)) + 1)^2} \right) = S(N, m), \quad (13)$$

$$\varphi = S(N, m) \times \eta_{i,k}^{\text{dema}}, \quad (14)$$

where $\eta_{i,k}^{\text{dema}}$ refers to USD based on user service attribute value. The CSP cost is defined as a paid infrastructure tenant cost and the power cost of system function, and it is assessed with equation (15). The server energy consumption is also estimated with equation (17):

$$\vartheta(t) = N \cdot s \times t, \quad (15)$$

$$\xi(t) = N \cdot (\Omega_{nst} \times \partial + \Omega_{st}) \cdot t \cdot \xi_s^n, \quad (16)$$

where ∂ refers to server usage, Ω_{nst} refers to dynamic power usage, and Ω_{st} refers to static power usage. Assuming that $\xi_s^n(t)$ refers to energy usage cost at processing time t , the electricity bill ($\omega(t)$) is defines as

$$\omega(t) = \xi(t) \times \xi_s^n(t). \quad (17)$$

The CSP profit at t is described as the revenue minus from the rental and electricity cost, and it is estimated with equation (18):

$$G(N, m) = \varphi - \vartheta(t) - \omega(t). \quad (18)$$

3.5.1. CSP Profit Maximization Factor. The probability of having N SRs is described with equation (19). The Taylor series influences approximately $(N! \approx \sqrt{2\pi N} (N/e)^N)$ to assess the CSP profit as follows:

$$\rho_N = \rho_0 \times \frac{(N \cdot Z)^N}{N!}, \quad (19)$$

$$\rho_N = \frac{1 - Z}{1 - Z \times \sqrt{2\pi N} ((e^{Z-1}/Z))^N + 1}, \quad (20)$$

updated derivation

$$\rho_N = \frac{1}{\sqrt{2\pi N} ((e^{Z-1}/Z))^N + 1}. \quad (21)$$

The CSP maximized profit assess as follows:

$$S(N, m) = -0.25 \times \rho_a \cdot c \cdot \bar{x} \times \left(\frac{1}{((N \cdot m - u\bar{x}) \times ((b/m_0) - (1/m)) + 1)^2 \times (1/\sqrt{2\pi N} ((e^{Z-1}/Z))^N + 1)} \right). \quad (22)$$

3.6. DAG Task Scheduling Methodology. The errands are assigned through a computational method, which comes under the DAG-based process by considering the framework's performance weight. It can be observed in Figure 4. We characterize a graph $G = \{V, E\}$. $V = \{v_1, v_2, v_3, \dots, v_n\}$ where v_i speaks to a comparing errand t_i and it executes consecutively on a machine. $E = e_1, e_2, e_3, \dots, e_m$ remains priority connection among errands because of information reliability. An errand is not initiated until the last errand remains finished.

Because of dissimilar conditions in the cloud, each PM ability remains to differ. Therefore, we consider the $X[i, j]$

server speed (i.e., N and m). The average revenue of CSP is estimated as a product of the expected cost of SR and user service demand:

matrix to identify and for a keen track of each errand processing time t_i on j^{th} VM. Here, we have not considered weight and performance factors to measure the assets. In our system, we deliberately utilize a matrix to measure performance time on various VMs, rather than utilizing a consistent weight factor to estimate execution time. As per the data analysis model dataset, we measure each level (L_i) of the convolution network with DAG-based spark. Specifically, each spark stage alludes a vertex, and the connection among 2 phases is compared with organized point. The apexes with 0 degree remain reflected as phases that complete in parallel (P_i). The 0-degree vertices of DAG indicate with L . The

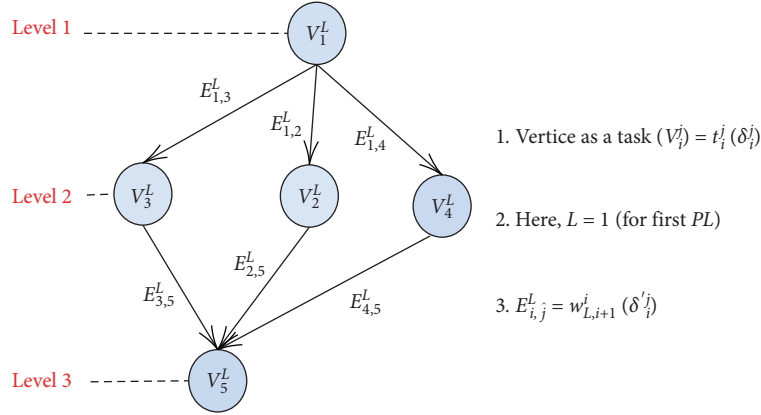


FIGURE 4: Representation of DAG task, where $v_1 \rightarrow v_5$ entry and exit nodes with weight factor.

organizing system remains recursively performed and forwards the outcome to any phase of DAG. According to equation (23), we measure most outrageous performance time of all processing phases in parallel (P_i) and that recursively upgrades the task finish time:

$$\text{Max}_{i \in S} T_L^i \leq T_{\text{Task}} \leq \sum_{i \in S} T_L^i. \quad (23)$$

3.7. Estimation of Optimal Price. The price-demand function estimates optimal price of service by considering the trade-off between service price ϕ and the concern service demand Δ based on their service request mode such as on-demand service, reserved service, and spot instance service. It is formulated as

$$\Delta = \Delta_{re} + (\Delta_{od} - \Delta_{re}) \times \left(\frac{\phi - \phi_{re}}{\phi_{od} - \phi_{re}} \right), \quad (24)$$

where Δ_{od} refers to price-demand of on-demand service and Δ_{re} refers to price-demand of reserved service, and similarly, for price, ϕ_{od} refers to price for on-demand service and ϕ_{re} refers to price for reserved service.

Theorem 1. Let us assume that the CSP considers \hbar units of time. If service price is ϕ and average service execution time is t , then the anticipated service price is

$$\phi_{\text{exp } e} = \hbar \phi \left(\frac{1}{1 - e^{-(\hbar \hat{t})}} \right). \quad (25)$$

Proof. The CSP considers \hbar units of time, the optimal price is measured with average service execution time t , and it can be measured as

$$\phi(t) = \hbar \left[\frac{t}{\hbar} \right] \times \phi. \quad (26)$$

It is defined as follows: the service request price is $(n + 1)\hbar\phi$ in $(n\hbar, (n + 1)\hbar]$ time interval.

The probability distribution function of t is

$$g(\tau) = \frac{1}{\tau} \times e^{-(\hbar \hat{t})}. \quad (27)$$

The expected price is

$$\begin{aligned} \phi_{\text{exp } e} &= \int_0^{\infty} \phi(t) \cdot g(t) dt, \\ &= \sum_0^{\infty} \int_{n\hbar}^{(n+1)\hbar} g(t) (n+1) \cdot \hbar \cdot \phi dt \\ &= \sum_0^{\infty} (n+1) \cdot \hbar \cdot \phi \left(-e^{-(\hbar \hat{t})} \right)_{n\hbar}^{(n+1)\hbar} \\ &= \hbar \cdot \phi \sum_0^{\infty} e^{-(n\hbar/\hat{t})}, \\ &= \hbar \cdot \phi \lim_{n \rightarrow \infty} \frac{1 - \left(e^{-(n\hat{t})} \right)^n}{1 - \left(e^{-(n\hat{t})} \right)} \\ &= \hbar \cdot \phi \frac{1}{1 - \left(e^{-(n\hat{t})} \right)}. \end{aligned} \quad (28)$$

Hence, the theorem is proved and the forecasting service arrival demand is approximately $u = \Delta u_{\text{max}} = \Delta_{re} + (\Delta_{od} - \Delta_{re}) \times ((\phi - \phi_{re}) / (\phi_{od} - \phi_{re})) u_{\text{max}}$.

The forecasting service price is $S_{\text{exp } e} = \phi - \text{CSPcost} = \phi - n\phi_{re}$. So, the maximum price must have to measure $(\partial S_{\text{exp } e} / \partial \phi) = 0$, such that

$$\frac{\partial u}{\partial \phi} = \left(\frac{\Delta_{od} - \Delta_{re}}{\phi_{od} - \phi_{re}} \right) u_{\max}, \quad \text{where } \frac{\partial u \phi}{\partial \phi} = u + \left(\frac{\Delta_{od} - \Delta_{re}}{\phi_{od} - \phi_{re}} \right) u_{\max} \phi, \quad (29)$$

where $s^{\text{los}} = ((Z^n e^{n(1-Z)})/\sqrt{2\pi n})$ refers to loss of server profit, but the probability of expected server profit loss is

$$\begin{aligned} \frac{\partial s_{\text{exp ec}}^{\text{los}}}{\partial \phi} &= \frac{1}{\sqrt{2\pi n}} \left[\frac{\partial (e^{n(1-Z)})}{\partial \phi} Z^n + e^{n(1-Z)} \frac{\partial Z^n}{\partial \phi} \right] \\ &= \frac{1}{\sqrt{2\pi n}} \left[\left(Z^n e^{n(1-Z)} \left(-\frac{1}{\ell} \frac{\partial u}{\partial \phi} \right) \right) \right. \\ &\quad \left. + Z^{n-1} \cdot n \cdot e^{n(1-Z)} \left(-\frac{1}{n\ell} \frac{\partial u}{\partial \phi} \right) \right] \quad (30) \\ &= \frac{Z^n e^{n(1-Z)}}{\sqrt{2\pi n}} \frac{1-Z}{\ell Z} \frac{\partial u}{\partial \phi} \\ &= s^{\text{los}} \frac{1-Z}{\ell Z} \frac{\partial u}{\partial \phi}, \quad \text{since } s^{\text{los}} = \frac{Z^n e^{n(1-Z)}}{\sqrt{2\pi n}}. \end{aligned}$$

Subsequently, the probability of forecasting service price is

$$\begin{aligned} \frac{\partial s_{\text{exp ec}}}{\partial \phi} &= \left(u + \frac{\partial u}{\partial \phi} \phi (1 - s^{\text{los}}(t)) + u \phi t \left(-s^{\text{los}} \frac{1-Z}{\ell Z} \frac{\partial u}{\partial \phi} \right) \right) \\ &= u(1 - s^{\text{los}}(t)) \\ &\quad + \phi t \frac{\partial u}{\partial \phi} \left[(1 - s^{\text{los}}(t)) - s^{\text{los}}(t)n(1-Z) \right]. \quad (31) \end{aligned}$$

3.8. Estimating Optimal Price. In Algorithm 2, the partial derivative is formulated through s^{los} . It formulates accurate service price though the service arrival rate is high with low profit loss. Lines 1–3 define the input variables, and line 4 applies the models to all arrived service requests. Lines 6–9 estimate the optimal price demand, and lines 10–19 estimate optimal price value based on equations (31) and (13).

3.9. DAWM Algorithm for Cloud Server Size and Cost Analysis. Algorithm 3 assess the server size and performance cost. It assesses the customer satisfaction from the machine economic growth ratio by leveraging the cloud server configuration (CSC) called server size, task arrival rate, and performance cost of the machine. Therefore, the CSC has a direct impact on the cloud user service satisfaction rate and the inadequate customer satisfaction, and it also has direct impact on service request arrival rate. Line 1 defines the essential input parameters to accomplish the objectives. Lines 2–5 assess the service execution cost using equation (13) and update the machine matrix $H[i, j]$, for effective prognostication of server configuration size. Lines 6 and 7

update the all machine execution speed rates and maintained in an array. Lines 8 to 10 assess the performance cost in association with CSC (s), service resource requirement rate (K), SLAV penalty rate (L), and energy and resource tenant cost. Lines 12–15 update the iterative value to mitigate performance rate and system execution cost.

4. Experimental Result Analysis

The proposed DAWM is simulated with real data in MATLAB R2017b, and the system specifications are 8 GB DDR4 memory and an Intel Core i7-6700HQ CPU with 2.6 GHz. We consider DAG $[V, E]$ consisting 25–150 sensors. Every network enables 5% of data centres in the network size, and its capacity varies from 5000 to 75000 GHz. The active servers are varying from 1000 to 1500. The idle server constant energy consumption is 90 – 180 Watt; else the energy consumption is measured based on its energy usage rate, and it is in range $[0.5, 1.5]$; energy price is $[15, 55]/\text{Mwh}$. The link bandwidth between sensors varies from 1500 to 25,000 Mbps and delay transmission is 3 – 6 ms. The revenue gain is $[0.15, 0.25]$, which is not static. Each service execution bandwidth is set from 15 – 25 Mbps, computing demand is 3 – 5 GHz, and the execution of each service is 5 – 30 (data packets/ms). The simulation parameters related to power cost, constant workloads, CSC, service requirement rate, SLAV penalty rate, energy cost, tenant cost, and current CSP margin profit are listed in Table 1.

Figure 5 illustrates the average execution time required to process the user service request. It has been compared with four state-of-the-art approaches (SPEA2, COMCPM, NSGA-II, and OMCPM) which are published recently. It is noticed that the proposed approach has high-performance rate than remaining approaches such as 41.2%, 55.56%, 59.89%, and 61.52% faster than SPEA2, COMCPM, NSGA-II, and OMCPM, respectively.

Figure 6 illustrates profit, revenue, and cost of the proposed system and SPEA2, COMCPM, NSGA-II, and OMCPM approaches. The proposed system achieved moderately high revenue by 10%, 8.1%, 8.9%, and 8.91% than SPEA2, COMCPM, NSGA-II, and OMCPM approaches. Subsequently, our approach achieves 2.31%, 2.01%, 1.7%, and 1.37% high profit than four approaches, since our approach estimates the demand of service request and it analyses the machine performance before assigning the load. The reason is that user service request (USR) is submitted to the service provider, which runs on a multiserver system to deliver the response for the received service requests. The CSP assesses the machine data with our deep learning data analytical model. It makes an accurate decision to enhance the system performance by preserving service cost and to enhance the revenue gain consolidating each machine performance. The second reason is that the task is being scheduled base on DAG theory which influences the energy and resource of the system leads to enhance the revenue and optimizes the service request cost.

Figure 7 shows the user service demand flexibility impact. We can observe that the active cloud server (from 15 to 75) count and the processing speed m of active servers are

```

input: CPS:  $N = \{N_1 + N_2 + N_3 + \dots + N_n\}$ 
output: user demand service
(1) Let initialize  $\alpha \neq 0, \beta \neq 0, \eta_{i,k}^{\text{expect}} \neq 0$ ;
(2) Int  $u$ , define range  $[u_l^k, u_h^k], \eta_{i,x}^{\text{dema}}(u_l^k) > 0, \eta_{i,x}^{\text{dema}}(u_h^k) < 0$ ;
(3) for each  $N_i \in N$  do
(4) Estimate  $\eta_{i,k}^{\text{dema}} = 0.25 \times (\alpha + \beta \times \eta_{i,k}^{\text{expect}}) - u$ ;
(5) while  $(\eta_{i,x}^{\text{dema}}(u_l^k) - \eta_{i,x}^{\text{dema}}(u_h^k)) > \mathbb{R}$  do
(6)  $\eta_{i,k}^{\text{dema}}(u_m^k) = ((\eta_{i,k}^{\text{dema}}(u_l^k) - \eta_{i,k}^{\text{dema}}(u_h^k))/2)$ ;
(7) if  $\eta_{i,x}^{\text{dema}}(u_m^k) < 0$  then
(8) Assign  $\eta_{i,x}^{\text{dema}}(u_h^k) \leftarrow \eta_{i,x}^{\text{dema}}(u_m^k)$ ;
(9) else
(10) Assign  $\eta_{i,x}^{\text{dema}}(u_l^k) \leftarrow \eta_{i,x}^{\text{dema}}(u_m^k)$ ;
(11) end
(12) Update  $\eta_{i,x}^{\text{dema}}(u_l^k)$  and  $\eta_{i,x}^{\text{dema}}(u_m^k)$ ;
(13) Estimate  $\eta_{i,k}^{\text{dema}}(u_m^k) = ((\eta_{i,k}^{\text{dema}}(u_l^k) - \eta_{i,k}^{\text{dema}}(u_h^k))/2)$ ;
(14) end
(15) Confine it as potential value for next iteration  $\eta_{i,k}^{\text{dema}}(u_m^k)$ ;
(16) Return user demand service value.
(17) end

```

ALGORITHM 1: MHUDS algorithm.

```

input:  $u, t, n, \phi_{re}, \phi_{od}, \Delta_{re}, \Delta_{od}$ 
output: optimal price of service
(1) Let  $\phi_{\text{opti}} = -\infty, \Delta_{\text{opti}} = -\infty$ ;
(2)  $\phi_{st} \leftarrow$  least price, server usage < 1;
(3)  $\phi_{en} \leftarrow \phi_{od}$ ;
(4) for each  $N_i \in N$  do
(5) Estimate  $\Delta_{st}$  and  $\Delta_{ed}$  using equations (30) and (13);
(6) if  $\Delta_{st} \times \Delta_{ed} > 0$  then
(7)  $\phi_{\text{opti}} = \phi_{st}$ ;
(8) Estimate  $\Delta_{\text{opti}}$  using (13) and with  $S_{\text{expect}} = \phi - \text{CSPcost} = \phi - n\phi_{re}$ ;
(9) end
(10) while  $\Delta_{st} \times \Delta_{ed} > \text{error}$  do
(11)  $\phi_{\text{mid}} = ((\phi_{st} + \phi_{en})/2)$ ;
(12) Estimate  $\Delta_{\text{mid}}$  using (13) and (31);
(13) if  $\Delta_{st} \times \Delta_{ed} > 0$  then
(14)  $\phi_{st} \leftarrow \phi_{\text{mid}}$ ;
(15) else
(16)  $\phi_{en} \leftarrow \phi_{\text{mid}}$ ;
(17) end
(18) end
(19)  $\phi_{\text{opti}} = ((\phi_{st} + \phi_{ed})/2)$ ;
(20) end

```

ALGORITHM 2: Optimal price estimation algorithm.

high, but there is no impact on the service execution demand rate. If the server count increases, then the user service demand execution rate does not increase, and it is sometimes stable to cope up the reliable quality of service with adequate computing performance. If the USD is high, the server system is frequently unable to meet the service demand requirement synchronously. In such cases, if the customer waits for a long time, then the USD rate becomes low due to low service demand. Usually, the USD may remain constant when the USD market is stable, which would not affect third-party factors.

Figure 8 shows CSP profit outcomes. As we can observe, the profit rate is drastically decreased when the active servers are increased from 15 to 75. The high server processing speed m has no impact as we expected. The profit ratio is increased due to the USD rate increment than the new active server cost. The revenue enhancement and server size factors are not impacting server cost, but USD will get diminished due to the decrement of CSP profit. Consequently, the profit returns stable when the USD becomes constant. Figure 9 shows the server processing speed comparative study. The server processing speed is decreased when the server size

input: (1) Host set: $\mathbf{N} = \{N_1 + N_2 + N_3 + \dots + N_n\}$,
(2) Ex : execution time matrix of host
(3) C : cost weight matrix of host/VM
output: performance cost of server
(1) Let $T = \{T_1 + T_2 + T_3 + \dots + T_i\}$
(2) **for each** $N_i \in \mathbf{N}$ **do**
(3) Find minimum cost-effective host (6) and (13)
(4) $N_j[i] = N_j + H[i, j]$
(5) **end**
(6) $T_{tot} = \sum_{\forall T_i} T_i$
(7) Update $N_j[i]$, $\leftarrow SortHostCostQueue(\varphi, \vartheta(t), \omega(t))$
(8) **for each** i to h_m **do**
(9) $\lambda = Cost(K, S, L) = \sum_{1 \leq i \leq K} C[S[i]] \times T_{ToT}$
(10) $R_i = Cost(N_j) - \lambda_i$
(11) **end**
(12) $K' = K - \lambda_i$
(13) **for each** $R_i \in N_n$ **do**
(14) λ_i^{++}
(15) **end**
(16) Return performance cost of server

ALGORITHM 3: DAWM algorithm.

TABLE 1: Simulation parameters.

S. no.	Notation	Value
1.	m_0	1.5 BIPS
2.	b	5
3.	\bar{x}	1.5 BI
4.	c	20 units/BI
5.	s	20 units/sec
6.	N_c	9.5
7.	p	5
8.	Ω_{st}	2 Watts/sec
9.	$\xi_s^n(t)$	0.1 unit/Watt \times sec

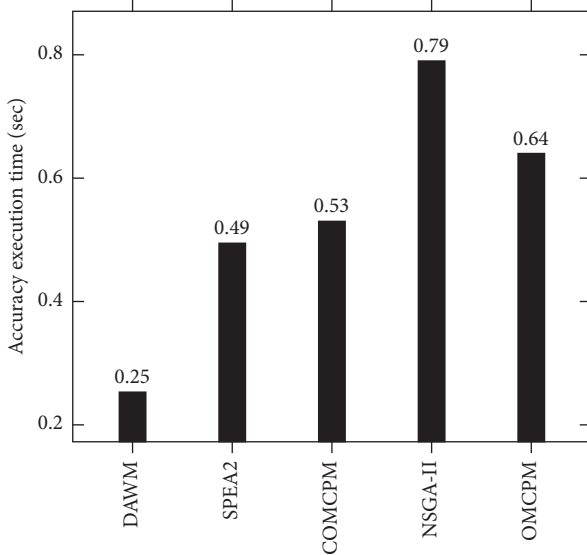


FIGURE 5: Average execution time of DAWM, SPEA2, COMCPM, NSGA-II, and OMCPM.

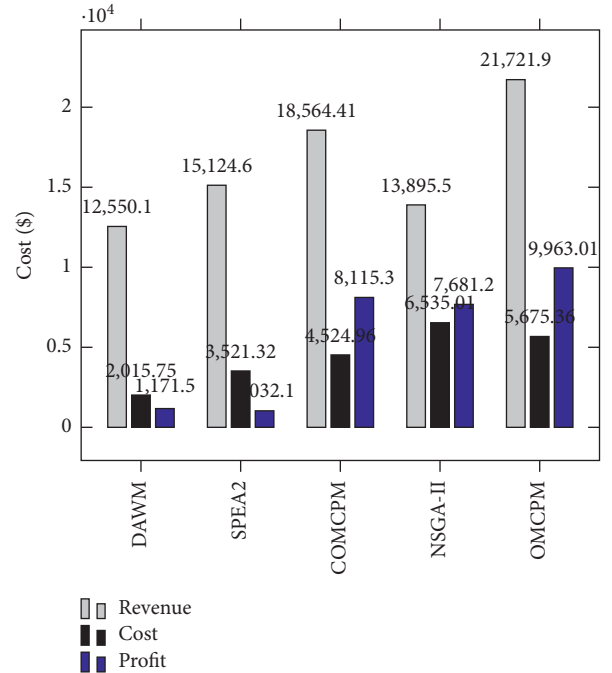


FIGURE 6: Cost, profit, and revenue of DAWM, SPEA2, COMCPM, NSGA-II, and OMCPM.

increases; the computation size is fixed, which restricts the execution of the services. The increased server count demands to decrease the systems service execution speed. Figure 10 illustrates the increased profit during server size, and USD rates are increased. The high-computation-required USDs are led to enhance the CSP profit. We can observe that the USD is moderate due to server size enhancement. We noticed that if active servers are less but the server speed is high, the profit increases. If we maintain

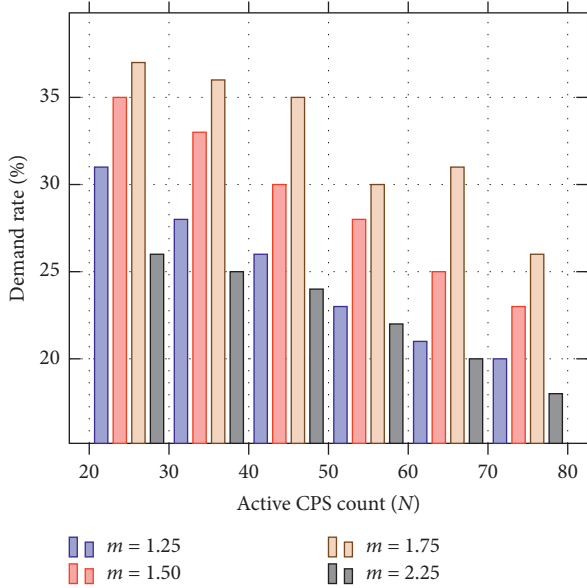


FIGURE 7: User service demand analysis over server infrastructure size with various service execution speeds.

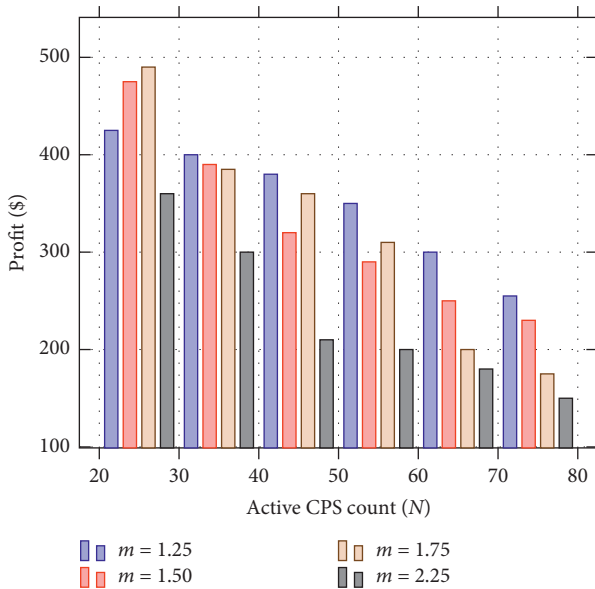


FIGURE 8: Profit analysis over server infrastructure size with various service execution speeds.

constant computing capacity, the server speed is impacted by the increase of active server count, which causes a decrease in the profit. Therefore, if the server size is peak and speed remains constant, it saves the energy cost and impacts CSP profit.

Figure 11 shows the comparative analyses of the server size and profit by regulating the server speed and USD rate. To assess the outcomes, we have used Table 1 listed parameters. If we increase the m value, then the active server size gets low due to m value increment under USD certainty. The profit gets impact when the energy cost is high and influences service execution speed to diminish CSP profit.

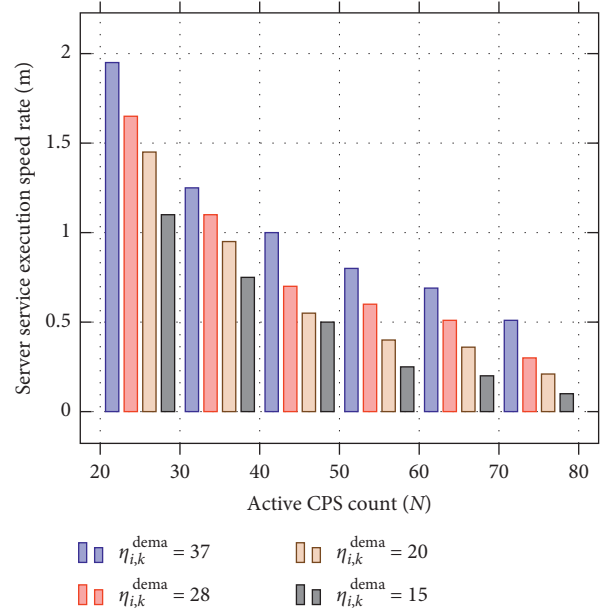


FIGURE 9: Server service execution speed rate over server infrastructure size with various user service demands ($\eta_{i,k}^{dema}$).

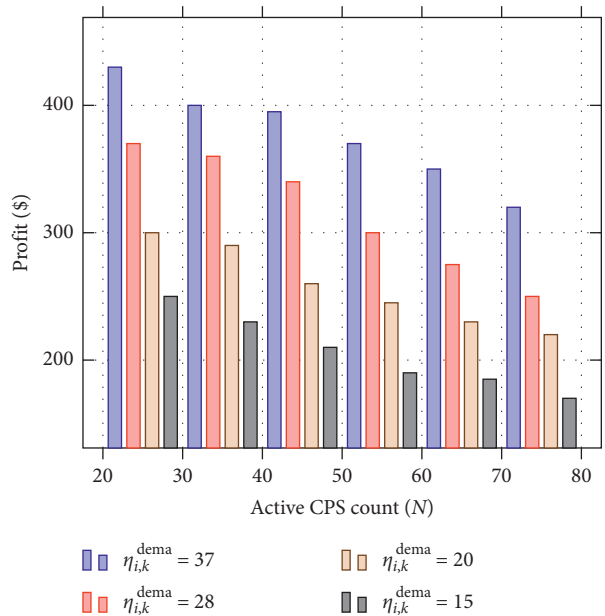


FIGURE 10: Profit analysis over server infrastructure size with various user service demands ($\eta_{i,k}^{dema}$).

Table 2 shows the comparative study analysis concerning all state-of-the-art approaches. The proposed system has outstanding profit, such as a 35.5% average. Subsequently, the profit is accomplished due to the data analysis model, and also performance rate of our system remains increased than existing approaches. The machine performance and execution cost measurement estimations played an essential role to gain adequate noticeable profit for CSP.

Table 3 illustrates our approach's simulation outcomes with the unit price 0.6\$ and average execution time 0.6 ms.

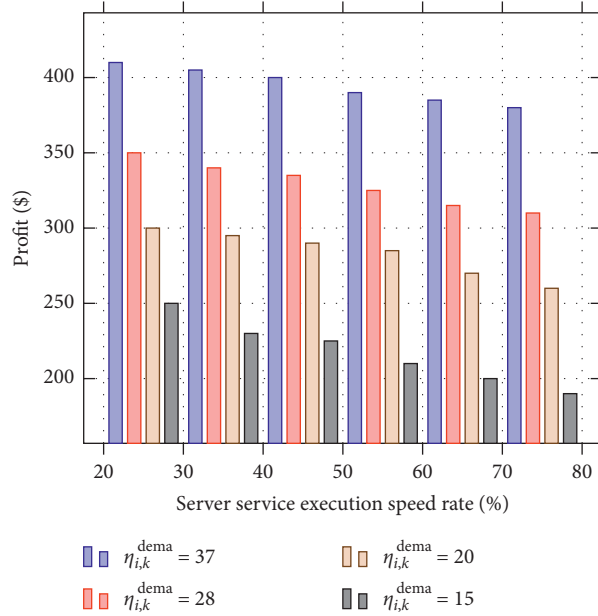


FIGURE 11: Profit analysis over server infrastructure size with various user service demands ($\eta_{i,k}^{dema}$).

TABLE 2: Profit comparative analysis based on server size and server speed.

Server size	DAWM		SPEA2		COMCPM		NSGA-II		OMCPM	
	SS	Profit	SS	Profit	SS	Profit	SS	Profit	SS	Profit
m2.6xlarge	4.5	590	3.9	520	3.5	480	3	440	2.8	530
m2.2xlarge	3.5	510	3.3	480	2.9	415	2.2	395	2.1	490
m1.2xmedi	2	440	2.2	315	2.5	335	1.8	360	1.9	450
m1.xsmall	1.5	370	1.9	255	1.75	290	1.7	300	1.5	320

Note: SS, server speed.

TABLE 3: Simulation outcome with the unit price 0.6\$ and average execution time 0.6 ms.

u_{max}	n_{opti}	ϕ_{opti}	Δ_{opti}	User cost (%)	Error (%)
50	110	7.59	251.32	48.912	2.15
60	118	7.21	310.68	48.245	1.91
70	125	7.84	372.98	48.329	1.88
80	167	7.99	415.25	49.786	1.52
90	182	7.23	490.89	49.791	1.49
100	195	7.51	525.15	51.012	1.32
110	229	7.58	590.69	40.452	1.31
120	250	7.32	610.15	45.697	1.28

The service price, service price-demand, maximum average service arrival rate, error rate, and user cost are assessed with average service execution time.

5. Conclusion

The proposed approach has been designed based on a belief propagation-influenced analytical data model to enhance CSP profit through DAG-based task and resource scheduling policy. It optimizes the CDC asset usage rate by consolidating overprovisioning machines. Cloud service suppliers drive the data utility analytic method on machines with low-resource usage rates to preserve CDC usage and

performance cost and avoid instant repudiations/migrations.

It initially recognizes feasible servers after the first iteration by concocting the data analytic weight measurement (DAWM) model. The DAWM model optimizes the cloud service provider's average cost by 51% due to considering each machine's cost and revenue during an effective resource slicing process. The multiobjective heuristic user service demand (MHUSD) algorithm accomplished average server performance by 41% and average CSP revenue gain by 35% due to CPS profit estimation model and the user service demand (USD) model with dynamic acyclic graph (DAG) phenomena by providing adequate service reliability. It

considers service demand weight, service tenant cost, and machine energy cost. Subsequently, the MHUSD algorithm also considers maximum bearing wait-time of end-user to maximize CSP revenue and optimize operational energy cost. Google cloud tracer confines the optimized average system profit by 590\$, and service execution speed is 4.5 sec/MIPS with the m2.6X large core system. The simulation results show that our system has an average service execution speed faster than the remaining approaches, such as 41.2%, 55.56%, 59.89%, and 61.52% faster than SPEA2, COMCPM, NSGA-II, and OMCPM, respectively. Subsequently, the proposed system achieved moderately high revenue by 10%, 8.1%, 8.9%, and 8.91% than SPEA2, COMCPM, NSGA-II, and OMCPM approaches and profit by 2.31%, 2.01%, 1.7%, and 1.37% than the state-of-the-art approaches.

Data Availability

No data were used to support the findings of this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Cisco, "An innovative business model for cloud providers," *Whitepaper*, vol. 2019, 2019, <https://www.cisco.com/c/dam/enus/solutions/trends/cloud/docs/aninnovativebusinesswhitepaper.pdf>.
- [2] J. Ru, Y. Yang, J. Grundy, J. Keung, and L. Hao, "A systematic review of scheduling approaches on multi-tenancy cloud platforms," *Information and Software Technology*, vol. 132, Article ID 106478, 2020.
- [3] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, "A comprehensive survey for scheduling techniques in cloud computing," *Journal of Network and Computer Applications*, vol. 143, pp. 1–33, 2019.
- [4] V. Seethalakshmi, V. Govindasamy, and V. Akila, "Hybrid gradient descent spider monkey optimization (hgdsmo) algorithm for efficient resource scheduling for big data processing in heterogenous environment," *Journal of Big Data*, vol. 7, no. 1, pp. 1–25, 2020.
- [5] E. Buggingo, D. Zhang, Z. Chen, and W. Zheng, "Towards decomposition based multi-objective workflow scheduling for big data processing in clouds," *Cluster Computing*, vol. 24, pp. 115–139, 2020.
- [6] Y. Wen, J. Liu, W. Dou, X. Xu, B. Cao, and J. Chen, "Scheduling workflows with privacy protection constraints for big data applications on cloud," *Future Generation Computer Systems*, vol. 108, pp. 1084–1091, 2020.
- [7] W. Ahmad, B. Alam, S. Ahuja, and S. Malik, "A Dynamic vm provisioning and de-provisioning based cost-efficient deadline-aware scheduling algorithm for big data workflow applications in a Cloud environment," *Cluster Computing*, vol. 24, pp. 249–278, 2020.
- [8] Y. Tang, Y. Yuan, and Y. Liu, "Cost-aware reliability task scheduling of automotive cyber-physical systems," *Microprocessors and Microsystems*, Article ID 103507, 2020.
- [9] S. Long, W. Long, Z. Li, K. Li, Y. Xia, and Z. Tang, "A game-based approach for cost-aware task assignment with qos constraints in large data centers," *IEEE Annals of the History of Computing*, vol. 32, pp. 1629–1640, 2020.
- [10] T. D. Nguyen, V.-N. Pham, L. N. Huynh, M. D. Hossain, E.-N. Huh et al., "Modeling data redundancy and cost-aware task allocation in mec-enabled internet-of-vehicles applications," *IEEE Internet of Things Journal*, vol. 8, no. 3, 2020.
- [11] J. Khamse-Ashari, I. Lambadaris, G. Kesidis, B. Urgaonkar, and Y. Zhao, "An efficient and fair multi-resource allocation mechanism for heterogeneous servers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 12, pp. 2686–2699, 2018.
- [12] K. Karthiban and J. S. Raj, "An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm," *Soft Computing*, vol. 24, no. 19, pp. 14933–14942, 2020.
- [13] W. Bai, J. Zhu, S.-W. Huang, and H. Zhang, "A queue waiting cost-aware control model for large scale heterogeneous cloud datacenter," *IEEE Transactions on Cloud Computing*, no. 1, p. 1, 2020.
- [14] P. Cong, "Personality-and value-aware scheduling of user requests in cloud for profit maximization," *IEEE Transactions on Cloud Computing*, vol. 99, p. 1, 2020.
- [15] K. K. Chakravarthi, L. Shyamala, and V. Vaidehi, "Budget aware scheduling algorithm for workflow applications in IaaS clouds," *Cluster Computing*, vol. 23, pp. 3405–3419, 2020.
- [16] C. Li, Y. Zhang, Z. Hao, and Y. Luo, "An effective scheduling strategy based on hypergraph partition in geographically distributed datacenters," *Computer Networks*, vol. 170, no. 4, Article ID 107096, 2020.
- [17] P. Cong, G. Xu, T. Wei, and K. Li, "A survey of profit optimization techniques for cloud providers," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–35, 2020.
- [18] J. Mei, K. Li, Z. Tong, Q. Li, and K. Li, "Profit maximization for cloud brokers in cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 1, pp. 190–203, 2019.
- [19] Y. Ma, W. Liang, Z. Xu, and S. Guo, "Profit maximization for admitting requests with network function services in distributed clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 5, pp. 1143–1157, 2019.
- [20] J. Wan, R. Zhang, X. Gui, and B. Xu, "Reactive pricing: an adaptive pricing policy for cloud providers to maximize profit," *IEEE Transactions on Network and Service Management*, vol. 13, no. 4, pp. 941–953, 2016.
- [21] P. Kaur and S. Mehta, "Resource provisioning and work flow scheduling in clouds using augmented shuffled frog leaping algorithm," *Journal of Parallel and Distributed Computing*, vol. 101, pp. 41–50, 2017.
- [22] H. Won, M. C. Nguyen, M.-S. Gil, and Y.-S. Moon, "Advanced resource management with access control for multitenant hadoop," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 592–601, 2015.
- [23] M. Tanaka and Y. Murakami, "Strategy-proof pricing for cloud service composition," *IEEE Transactions on Cloud Computing*, vol. 4, no. 3, pp. 363–375, 2014.
- [24] C. Liu, K. Li, K. Li, and R. Buyya, "A new cloud service mechanism for profit optimizations of a cloud provider and its users," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 14–26, 2017.
- [25] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 158–171, 2013.

- [26] R. Mohamadi Bahram Abadi, A. M. Rahmani, and S. H. Alizadeh, "Server consolidation techniques in virtualized data centers of cloud environments: a systematic literature review," *Software: Practice and Experience*, vol. 48, no. 9, pp. 1688–1726, 2018.
- [27] M. S. Mekala, A. Jolfaei, G. Srivastava, X. Zheng, A. Anvari-Moghaddam, and P. Viswanathan, "Resource offload consolidation based on deep-reinforcement learning approach in cyber-physical systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–10, 2020.
- [28] F. Farahnakian, A. Ashraf, T. Pahikkala et al., "Using ant colony system to consolidate vms for green cloud computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, 2014.
- [29] S. Singh and I. Chana, "A survey on resource scheduling in cloud computing: Issues and challenges," *Journal of Grid Computing*, vol. 14, no. 2, pp. 217–264, 2016.
- [30] C. Liu, K. Li, and K. Li, "A game approach to multi-servers load balancing with load-dependent server availability consideration," *IEEE Transactions on Cloud Computing*, vol. 9, pp. 1–13, 2018.
- [31] I. Mohiuddin and A. Almogren, "Workload aware vm consolidation method in edge/cloud computing for iot applications," *Journal of Parallel and Distributed Computing*, vol. 123, pp. 204–214, 2019.
- [32] J. Cao, K. Li, and I. Stojmenovic, "Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 45–58, 2013.
- [33] M. Guo, Q. Guan, and W. Ke, "Optimal scheduling of vms in queueing cloud computing systems with a heterogeneous workload," *IEEE Access*, vol. 6, pp. 15178–15191, 2018.
- [34] H. Chen, Y. Li, R. H. Y. Louie, and B. Vucetic, "Autonomous demand side management based on energy consumption scheduling and instantaneous load billing: an aggregative game approach," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1744–1754, 2014.
- [35] M. S. Mekala and V. Perumal, "Machine learning inspired phishing detection (pd) for efficient classification and secure storage distribution (ssd) for cloud-iot application," in *Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 202–210, Canberra, Australia, 2020.
- [36] J. Mei, K. Li, and K. Li, "Customer-satisfaction-aware optimal multiserver configuration for profit maximization in cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 1, pp. 17–29, 2017.
- [37] J. Cao, K. Hwang, K. Li, and A. Y. Zomaya, "Optimal multiserver configuration for profit maximization in cloud computing," *Ieee Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1087–1096, 2012.
- [38] N. Gholipour, E. Arianyan, and R. Buyya, "A novel energy-aware resource management technique using joint vm and container consolidation approach for green computing in cloud data centers," *Simulation Modelling Practice and Theory*, vol. 104, p. 102127, 2020.
- [39] Y.-L. Chou, J.-M. Yang, and C.-H. Wu, "An energy-aware scheduling algorithm under maximum power consumption constraints," *Journal of Manufacturing Systems*, vol. 57, pp. 182–197, 2020.
- [40] I. H. Sin and B. D. Chung, "Bi-objective optimization approach for energy aware scheduling considering electricity cost and preventive maintenance using genetic algorithm," *Journal of Cleaner Production*, vol. 244, Article ID 118869, 2020.
- [41] M. S. M. and P. Viswanathan, "Equilibrium transmission bi-level energy efficient node selection approach for internet of things," *Wireless Personal Communications*, vol. 108, no. 3, pp. 1635–1663, 2019.
- [42] M. S. M. and P. Viswanathan, "A survey: energy-efficient sensor and vm selection approaches in green computing for x-iot applications," *International Journal of Computers and Applications*, vol. 42, no. 3, pp. 290–305, 2020.
- [43] M. Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, "Energy-aware marine predators algorithm for task scheduling in iot-based fog computing applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5068–5076, 2020.
- [44] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, "A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm," *Journal of Parallel and Distributed Computing*, vol. 143, pp. 88–96, 2020.
- [45] A. S. Abohamama and E. Hamouda, "A hybrid energy-Aware virtual machine placement algorithm for cloud environments," *Expert Systems with Applications*, vol. 150, p. 113306, 2020.
- [46] X. Li, W. Yu, R. Ruiz, and J. Zhu, "Energy-aware cloud workflow applications scheduling with geo-distributed data," *IEEE Transactions on Services Computing*, vol. 9, p. 1, 2020.
- [47] H. Li, Y. Zhao, and S. Fang, "Csl-driven and energy-efficient resource scheduling in cloud data center," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 481–498, 2020.
- [48] S. Ghanavati, J. H. Abawajy, and D. Izadi, "An energy aware task scheduling model using ant-mating optimization in fog computing environment," *IEEE Transactions on Services Computing*, vol. 9, p. 1, 2020.
- [49] M. S. Mekala and P. Viswanathan, "Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for iot," *Computers & Electrical Engineering*, vol. 73, pp. 227–244, 2019.
- [50] T. Dong, F. Xue, C. Xiao, and J. Li, "Task scheduling based on deep reinforcement learning in a cloud manufacturing environment," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 11, Article ID e5654, 2020.

Research Article

Fully Constant-Size CP-ABE with Privacy-Preserving Outsourced Decryption for Lightweight Devices in Cloud-Assisted IoT

Zhishuo Zhang , Wei Zhang, and Zhiguang Qin 

School of Information and Software Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China

Correspondence should be addressed to Zhiguang Qin; qinzg@uestc.edu.cn

Received 1 December 2020; Revised 27 December 2020; Accepted 20 April 2021; Published 5 May 2021

Academic Editor: Chien-Ming Chen

Copyright © 2021 Zhishuo Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, ciphertext-policy attribute-based encryption (CP-ABE) has been recognized as a solution to the challenge of the information privacy and data confidentiality in cloud-assisted Internet-of-Things (IoT). Since the devices in cloud-assisted IoT are generally resource-constrained, the lightweight CP-ABE is more suitable for the cloud-assisted IoT. So how to construct the lightweight CP-ABE for the cloud-assisted IoT to achieve the fine-grained access control and ensure the privacy and confidentiality simultaneously is a prominent challenge. Thus, in this paper, we propose a constant-size CP-ABE scheme with outsourced decryption for the cloud-assisted IoT. In our scheme, the ciphertexts and the attribute-based private keys for users are both of constant size, which can alleviate the transmission overhead and reduce the occupied storage space. Our outsourced decryption algorithm is privacy-protective, which means the proxy server cannot know anything about the access policy of the ciphertext and the attributes set of the user during performing the online partial decryption algorithm. This will prevent the privacy from leaking out to the proxy server. And we rigorously prove that our scheme is selectively indistinguishably secure under the chosen ciphertext attacks (IND-CCA) in the random oracle model (ROM). Finally, by evaluating and implementing our scheme as well as other CP-ABE schemes, we can observe that our scheme is more suitable and applicable for cloud-assisted IoT.

1. Introduction

IoT has been recognized as a new paradigm in the network and information area in recent years [1, 2]. By means of the widespread deployment of spatially distributed devices, such as sensors, radio-frequency identification (RFID), wireless devices, and smartphones, IoT has the perfect sensing and actuation capabilities and makes the existing information system intelligent. Though IoT gives a new dimension to the Internet and has envisioned a future in which digital and physical entities can be linked in anywhere [3–5], security is still a critical obstacle for enabling the widespread adoption of the cloud-assisted IoT. To solve the security and privacy problem in IoT environment, many works design some authentication protocols [6], signature schemes [7] for Industrial Internet of Things (IIoT) [8, 9], Internet of Vehicles (IoV) [10, 11], and RFID networks [12]. But how to design a one-to-many and fine-grained access control encryption

mechanism for the cloud-assisted IoT is still being an open issue.

In cloud-assisted IoT, the data owners and the users all use the smart IoT devices. In traditional cloud-assisted IoT system, data owners transmit the data to the cloud server over the transmission media and the users download the data from the cloud storage. A hacker can easily access and steal the data in cleartext stored on the cloud storage. So, an encryption mechanism should be deployed in the cloud-assisted IoT architecture to ensure the data confidentiality and prevent the unauthorized access of the data [13, 14]. Figure 1 shows the comparison of the traditional cloud-assisted IoT system and the encryption mechanism-based cloud-assisted IoT system.

ABE [15] is a new cryptographic primitive widely researched in recent years which supports one-to-many encryption and refines the access control to the attribute level. So, ABE has been regarded as a powerful encryption mechanism for the cloud-assisted IoT. Particularly, CP-ABE [16–18], which is a type of

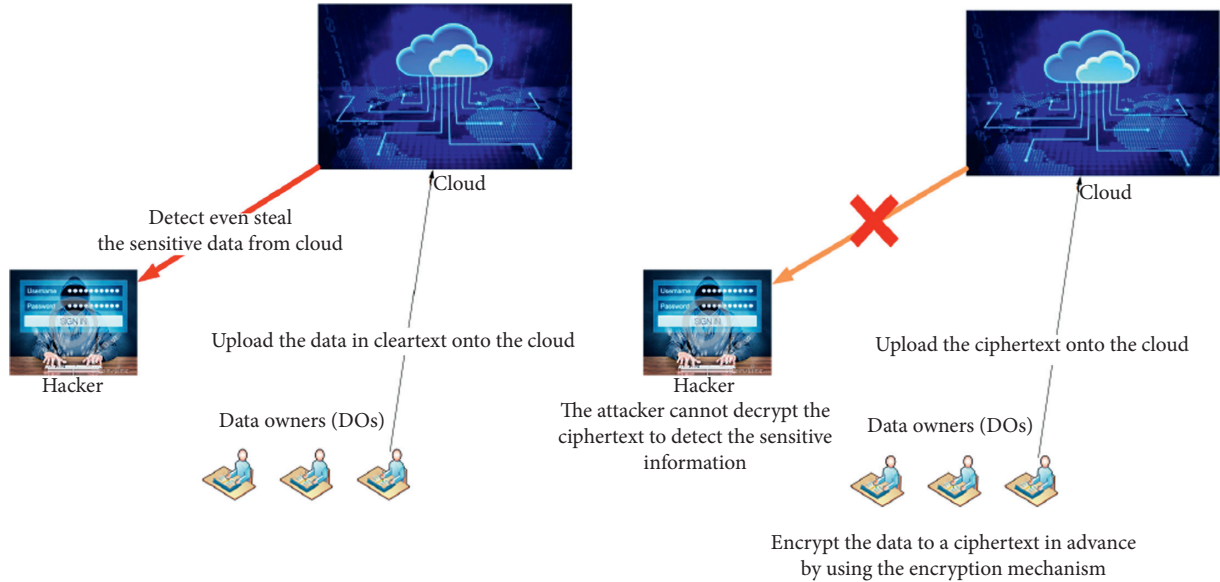


FIGURE 1: The comparison of the traditional cloud-assisted IoT system and the encryption mechanism-based cloud-assisted IoT system.

ABE, enables the data owner to customize an access formula over a set of attributes for each ciphertext and only if the user's attributes set meets the access policy, the user can decrypt the ciphertext. So, in CP-ABE, the data owner can precisely control the access to his/her data, and this makes CP-ABE a more applicable encryption tool for the cloud-based system. Nevertheless, in cloud-assisted IoT, the devices are generally resource-constrained (e.g., limited battery life, storage, and computing capability); the traditional CP-ABE is too complex to be fit-for-purpose. In typical CP-ABE, as [16–18], the ciphertext length grows linearly with the number of the attributes in the access policy and the size of the user's attribute-based private key also grows linearly with the size of the user's attributes set. Furthermore, as the access structure becomes more complex, the decryption time by the user will become longer, which not only increases the power consumption of the user's portable devices, but also makes the system less useful. To make CP-ABE applicable for the lightweight devices in the cloud-assisted IoT, in this paper, we propose a lightweight CP-ABE scheme with both constant-size ciphertexts and private keys. And we also invent a privacy-preserving outsourced decryption algorithm for the users to alleviate their computing burden. The privacy-preserving outsourced decryption algorithm can protect the privacy of the users and the data owners from divulging to the proxy server that means during performing the online partial decryption phase, the proxy server cannot know anything about the access policy associated with the ciphertext and the attributes set of the user. This will prevent the privacy from leaking out to the proxy server. To rigorously prove that our scheme is selectively IND-CCA secure in ROM, we reduce our scheme to n-aMSE-DDH problem [19–21].

1.1. Related Works. Lately, some researchers improve CP-ABE in two approaches to make the pure CP-ABE schemes applicable for the resource-constrained devices in IoT environment. One way is to construct the lightweight CP-ABE

to mitigate the transmission overhead of the system. And another way is outsourcing the decryption phase to proxy server to relieve the computing burden of the users used IoT devices.

1.1.1. Constant-Size CP-ABE. These works [20, 22] construct the constant-size ciphertext CP-ABE schemes which are using “Threshold policy” as their access structures. The scheme in [21] improves the work [20] to make a constant-size ciphertext CP-ABE scheme based on “Threshold policy” without dummy attributes. Emura et al. [23] build a fully constant-size CP-ABE scheme with both constant-size ciphertexts and private keys, but the access structure in their scheme [23] is using the less expressive “Strict AND-gate Policy.” And these works [24, 25] use [23] as their base construction also using the less expressive “Strict AND-gate Policy.” To make a trade-off between the expressiveness of the access structure and scale of the scheme, Yang et al. [26], Doshi and Jinwala [27], and Han et al. [28] use “AND-gate Policy with Wildcards” as their access structures to build the CP-ABE schemes with constant-size ciphertexts. To further lighten the CP-ABE schemes and reduce the transmission pressure, these schemes [19, 29] use “Tolerant AND-gate Policy based on Bits String” as their access structures which encoding an access structure to a bit string.

1.1.2. Outsourced Decryption. Green et al. firstly proposed a new cryptographic primitive of outsourced decryption CP-ABE in [30]. But in their schemes, a malicious proxy server could return a wrong transformed ciphertext to the user by disloyally running the outsourced transforming algorithm. Thus, their scheme [30] does not strictly guarantee the correctness of the transformed ciphertext sent to users. To solve this flaw, Lai et al. [31] add a verification function to

[30], but their scheme [31] adds some redundant components to the original ciphertext; this will make their ciphertext being twice length of the original ciphertext. To increase the efficiency of [31], Lin et al. [32], Qin et al. [33], and Mao et al. [34], respectively, designed a CP-ABE scheme with outsourced decryption and efficient decryption verification simultaneously. And all the schemes above [30–34] are based on [17]. Recently, Ning et al. [35] proposed an auditable σ time outsourced CP-ABE scheme based on [18], which can achieve higher security and can resist various types of attacks such as key-leakage attacks. And some schemes [36–38] with different properties combine with the outsourced decryption to make their schemes more suitable for IoT devices. But the users in all the above outsourced CP-ABE schemes will expose their attribute sets to the proxy server for running the semidecryption, which will lead to the disclosure of the privacy.

2. Preliminaries

2.1. AND-gate Access Structures

2.1.1. Strict AND-gate Policy. Let $N = \{\text{name}_1, \text{name}_2, \dots, \text{name}_n\}$ be the set of the attribute names. And $S_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,n_i}\}$ is the possible values set of the name name_i . $L = [l_1, l_2, \dots, l_n]$ is the attribute set of a user, where l_i is an element in S_i ($l_i \in S_i, 1 \leq i \leq n$). The $W = [w_1, w_2, \dots, w_n]$ is a strict AND-gate policy where w_i is an element in S_i ($w_i \in S_i, 1 \leq i \leq n$). Iff for all $i \in [1, n]$, $l_i = w_i$ holds, we call L satisfies the policy W . The scheme in [23] uses the “Strict AND-gate Policy” as its access structure.

2.1.2. AND-Gate Policy with Wildcards. Let $N = \{\text{name}_1, \text{name}_2, \dots, \text{name}_n\}$ be the set of the attribute names. And $S_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,n_i}\}$ is the possible values set of the name name_i . $L = [l_1, l_2, \dots, l_n]$ is the attribute set of a user where l_i is an element in S_i ($l_i \in S_i, 1 \leq i \leq n$). The $W = [w_1, w_2, \dots, w_n]$ is an AND-gate policy with wildcards where w_i is an element in S_i or the wildcard $*$ ($w_i \in \{S_i, *\}, 1 \leq i \leq n$). I_W is the set of indices i ($1 \leq i \leq n$) in which $w_i \neq *$; that is, $I_W = \{i | 1 \leq i \leq n, w_i \neq *\}$. Iff for all $i \in I_W$, $l_i = w_i$ holds, we call L satisfies the policy W . The schemes in [26, 27] use the “AND-gate Policy with Wildcards” as their access structure.

2.1.3. Tolerant AND-Gate Policy Based on Bits String. Let $U = \{\text{Attr}_1, \text{Attr}_2, \dots, \text{Attr}_n\}$ be the attribute universe. $L = l_1 l_2 \dots l_n$ is an n -bit string used to denote a user’s attribute set where $l_i \in \{0, 1\}$ ($1 \leq i \leq n$). If $l_i = 1$, it means that the user has the attribute Attr_i and if $l_i = 0$, it means that the user does not have the attribute Attr_i . And $W = w_1 w_2 \dots w_n$ ($w_i \in \{0, 1\}, 1 \leq i \leq n$) is the policy n -bit string. If $w_i = 1$, it means that the access policy W needs the attribute Attr_i and if $w_i = 0$, it means that the access policy W does not care about attribute Attr_i . I_W is the set of indices i ($1 \leq i \leq n$) in which $w_i = 1$; that is, $I_W = \{i | 1 \leq i \leq n, w_i = 1\}$. $|I_W|$ denotes the size of I_W . Iff for all $i \in I_W$, $l_i = w_i = 1$ holds, we call the attributes set L satisfies the access policy W . For instance, suppose $U = \{\text{Attr}_1, \text{Attr}_2, \text{Attr}_3, \text{Attr}_4, \text{Attr}_5\}$

and two attribute sets as $L_1 = 10011$ and $L_2 = 00111$. The access policy is $W = 10001$. So, we can observe that L_1 can satisfy W and L_2 cannot meet W . The schemes in [19, 39] use the “Tolerant AND-gate Policy based on Bits String” as their access structure.

Through the description of the three types of AND-gate access structures, we can observe that the “AND-gate Policy with Wildcards” and “Tolerant AND-gate Policy based on Bits String” are more flexible and expressive than the “Strict AND-gate Policy.” Furthermore, encoding an access structure to a bit string can compress the size of the access structure and which also can mitigate the communication burden. Our scheme uses the “Tolerant AND-gate Policy Based on Bits String” as the access structure.

2.2. Bilinear Pairings. G_1, G_2 are two elliptic groups and G_T is a multiplicative group. g_0 is a generator of G_1 and h_0 is a generator of G_2 . G_1, G_2, G_T are all with prime order p . $e: G_1 \times G_2 \longrightarrow G_T$ is called the bilinear pairing if

- (i) For any $g \in G_1, h \in G_2$ and $a, b \in \mathbb{Z}_p^*$, we have $e(g^a, h^b) = e(g, h)^{ab}$.
- (ii) If g_0 is a generator of G_1 and h_0 is a generator of G_2 , $e(g_0, h_0)$ is a generator of G_T .
- (iii) Group operations in G_1, G_2 and $e: G_1 \times G_2 \longrightarrow G_T$ are both efficiently computable. If G_1 and G_2 are the same group, that is, $G_1 = G_2 = G$, we call $e: G \times G \longrightarrow G_T$ the symmetric bilinear pairing.

And, the terms $BP = \{G_1, G_2, G_T, p, g_0, h_0, e\}$ are called the bilinear pairing terms.

2.3. n -aMSE-DDH Problem [19–21]. Let $BP = \{G_1, G_2, G_T, p, g_0, h_0, e\}$ be the bilinear pairing terms. Let $f(x)$ and $\theta(x)$ be two coprime polynomials in $\mathbb{Z}_p[x]$. Choose $a, \gamma \xleftarrow{R} \mathbb{Z}_p^*$ where “ \xleftarrow{R} ” means “randomly choose from.” Give T, \vec{p} to any probabilistic polynomial-time (PPT) adversary. Then, no adversary has the nonnegligible advantage to distinguish $T = e(g_0, h_0)^{\gamma f(a)}$ or $T = R$, where R is a random element in G_T . And $\vec{p} =$

$$\left\{ \begin{array}{l} \{g_0^{a^i}\}_{0 \leq i \leq n-1}, g_0^{af(a)}, g_0^{\gamma af(a)}; \\ \{h_0^{a^i}\}_{0 \leq i \leq n}, h_0^\gamma; \\ \{h_0^{(a^i/\theta(a))}\}_{0 \leq i \leq n} \end{array} \right. \quad (1)$$

3. Our Constant-Size CP-ABE Scheme with Privacy-Preserving Outsourced Decryption

3.1. System Architecture. The framework of our cloud-assisted IoT system used our scheme is shown in Figure 2. There are six entities involved in our system which are stated as follows.

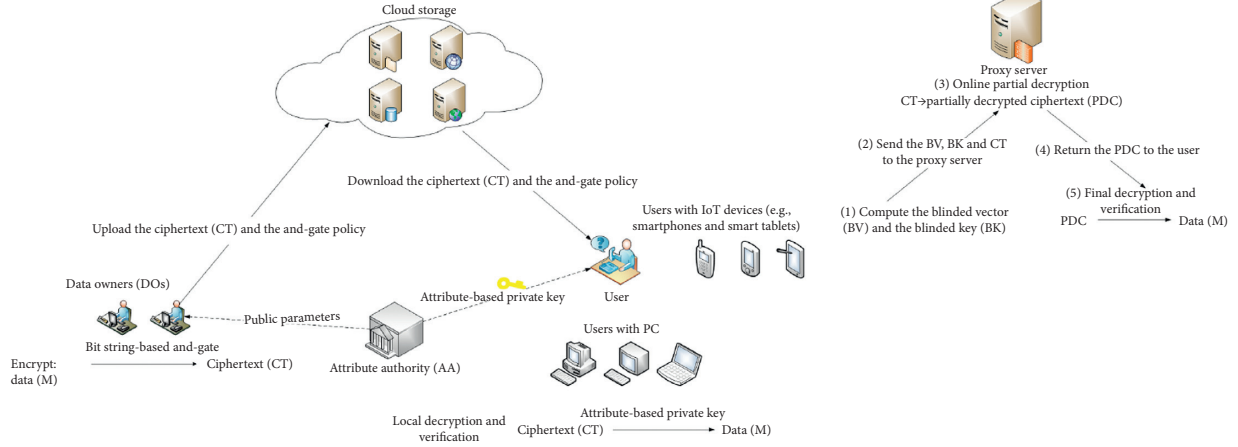


FIGURE 2: Our system framework.

3.1.1. *Attribute Authority (AA)*. AA is in charge of initializing the system and generating the private keys for users.

3.1.2. *Cloud Storage*. The cloud storage stores the ciphertexts for data owners (DOs).

3.1.3. *Data Owner (DOs)*. DOs encrypt the data to ciphertexts and upload the ciphertexts to the cloud.

3.1.4. *Users*. The users download the ciphertexts from the cloud storage then retrieve the plaintext by the decryption algorithm. The users have two types. One type is Users with PCs and the other is Users with smart IoT devices.

- (i) Users with PCs: users with PCs retrieve the plaintext by running the local-decryption phase
- (ii) Users with IoT devices: the users with smartphones or the smart tablets can retrieve the data by performing the privacy-preserving outsourced decryption phase

3.1.5. *Proxy Server*. Proxy servers take charge of running the online partial decryption algorithm for the users with smart IoT devices. Note that the proxy servers cannot know anything about the user's attributes and the access policy associated with the ciphertext during running the partial decryption.

3.2. *Algorithm Definitions*. The workflow of our cloud-assisted IoT system used in our scheme is shown in Figure 3. There are four algorithms in our scheme described as below.

3.2.1. *Setup*. AA initializes the system by executing the Setup algorithm to export the public parameters PK and master private key MK of the system. AA preserves the private master key privately and publishes the public parameters to all the entities in the system.

3.2.2. *AttrKeyGen*. A user forms his attribute set as a bit string then sends his/her bit string-based attribute set to the AA; AA runs the AttrKeyGen algorithm to generate the constant-size attribute-based private key for the user. Then, the user will preserve the attribute-based key privately. If the user's attribute set can meet the access policy associated with the ciphertext, he/she can use his/her private key to decrypt the ciphertext.

3.2.3. *Encrypt*. A DO customizes a bit string formed attribute-based access policy for the data; then, by the Encrypt algorithm, the DO encrypts the data under the customized access policy to a ciphertext, which is constant size. Then, the DO uploads the ciphertext with the bit string formed access policy onto the cloud storage.

3.2.4. *Decrypt*. A user downloads the ciphertext with the access policy from the cloud storage. If the user's attribute set meets the access policy, then he/she can retrieve the data by running the Decrypt algorithm. And, the Decrypt algorithm has two modes. One mode is local decryption. The local decryption means all the computations are running on the user's local device, and this mode is suitable for the users with PCs. If the user is using the smart IoT devices, then the user can choose the other decryption mode called privacy-preserving outsourced decryption to securely and privately outsource some complex computations to the proxy server. This will reduce the decryption time of the user and save the

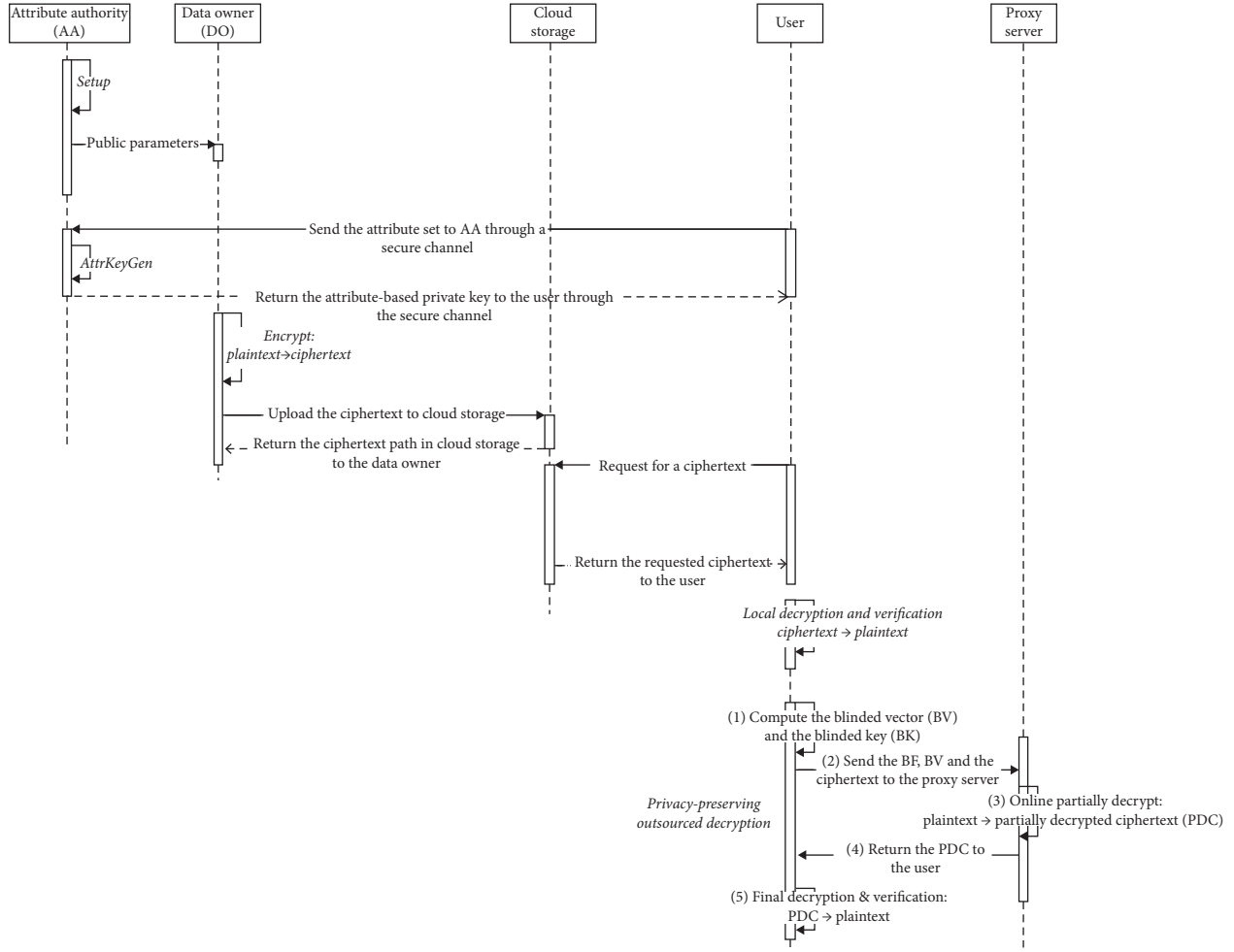


FIGURE 3: Our system framework.

battery power of the user's smart IoT device. Note that the proxy servers cannot know anything about the user's attributes and the access policy associated with the ciphertext during partially decrypting the ciphertext.

3.3. Security Model. We define a selectively IND-CCA security game for our scheme which involves an adversary algorithm \mathcal{A} and a challenge algorithm \mathcal{C} in.

- (i) *Initialization.* \mathcal{A} sends a bit string based AND-gate challenge access structure W^* to \mathcal{C} .
- (ii) *Setup.* \mathcal{C} runs setup algorithm to generate the master private key MK and public parameters PK . Then, \mathcal{C} sends PK to \mathcal{A} .
- (iii) *Key Query 1.* \mathcal{A} queries a list of bit strings to \mathcal{C} for the key queries. Note that all the key queries cannot satisfy the challenge access structure W^* .
- (iv) *Decryption Query 1.* \mathcal{A} queries the decryption of ciphertext $\text{Encrypt}(W_i, M_i)$ from \mathcal{C} .

(v) *Challenge.* \mathcal{A} sends two messages $M_0 \in \{0, 1\}^l$ and $M_1 \in \{0, 1\}^l$ to \mathcal{C} for challenge. $\mathcal{C} \xleftarrow{R} \{0, 1\}$ and sends $\{W^*, \text{Encrypt}(W^*, M)_b\}$ back to \mathcal{A} .

(vi) *Key Query 2.* Same as Key Query 1. Notice that all key queries in this phase also cannot satisfy the access structure W^* .

(vii) *Decryption Query 2.* Same as Decryption Query 1. And notice that the decryption queries cannot be the challenge messages M_0 and M_1 .

(viii) *Guess.* \mathcal{A} outputs a guess $b'b$.

$\text{Adv}_{\mathcal{A}} = \Pr[b' = b] - (1/2)$. \mathcal{A} wins the confidentiality game if $\text{Adv}_{\mathcal{A}}$ is nonnegligible.

3.4. Scheme Construction

3.4.1. Setup. AA performs the Setup phase to initialize the system by the following steps.

- (i) AA exports a bilinear pairing $BP = \{G_1, G_2, G_T, p, g, h, e\}$ from the security parameter κ . g is a generator of G_1 and h is a generator of G_2 . Then, AA chooses four one-way collision-resistance hash function as $H_1, H': \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$, $H_2: G_T \rightarrow \{0, 1\}^{l_\beta}$, $H_3: \{0, 1\}^{l_\beta} \rightarrow \{0, 1\}^{l_m}$.
- (ii) AA defines the attribute universe $U = \{\text{Attr}_1, \text{Attr}_2, \dots, \text{Attr}_n\}$ of the system, $n = |U|$. Then, AA chooses $\{a, k_1, k_2\} \xleftarrow{R} \mathbb{Z}_p^*$ and computes $\{h_i = h^{a^i}, u_i = h^{k_1 a^i}, v_i = h^{k_2 a^i}\}_{i=0}^n$.
- (iii) Finally, AA preserves the master private key (MK) and publishes the public parameters (PK) as
- $$MK = g, a, k_1, k_2;$$
- $$PK = U, \{h_i, u_i, v_i\}_{i=0}^n, H_1, H_2, H_3, H', e(g, h), g^a. \quad (2)$$

3.4.2. AttrKeyGen. A user forms his attribute set as a bit string $L = l_1 l_2, \dots, l_n$ where $l_i \in \{0, 1\}$ ($1 \leq i \leq n$) and then sends L to the AA via a secure channel. Then, AA generates the attribute-based private key for the user by the following steps.

- (i) AA generates an n -degree at most polynomial function $f(x, L) = \prod_{i=1}^n (x + H'(\text{Attr}_i))^{1-l_i}$ in $\mathbb{Z}_p[x]$ by using the bit string L . Then, AA computes $f(a, L) = \prod_{i=1}^n (a + H'(\text{Attr}_i))^{1-l_i}$.
- (ii) AA $r_u \xleftarrow{R} \mathbb{Z}_p^*$ and computes s_u with the condition $(1/f(a, L)) = k_1 s_u + k_2 r_u$, that is, $s_u = (((1/f(a, L)) - k_2 r_u)/k_1)$.
- (iii) Finally, AA computes the attribute-based private key $K_u = \{K_{u,1} = g^{r_u}, K_{u,2} = g^{s_u}\}$ for the user and sends K_u to the user via a secure channel.

3.4.3. Encrypt. DO performs the following steps to encrypt the data $M \in \{0, 1\}^{l_m}$.

- (i) DO customizes an AND-gate access structure based on bit string as $W = w_1 w_2, \dots, w_n$ where $w_i \in \{0, 1\}$ ($1 \leq i \leq n$) for the data $M \in \{0, 1\}^{l_m}$. I_w is the set of indices i ($1 \leq i \leq n$) in which $w_i = 1$, that is, $I_w = \{i | 1 \leq i \leq n, w_i = 1\}$. And $|I_w|$ denotes the size of

I_w . Notice that $|I_w| > 0$. Then, DO generates a $(n - |I_w|)$ -degree polynomial $f(x, W) = \prod_{i=1}^{n-|I_w|} (x + H'(\text{Attr}_i))^{1-w_i}$ in $\mathbb{Z}_p[x]$ by using the access bit string W . Let $f_{i,W}$ be the coefficient of x^i in $f(x, W)$.

- (ii) DO $\beta_m \xleftarrow{R} \{0, 1\}^{l_\beta}$ and computes $r_m = H_1(W, M, \beta_m)$.
- (iii) Finally, DO computes the ciphertext for the data $M \in \{0, 1\}^{l_m}$ as $CT = \{C_1, C_2, C_3, C_4, C_5\}$ and then sends $\{CT, W\}$ to the cloud storage.

$$C_1 = (g^a)^{r_m};$$

$$C_2 = \left(\prod_{i=0}^{n-|I_w|} (u_i)^{f_{i,W}} \right)^{r_m} = h^{r_m k_1 \sum_{i=0}^{n-|I_w|} a^i f_{i,W}} = h^{r_m k_1 f(a, W)};$$

$$C_3 = \left(\prod_{i=0}^{n-|I_w|} (v_i)^{f_{i,W}} \right)^{r_m} = h^{r_m k_2 \sum_{i=0}^{n-|I_w|} a^i f_{i,W}} = h^{r_m k_2 f(a, W)};$$

$$C_4 = H_2(e(g, h)^{r_m}) \oplus \beta_m;$$

$$C_5 = H_3(\beta_m) \oplus M. \quad (3)$$

3.4.4. Decryption. The user downloads the ciphertext $\{CT, W\}$ from the cloud storage. If the user's attributes set L can meet the access policy W associated with the ciphertext, the user can decrypt the ciphertext in two ways. One way is the local decryption and another is the privacy-preserving outsourced decryption. If the user uses the PC, he/she can use the local decryption algorithm to obtain the data. Or if the user uses the IoT device, such as smartphone, he/she can use the privacy-preserving outsourced decryption to obtain the data without the computing pressure. Notice that if and only if L meets W , the user can generate a $(n - |I_w|)$ -degree at most polynomial $F(x, L, W) = (f(x, W)/f(x, L)) = \prod_{i=1}^n (x + H'(\text{Attr}_i))^{l_i - w_i} = \sum_{i=0}^{n-|I_w|} x^i F_{i,L,W}$ in $\mathbb{Z}_p[x]$ where $F_{i,L,W}$ is the coefficient of x^i and it is clear that $F_{0,L,W} \neq 0$. $\vec{F} = (F_{0,L,W}, F_{1,L,W}, \dots, F_{n-|I_w|,L,W})$ is the coefficient vector of $F(x, L, W)$ in \mathbb{Z}_p^* .

Local decryption: the user runs the local decryption by the following steps:

$$\begin{aligned}
W &= e\left(C_1, \prod_{i=1}^{n-|I_W|} (h_{i-1})^{F_{i,L,W}}\right) = e\left(g^{ar_m}, \prod_{i=1}^{n-|I_W|} h^{a^{i-1}F_{i,L,W}}\right) = e(g, h)^{ar_m \sum_{i=1}^{n-|I_W|} a^{i-1}F_{i,L,W}} = e(g, h)^{r_m \sum_{i=1}^{n-|I_W|} a^i F_{i,L,W}} \\
&= e(g, h)^{r_m \left(\sum_{i=1}^{n-|I_W|} a^i F_{i,L,W} + F_{0,L,W} - F_{0,L,W}\right)} = e(g, h)^{r_m F(a,L,W) - r_m F_{0,L,W}}, \\
U &= e(K_{u,2}, C_2) = e(g, h)^{k_1 f(a,W) r_m s_u}, \\
V &= e(K_{u,1}, C_3) = e(g, h)^{k_2 f(a,W) r_m r_u}, \\
UV &= e(g, h)^{k_1 f(a,W) r_m s_u + k_2 f(a,W) r_m r_u} = e(g, h)^{r_m f(a,W) (k_1 s_u + k_2 r_u)} = e(g, h)^{r_m (f(a,W)/f(a,L))} \\
&= e(g, h)^{r_m F(a,L,W)}, \\
\left(\frac{UV}{W}\right)^{(1/F_{0,L,W})} &= \left(\frac{e(g, h)^{r_m F(a,L,W)}}{e(g, h)^{r_m F(a,L,W) - r_m F_{0,L,W}}}\right)^{(1/F_{0,L,W})} = e(g, h)^{r_m} = V_1, \\
\beta_{m'} &= C_4 \oplus H_2(V_1), \\
M' &= C_5 \oplus H_3(\beta_{m'}).
\end{aligned} \tag{4}$$

Then, the user computes $r_{m'} = H_1(W, M', \beta_{m'})$ and verifies $V_1 \stackrel{?}{=} e(g, h)^{r_{m'}}$. If the equation holds, this indicates the user decrypts the ciphertext successfully ($M' = M$).

Privacy-preserving outsourced decryption: the user $\{u_1, u_2 \xleftarrow{R} \mathbb{Z}_p^*\}$ and computes the blinded coefficient vector \vec{BV} in \mathbb{Z}_p^* and the blinded private key BK as

$$\begin{aligned}
\vec{BV} &= \left(0, u_1 F_{1,L,W}, \dots, u_1 F_{n-|I_W|,L,W}\right) \bmod p; \\
BK &= \{BK_{u,1} = K_{u,1}^{u_2}, BK_{u,2} = K_{u,2}^{u_2}\}.
\end{aligned} \tag{5}$$

Then, the user sends $\{C_1, C_2, C_3, \vec{BV}, BK\}$ to the proxy server. It is clear that the proxy server only cannot know anything about L and W from the blind coefficient vector \vec{BV} and the blind private key BK . The proxy server uses $\{C_1, C_2, C_3, \vec{BV}, BK\}$ to compute

$$\begin{aligned}
P_1 &= e\left(C_1, \prod_{i=1}^{n-|I_W|} (h_{i-1})^{u_1 F_{i,L,W}}\right) = e\left(g^{ar_m}, \prod_{i=1}^{n-|I_W|} h^{a^{i-1} u_1 F_{i,L,W}}\right) = e(g, h)^{ar_m \sum_{i=1}^{n-|I_W|} a^{i-1} u_1 F_{i,L,W}} \\
&= e(g, h)^{u_1 r_m \sum_{i=1}^{n-|I_W|} a^i F_{i,L,W}} = e(g, h)^{u_1 r_m \left(\sum_{i=1}^{n-|I_W|} a^i F_{i,L,W} + F_{0,L,W} - F_{0,L,W}\right)} = e(g, h)^{u_1 (r_m F(a,L,W) - r_m F_{0,L,W})}, \\
U' &= e(BK_{u,2}, C_2) = e(g, h)^{u_2 k_1 f(a,W) r_m s_u}, \\
V' &= e(BK_{u,1}, C_3) = e(g, h)^{u_2 k_2 f(a,W) r_m r_u}, \\
P_2 &= U' V' = e(g, h)^{u_2 (k_1 f(a,W) r_m s_u + k_2 f(a,W) r_m r_u)} = e(g, h)^{u_2 (r_m f(a,W) (k_1 s_u + k_2 r_u))} \\
&= e(g, h)^{u_2 r_m F(a,L,W)}.
\end{aligned} \tag{6}$$

Then, proxy server sends $\{P_1, P_2\}$ back to the user. The user uses $\{C_4, C_5, P_1, P_2, u_1, u_2\}$ to compute

$$\begin{aligned}
W &= P_1^{1/u_1} = e(g, h)^{r_m F(a, L, W) - r_m F_{0, L, W}}; J = P_2^{1/u_2} = e(g, h)^{r_m F(a, L, W)}; \\
\left(\frac{J}{W}\right)^{(1/F_{0, L, W})} &= \left(\frac{e(g, h)^{r_m F(a, L, W)}}{e(g, h)^{r_m F(a, L, W) - r_m F_{0, L, W}}}\right)^{(1/F_{0, L, W})} \\
&= e(g, h)^{r_m} = V_1, \\
\beta_{m'} &= C_4 \oplus H_2(V_1), \\
M' &= C_5 \oplus H_3(\beta_{m'}).
\end{aligned} \tag{7}$$

Then, the user computes $r_{m'} = H_1(W, M', \beta_{m'})$ and verifies $V_1 \stackrel{?}{=} e(g, h)^{r_{m'}}$. If the equation holds, this indicates the user decrypts the ciphertext successfully ($M' = M$).

3.5. Security Analysis

Theorem. If the n-aMSE-DDH problem holds, then our scheme is selectively IND-CCA-secure.

Proof. Suppose there is a PPT adversary \mathcal{A} who can break the security of our scheme with a nonnegligible advantage $\text{Adv}_{\mathcal{A}}$. Then, we can construct a PPT simulator algorithm \mathcal{C} which is able to solve the n-aMSE-DDH problem with the non-negligible advantage ($\text{Adv}_{\mathcal{A}} - (q_{H_2}/p)$) by interacting with \mathcal{A} in the following manner where p is the order of group G_T and q_{H_2} is the number of the queries to the oracle H_2 .

3.5.1. Initialization. Note that there are n attributes $U = \{\text{Attr}_1, \text{Attr}_2, \dots, \text{Attr}_n\}$ in the scheme. \mathcal{A} submits the challenge access bit string $W^* = w_1^* w_2^* \dots w_n^*$ where $w_i^* \in \{0, 1\}$ ($1 \leq i \leq n$) to \mathcal{C} . $I_{W^*} = \{i \mid 1 \leq i \leq n, w_i^* = 1\}$. $|I_{W^*}|$ is the size of I_{W^*} . $\mathcal{C} \left\{v_1, v_2, \dots, v_n\right\} \leftarrow \mathbb{Z}_p^*$ and sets

$$\begin{aligned}
\theta(x) &= f(x, W^*) = \prod_{i=1}^n (x + v_i)^{1-w_i^*}; \\
f(x) &= \prod_{i=1}^n (x + v_i)^{w_i^*},
\end{aligned} \tag{8}$$

$\theta(x)$ is a $(n - |I_{W^*}|)$ -degree polynomial in $\mathbb{Z}_p[x]$ and $f(x)$ is a (n) -degree polynomial in $\mathbb{Z}_p[x]$.

\mathcal{C} sends $\theta(x)$ and $f(x)$ to the n-aMSE-DHH problem and receives the problem instances $\{\vec{p}, T\}$ from n-aMSE-DHH problem. T is the challenge term and $T = e(g_0, h_0)^{y f(a)}$ or $T = R$ where R is a random element in G_T . And $\vec{p} =$

$$\begin{aligned}
&\left\{g_0^{a^i}\right\}_{0 \leq i \leq n-1}, g_0^{a f(a)}, g_0^{y a f(a)}; \\
&\left\{h_0^{a^i}\right\}_{0 \leq i \leq n}, h_0^y; \\
&\left\{h_0^{a^i / (\theta(a))}\right\}_{0 \leq i \leq n},
\end{aligned} \tag{9}$$

where g_0 is a generator of G_1 and h_0 is a generator of G_2 and $e: G_1 \times G_2 \rightarrow G_T \text{ mod } p$.

3.5.2. Setup. $\mathcal{C} \left\{w_1, w_2\right\} \leftarrow \mathbb{Z}_p^*$ and implicitly sets master private key MK as

$$g = g_0^{f(a)}, a = a, k_1 = \frac{w_1}{\theta(a)}, k_2 = \frac{w_2}{\theta(a)}. \tag{10}$$

The public parameters PK are computed as

$$\begin{aligned}
h &= h_0, e(g, h) = e(g_0, h_0)^{f(a)}, g^a = g_0^{a f(a)}; \\
\left\{h_i = h_0^{a^i}, u_i = \left(h_0^{a^i / (\theta(a))}\right)^{w_1}, v_i = \left(h_0^{a^i / (\theta(a))}\right)^{w_2}\right\}_{i=1}^n.
\end{aligned} \tag{11}$$

Finally, \mathcal{C} sends PK to \mathcal{A} .

3.5.3. Hash Queries. \mathcal{A} can access the hash oracles (H', H_1, H_2, H_3) , and \mathcal{C} maintains the hash lists $\{\mathcal{L}_{H'}, \mathcal{L}_{H_1}, \mathcal{L}_{H_2}, \mathcal{L}_{H_3}\}$ to record the queries and responses, respectively. If the query has a previous response and the output result recorded in the hash lists, \mathcal{C} will respond with the recorded result in the hash lists. Otherwise, \mathcal{C} will perform as follows.

- (i) H' Oracle. Let the input of H' be in $U = \{\text{Attr}_1, \text{Attr}_2, \dots, \text{Attr}_n\}$. If the input of H' is Attr_i , \mathcal{C} sets v_i as the output. And the term $\{\text{Attr}_i: v_i\}$ will be recorded in $\mathcal{L}_{H'}$.
- (ii) H_1 Oracle. Let the input of H_1 be (W_i, M_i, β_i) . \mathcal{C} responds $H_1(W_i, M_i, \beta_i)$ with a random $r_i \in \mathbb{Z}_p^*$. And the term $\{(W_i, M_i, \beta_i): r_i\}$ will be recorded in \mathcal{L}_{H_1} .
- (iii) H_2 Oracle. Let the query to H_2 be $e(g, h)^{r_i}$. \mathcal{C} responds $H_2(e(g, h)^{r_i})$ with a random $R_i \in \{0, 1\}^{l_\beta}$. And the term $\{e(g, h)^{r_i}: R_i\}$ will be recorded in \mathcal{L}_{H_2} .
- (iv) H_3 Oracle. Let the query to H_3 be β_i . \mathcal{C} responds $H_3(\beta_i)$ with a random $Q_i \in \{0, 1\}^{l_m}$. And the term $\{\beta_i: Q_i\}$ will be recorded in \mathcal{L}_{H_3} .

3.5.4. Key Query 1. \mathcal{A} sends an attribute bit string $L = l_1 l_2 \dots l_n$ where $l_i \in \{0, 1\}$ ($1 \leq i \leq n$) to \mathcal{C} for one key query. Note that L cannot meet the challenge policy W^* . \mathcal{C} sets

$$f(x, L) = \prod_{i=1}^n (x + H'(\text{Attr}_i))^{1-l_i} = f_f(x, L) f_\theta(x, L), \tag{12}$$

$f_f(x, L)$ are the terms in $f(x)$ and $f_\theta(x, L)$ are the terms in $\theta(x)$. $f(x, L)$ can be computed by the part of terms in $f(x)$

and $\theta(x)$. And if L does not fulfill the challenge access structure W^* , the degree of the polynomial $f_f(x, L)$ is nonzero.

\mathcal{C} $r \xleftarrow{R} \mathbb{Z}_p^*$ and implicitly sets $r_u = (k_1 r a / k_2)$ by computing

$$g^{r_u} = (g_0^{af(a)})^{(w_1 r / w_2)}. \quad (13)$$

Implicitly set

$$\frac{f(x)\theta(x)}{w_1 f(x, L)} = \frac{1}{w_1} \frac{\prod_{i=1}^n (x + H'(\text{Attr}_i))}{\prod_{i=1}^n (x + H'(\text{Attr}_i))^{1-l_i}} = \frac{1}{w_1} \prod_{i=1}^n (x + H'(\text{Attr}_i))^{l_i} = \widehat{f}(x, L). \quad (16)$$

Let $\widehat{f}_{i,L}$ be the coefficient of x^i in $\widehat{f}(x, L)$. $\widehat{f}(x, L)$ is a $(n-1)$ -degree at most polynomial in $\mathbb{Z}_p[x]$.

$g_0^{\widehat{f}(a,L)}$ can be computed by the terms in \vec{p} as

$$g_0^{\widehat{f}(a,L)} = g_0^{(1/w_1) \sum_{i=0}^{n-1} \widehat{f}_{i,L} a^i} = \left(\prod_{i=0}^{n-1} (g_0^{a^i})^{\widehat{f}_{i,L}} \right)^{(1/w_1)}. \quad (17)$$

So, g^{s_u} can be computed as

$$g^{s_u} = g_0^{\widehat{f}(a,L)} (g_0^{af(a)})^{-r}. \quad (18)$$

Finally, \mathcal{C} sends $K_u = \{K_{u,1} = g^{r_u}, K_{u,2} = g^{s_u}\}$ to \mathcal{A} .

3.5.5. Decryption Query 1. For any decryption query on $\text{Encrypt}(W_i, M_i)$, if there exists $(W_i, M_i, \beta_i, r_i, R_i, Q_i)$ in the hash lists $\{\mathcal{L}_{H'}, \mathcal{L}_{H_1}, \mathcal{L}_{H_2}, \mathcal{L}_{H_3}\}$ such that the ciphertext is generated using r_i , \mathcal{C} sets M_i as the output of the decryption query to \mathcal{A} . Otherwise, \mathcal{C} outputs *null*. No query will be aborted since all valid encryptions need the response from hash oracles $\{H', H_1, H_2, H_3\}$, and the response contains the random number r_i which is used in encryption.

3.5.6. Challenge. \mathcal{A} sends two messages $M_0 \in \{0, 1\}^l$ and $M_1 \in \{0, 1\}^l$ to \mathcal{C} for challenge. \mathcal{C} implicitly defines $r_m = \gamma$ by setting

$$\begin{aligned} C_1 &= g_0^{af(a)\gamma}; \\ C_2 &= (h_0^\gamma)^{w_1}; \\ C_3 &= (h_0^\gamma)^{w_2}. \end{aligned} \quad (19)$$

$$s_u = \frac{1}{k_1} \left(\frac{1}{f(a, L)} - k_2 r_u \right) = \frac{\theta(a)}{w_1 f(a, L)} - r a. \quad (14)$$

\mathcal{C} computes g^{s_u} as

$$g^{s_u} = g^{(\theta(a)/w_1 f(a, L)) - r a} = g_0^{(f(a)\theta(a)/w_1 f(a, L))} (g_0^{af(a)})^{-r}. \quad (15)$$

We denote

Then, \mathcal{C} randomly chooses $\beta_m \in \{0, 1\}^{l_\beta}$, $b \in \{0, 1\}$ and computes

$$\begin{aligned} C_4 &= H_2(T) \oplus \beta_m; \\ C_5 &= H_3(\beta_m) \oplus M_b. \end{aligned} \quad (20)$$

Finally, \mathcal{C} sends $\{W^*, C_1, C_2, C_3, C_4, C_5\}$ to \mathcal{A} .

3.5.7. Key Query 2. It is the same as Key Query 1. Notice that all key queries in this phase also cannot satisfy the access structure W^* .

3.5.8. Decryption Query 2. It is the same as Decryption Query 1. And notice that the decryption queries cannot be the challenge messages M_0 and M_1 .

3.5.9. Guess. Eventually, \mathcal{A} gives the guess b' of b to the simulator \mathcal{C} .

If $b' = b$, the simulator \mathcal{C} outputs 0 and guesses $T = e(g_0, h_0)^{\gamma f(a)}$; otherwise, \mathcal{C} outputs 1 and guesses $T = R$.

If the n-aMSE-DHH problem sends $T = e(g_0, h_0)^{\gamma f(a)}$ to the simulator \mathcal{C} . The attacker \mathcal{A} plays the real security game as our actual scheme. Referring to our supposition, the attacker has $\text{Adv}_{\mathcal{A}}$ selectively breaking our actual scheme. So,

$$\Pr[b' = b \mid T = e(g_0, h_0)^{\gamma f(a)}] = \frac{1}{2} + \text{Adv}_{\mathcal{A}}; \Pr[\mathcal{C}(\vec{p}, T = e(g_0, h_0)^{\gamma f(a)}) = 0] = \frac{1}{2} + \text{Adv}_{\mathcal{A}}. \quad (21)$$

If the n-aMSE-DHH problem sends $T = R$ to \mathcal{C} , all the bits in M_b are hidden due to R . So,

$$\Pr[b' = b \mid T = R] = \frac{1}{2}. \quad (22)$$

TABLE 1: Properties comparison.

Scheme	Access structure	Constant-size ciphertext	Constant-size private key	Outsourced decryption	Security	Bilinear group
[23]	Strict AND-gate Policy	√	√	×	Selectively IND-CPA secure	Prime order
[20]	Threshold	√	×	×	Selectively IND-CPA secure	Prime order
[21]	Threshold	√	×	×	Selectively IND-CPA secure	Prime order
[16]	Access tree	×	×	×	Selectively IND-CPA secure	Prime order
[17]	Linear secret sharing schemes (LSSS) [40]	×	×	×	Selectively IND-CPA-secure	Prime order
[18]	LSSS	×	×	×	Selectively IND-CPA secure	Prime order
[41]	LSSS	×	×	×	Fully (adaptively) IND-CPA secure	Composite order
[26]	AND-gate policy with wildcards	√	×	×	Selectively IND-CPA secure	Prime order
[27]	AND-gate policy with wildcards	√	×	×	Fully (adaptively) IND-CPA secure	Composite order
[19]	Tolerant AND-gate Policy based on bits string	×	√	×	Selectively IND-CCA secure	Prime order
[31]	LSSS	×	×	√(not privacy-preserving)	Selectively IND-CPA secure	Prime order
[34]	LSSS	×	×	√(not privacy-preserving)	Selectively IND-CPA secure	Prime order
[33]	LSSS	×	×	√(not privacy-preserving)	Selectively IND-CPA secure	Prime order
[29]	Tolerant AND-gate policy based on bits string	√	×	×	Selectively IND-CPA secure	Prime order
[25]	Strict AND-gate policy	√	×	×	Selectively IND-CPA secure	Prime order
[22]	Threshold	√	×	×	Selectively IND-CCA2 secure	Prime order
[24]	Strict AND-gate policy	√	√	×	Selectively IND-CPA secure	Prime order
[28]	AND-gate policy with wildcards	√	×	×	Selectively IND-CPA secure	Prime order
Ours	Tolerant AND-gate policy based on bits string	√	√	√(privacy-preserving)	Selectively IND-CCA secure	Prime order

TABLE 2: Notations for comparison.

Notations	Meaning
$ \mathbb{Z}_p^* / G / G_T $	Size of element in the group $\mathbb{Z}_p^*/G/G_T$. $e: G \times G \rightarrow G_T \bmod p$ is a symmetric bilinear pairing.
T_e^G	Time for a group exponential operation in G (3.351 ms).
$T_e^{G_T}$	Time for a group exponential operation in G_T (0.538 ms).
T_p	Time for a symmetric bilinear pairing $e: G \times G \rightarrow G_T \bmod p$ (5.325 ms).
l	The number of rows of the LSSS matrix.
n	The number of the attributes in system.
$ S $	The number of the attributes in the user's attribute set.
W	The bit string-based AND-gate access policy $W = w_1 w_2, \dots, w_n$, where $w_i \in \{0, 1\}$, $1 \leq i \leq n$.
$ I_W $	The size of $I_W = \{i 1 \leq i \leq n, w_i = 1\}$.
$ \mathcal{S} $	The size of the set \mathcal{S} . \mathcal{S} is one subset of $\{1, 2, \dots, l\}$, that is, $\mathcal{S} \subseteq \{1, 2, \dots, l\}$; all the attributes in \mathcal{S} can satisfy the LSSS access policy.
t	The threshold of the threshold policy.
n_a	The number of the attributes in the threshold policy.

TABLE 3: Transmission load comparison.

Scheme	Private key length	Ciphertext length
[20]	$ S G + (n-1) G + G $	$2 G + G_T $
[21]	$ S G + (n-1) G + G $	$2 G + G_T $
[23]	$2 G $	$ G_T + 2 G $
[24]	$2 G $	$ G_T + 2 G $
[26]	$ S G $	$ G_T + 2 G $
[34]	$2 G + S G $	$ G_T + G + 2 G + G $
[19]	$2 G $	$ G + (n - I_W + 1) G $
[29]	$2 G + S G $	$2 G + G_T $
Ours	$2 G $	$3 G $

The only error event is that $T = R$, but it is queried to H_2 oracle. This occurs with probability (q_{H_2}/p) at most where p is the order of group G_T and q_{H_2} is the number of the queries to the oracle H_2 . So,

$$Pr[\mathcal{E}(\vec{p}, T = R) = 0] = \frac{1}{2} + \frac{q_{H_2}}{p}, \quad (23)$$

$$Pr[\mathcal{E}(\vec{p}, T = e(g_0, h_0)^{y^f(a)}) = 0] - Pr[\mathcal{E}(\vec{p}, T = R) = 0] = \text{Adv}_{\mathcal{A}} - \frac{q_{H_2}}{p}. \quad (24)$$

So, the simulator \mathcal{C} can solve the n-aMSE-DHH problem in PPT.

4. Evaluation and Implementation

4.1. Properties Evaluation. In this section, we compare our scheme with some related CP-ABE schemes in terms of the properties in Table 1. From Table 1, we can know that only our scheme provides “constant-size ciphertext,” “constant-size private key,” and “privacy-preserving outsourced decryption” simultaneously. The schemes in [23, 24] are also with constant-size ciphertext and constant-size private key, but their access structures—“Strict AND-gate Policy” are less expressive and too strict. Thus, these schemes [23, 24] cannot achieve fine-grained access control. And the work [25] based on [23] also uses the less expressive “Strict AND-gate Policy” as its access structure. So, the data owner in [25] also cannot customize the flexible access policy for his/her ciphertext. And these works [20–22] apply the Threshold policy in their schemes, so their schemes [20–22] cannot realize the precise and flexible attribute-based access control.

4.2. Theoretical Analysis and Simulation Experiments. In this section, we choose some representative schemes [19–21, 23, 26, 29, 34] in Table 1 as well as our scheme for theoretical analysis in terms of the transmission load and computational complexity. To make the theoretical comparison clearer, we adopt the symmetric bilinear pairing $e: G \times G \rightarrow G_T$ for the schemes to be compared and evaluated. The definitions of the notations for theoretical analysis are presented in Table 2. The evaluation of the transmission load is shown in Table 3. From Table 3, we can observe that in our scheme, no matter how many attributes a

user has and how complexity an access policy is, the length of the user’s private key is only $2|G|$ and the size of the ciphertext is only $3|G|$. The comparison of the computational complexity in terms of the five algorithms as AttrKeyGen, Encrypt, Blind KeyGen (by user), Online Decryption (by proxy server), and Offline Decryption (by users) is presented in Table 4.

To evaluate the actual performance in terms of the transmission overhead and computational complexity of our scheme, we use the PBC [42] cryptographic library to run the simulation experiments of our scheme as well as the scheme in [19, 29], which are also using the “Tolerant AND-gate Policy based on Bits String” as their access policies to ensure the single-variable principle. The hardware for the experiments is the i5-1135G7 2.4 GHz with 16 GB 3200 MHz RAM and OS is Windows 10 1909. To realize the symmetric bilinear pairing $e: G \times GG_T$ with the security level of 80 bits, we adopt the supersingular (symmetric) curve $E(\mathbb{F}_q): y^2 = x^3 + x \pmod q$ with embedding degree $k = 2$ in the field \mathbb{F}_q with the prime q of 512 bits. And G is an additive subgroup in the $E(\mathbb{F}_q)$ with the prime order r of 160 bits. In this case, $|G| = |G_T| = 512 \text{ (bits)} \times 2 = 1024 \text{ (bits)} = 128 \text{ (bytes)}$, $|\mathbb{Z}_p^*| = 160 \text{ (bits)} = 20 \text{ (bytes)}$. The execution time of the cryptographic operations has been listed in Table 2. The results of the experiments are shown in Figures 4 and 5. And we compare the consumption time of local decryption algorithm and privacy-preserving outsourced decryption algorithm by performing a comparison simulation experiment between the two algorithms. And the result of the comparison simulation experiment is shown in Figure 6. By doing this, we can easily detect that our privacy-preserving outsourced decryption algorithm can greatly ease the computing burden of IoT devices. If the

TABLE 4: Computational complexity comparison.

Scheme	AttrKeyGen	Encrypt	Online decryption (by proxy server)	Blind KeyGen (by user)/offline decryption (by user)
[20]	$ S T_e^G + (n-1)T_e^G + T_e^G$	$T_e^G + T_{eT}^{G_r} + T_e^G + (n+t)T_e^G$	\times	$\times / (((t-1) + 1)t/2)T_e^G + T_p + (n-1)T_e^G + T_p + T_e^G + T_p$
[21]	$ S T_e^G + (n-1)T_e^G + T_e^G$	$T_e^G + (n_a + 1)T_e^G + T_e^G + T_p + T_{eT}^{G_r}$	\times	$\times / (((t-1) + 1)t/2)T_e^G + T_p + (n_a - t)T_e^G + T_p$
[23]	$3T_e^G$	$T_{eT}^{G_r} + 2T_e^G$	\times	$\times / 2T_p$
[24]	$nT_e^G + 3T_e^G$	$T_{eT}^{G_r} + 3T_e^G + nT_e^G$	\times	$\times / 6T_p$
[26]	$2 S T_e^G$	$T_{eT}^{G_r} + T_e^G + T_e^G$	\times	$\times / 2T_p$
[34]	$3T_e^G + S T_e^G$	$T_{eT}^{G_r} + T_e^G + 3 S T_e^G + 2T_e^G$	$T_p + 2 S T_p + S T_{eT}^{G_r}$	$2T_e^G + S T_e^G/3T_e^G$
[19]	$2T_e^G$	$2(n - I_W + 1)T_e^G + T_e^G + T_{eT}^{G_r}$	\times	$\times / (n - I_W)T_e^G + (n - I_W + 1)T_e^G + 3T_p + T_e^G$
[29]	$2T_e^G + S T_e^G$	$2T_e^G + T_{eT}^{G_r}$	\times	$\times / 3T_p$
Ours	$2T_e^G$	$T_e^G + 2(n - I_W + 2)T_e^G + T_{eT}^{G_r}$	$(n - I_W)T_e^G + T_p + 2T_p$	$2T_e^G/3T_e^G$

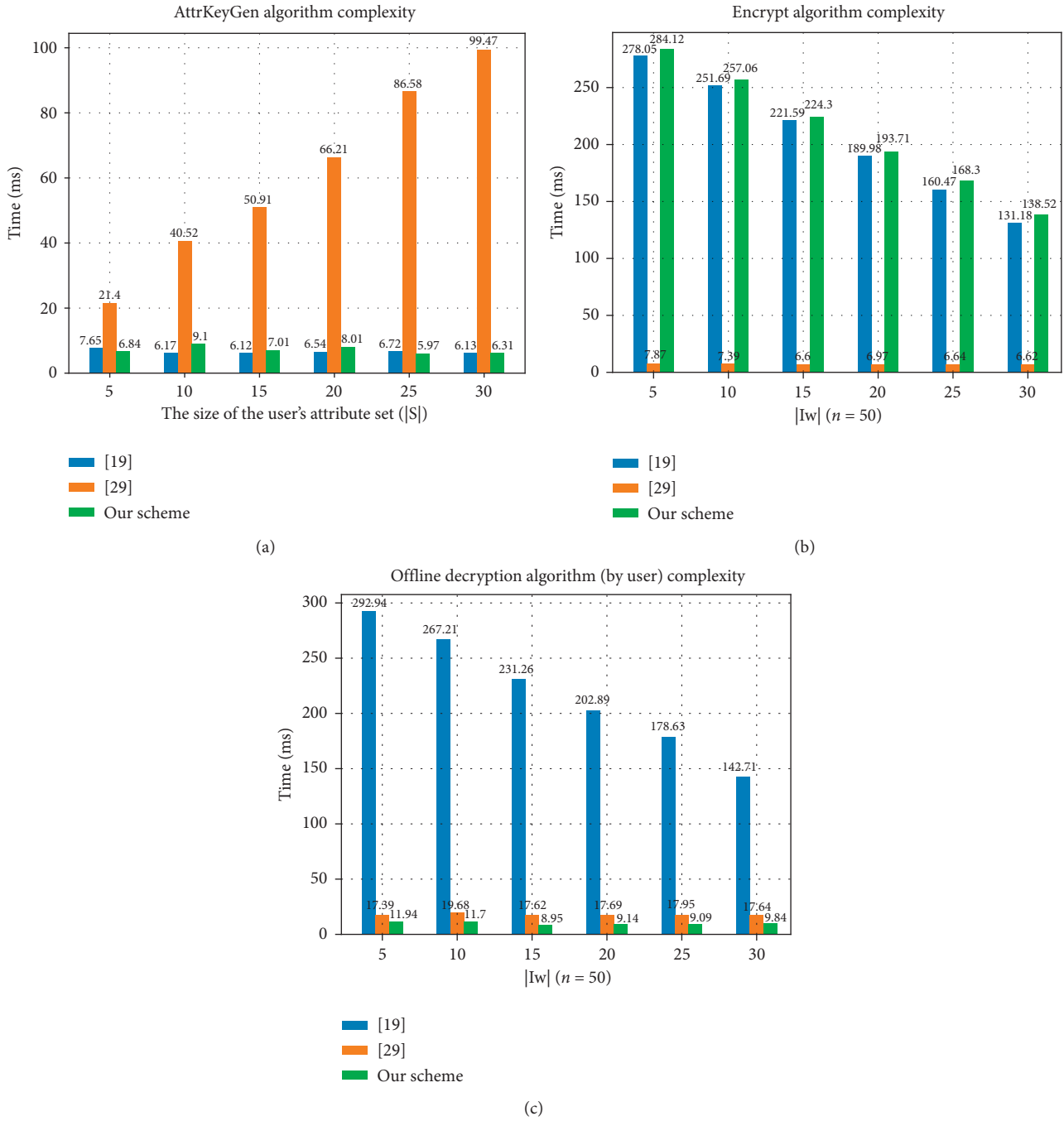


FIGURE 4: Algorithms' complexity.

user uses the local decryption algorithm to retrieve the data, the decryption time is positively correlated with the complexity of access structure. And what is more, by

using our privacy-preserving outsourced decryption algorithm, the decryption time by the IoT devices is constant.

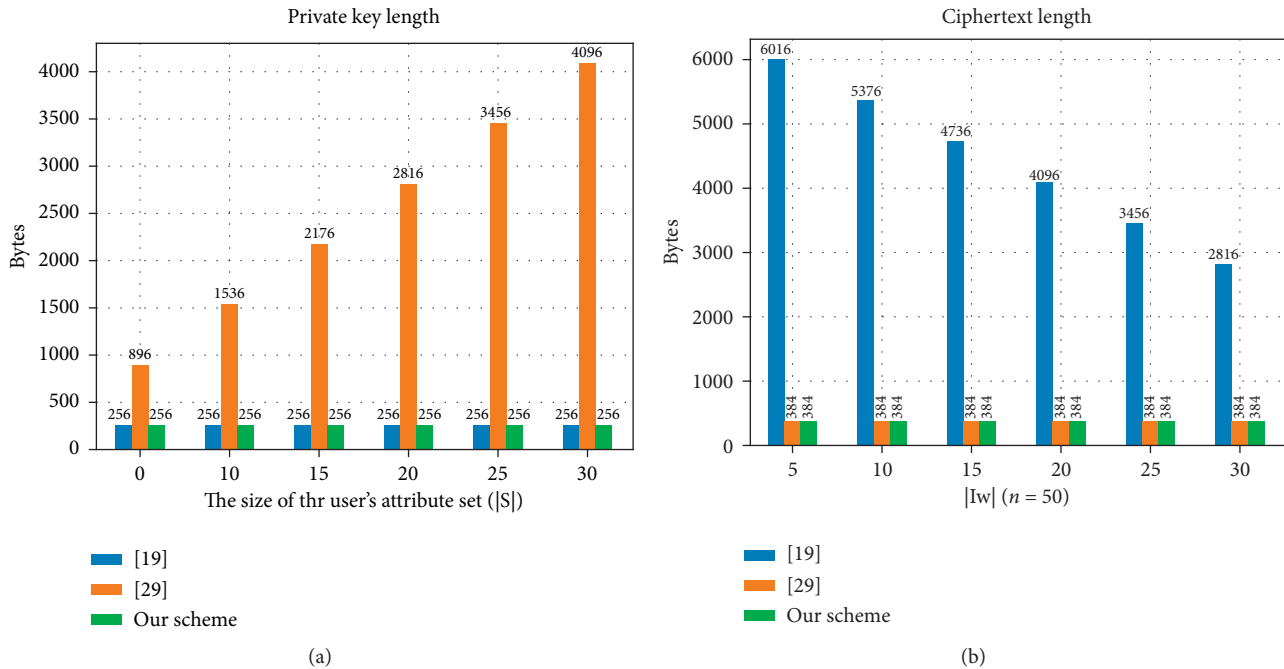


FIGURE 5: Comparison of occupied storage space (private key length) and transmission overhead (ciphertext length).

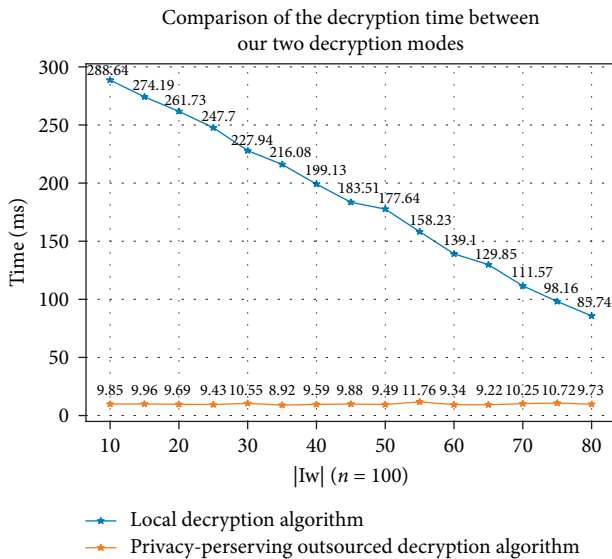


FIGURE 6: Comparison of decryption time between our two decryption modes.

5. Conclusion

In this paper, we propose a lightweight CP-ABE scheme with both constant-size ciphertexts and private keys for the IoT devices in cloud-assisted IoT environment. And users can outsource the decryption mission to the proxy server in a secure and private manner by using our privacy-preserving outsourced decryption algorithm. Our scheme can not only protect the privacy of users and confidentiality of the data but also reduce the communication overhead of the cloud-assisted IoT system and the computing pressure of users. Then, we rigorously prove that our scheme is selectively

IND-CCA secure by reducing the indistinguishability of our scheme to the n-aMSE-DHH problem. Finally, we compare our scheme with other CP-ABE schemes in terms of properties, transmission overhead, and computational complexity to show that our scheme is more applicable for the cloud-assisted IoT system. The main limitation and defect of our scheme is that our scheme cannot support the large universe attributes; that means users only can use the attributes which are defined by the AA in advance. In future research, to improve the flexibility and practicality of our scheme, we will make our scheme support the large universe attributes.

Data Availability

No data are used in this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

References

- [1] N. Shahid and S. Aneja, "Internet of things: vision, application areas and research challenges," in *Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 583–587, Palladam, India, February 2017.
- [2] O. Said and M. Masud, "Towards internet of things: survey and future vision," *International Journal of Computer Networks*, vol. 5, no. 1, pp. 1–17, 2013.
- [3] R. Mehta, J. Sahni, and K. Khanna, "Internet of things: vision, applications and challenges," *Procedia Computer Science*, vol. 132, pp. 1263–1269, 2018.

- [4] D. Mishra, A. Gunasekaran, S. J. Childe, T. Papadopoulos, R. Dubey, and S. Wamba, "Vision, applications and future challenges of internet of things," *Industrial Management & Data Systems*, vol. 116, no. 7, pp. 1331–1355, 2016.
- [5] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.
- [6] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, pp. 1–16, 2020.
- [7] H. Xiong, Z. Kang, J. Chen, J. Tao, C. Yuan, and S. Kumari, "A novel multiserver authentication scheme using proxy resignation with scalability and strong user anonymity," *IEEE Systems Journal*, pp. 1–12, 2020.
- [8] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.
- [9] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, San Francisco, CA, USA, June 2015.
- [10] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, 2014.
- [11] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3701–3709, 2018.
- [12] K. Finkenzerler, *RFID Handbook: fundamentals and Applications in contactless Smart cards, Radio frequency Identification and Near-field communication*, 478 pages, John Wiley & Sons, Hoboken, NJ, USA, 2010.
- [13] G. Gan, Z. Lu, and J. Jiang, "Internet of things security analysis," in *Proceedings of the 2011 International conference on Internet Technology and Applications*, pp. 1–4, Wuhan, China, August 2011.
- [14] Z. Hu, "The research of several key question of internet of things," in *Proceedings of the 2011 International Conference on Intelligence Science and Information Engineering*, pp. 362–365, Wuhan, China, August 2011.
- [15] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Aarhus, Denmark, May 2005.
- [16] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07)*, pp. 321–334, Berkeley, CA, USA, May 2007.
- [17] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 53–70, Berlin, Germany, April 2011.
- [18] Y. Rouselakis and B. Waters, "New constructions and proof methods for large universe attribute-based encryption," *IACR Cryptology EPrint Archive*, vol. 2012, p. 583, 2012.
- [19] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 5, pp. 763–771, 2014.
- [20] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *Proceedings of the International Workshop on Public Key Cryptography*, pp. 19–34, Beijing, China, April; 2010.
- [21] W. Susilo, G. Yang, F. Guo, and Q. Huang, "Constant-size ciphertexts in threshold attribute-based encryption without dummy attributes," *Information Sciences*, vol. 429, pp. 349–360, 2018.
- [22] W. Teng, G. Yang, Y. Xiang, T. Zhang, and D. Wang, "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 617–627, 2015.
- [23] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Information Security Practice and Experience*, F. Bao, H. Li, and G. Wang, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, pp. 13–23, 2009.
- [24] Y. Fan, S. Liu, G. Tan, and X. Lin, "Cscac: one constant-size cpabe access control scheme in trusted execution environment," *International Journal of Computational Science and Engineering*, vol. 19, no. 2, pp. 162–168, 2019.
- [25] D. Sethia, A. Shakya, R. Aggarwal, and S. Bhayana, "Constant size cp-abe with scalable revocation for resource-constrained iot devices," in *Proceedings of the 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0951–0957, New York, NY, USA, October 2019.
- [26] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *Provable Security*, S. S. M. Chow, J. K. Liu, L. C. K. Hui, and S. M. Yiu, Eds., Springer International Publishing, Cham, Switzerland, pp. 259–273, 2014.
- [27] N. Doshi and D. C. Jinwala, "Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption," *Security and Communication Networks*, vol. 7, no. 11, pp. 1988–2002, 2014.
- [28] J. Han, Y. Yang, J. K. Liu, J. Li, K. Liang, and J. Shen, "Expressive attribute-based keyword search with constant-size ciphertext," *Soft Computing*, vol. 22, no. 15, pp. 5163–5177, 2018.
- [29] W. Yang, R. Wang, Z. Guan, L. Wu, X. Du, and M. Guizani, "A lightweight attribute based encryption scheme with constant size ciphertext for internet of things," in *Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, June 2020.
- [30] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," in *Proceedings of the USENIX Security Symposium*, San Francisco, CA, USA, August 2011.
- [31] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.
- [32] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2119–2130, 2015.
- [33] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.
- [34] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on*

- Dependable and Secure Computing*, vol. 13, no. 5, pp. 533–546, 2015.
- [35] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, “Auditable σ -time outsourced attribute-based encryption for access control in cloud computing,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2017.
- [36] J. Li, X. Lin, Y. Zhang, and J. Han, “KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage,” *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.
- [37] H. Wang, D. He, and J. Han, “VOD-ADAC: anonymous distributed fine-grained access control protocol with verifiable outsourced decryption in public cloud,” *IEEE transactions on services computing*, vol. 13, no. 3, pp. 572–583, 2017.
- [38] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.-H. Yeh, “Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing,” *Future Generation Computer Systems*, vol. 97, pp. 453–461, 2019.
- [39] V. Odelu, A. K. Das, Y. S. Rao, S. Kumari, M. K. Khan, and K.-K. R. Choo, “Pairing-based CP-ABE with constant-size ciphertexts and secret keys for cloud environment,” *Computer Standards & Interfaces*, vol. 54, pp. 3–9, 2017.
- [40] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*, pp. 11–46, Technion-Israel Institute of Technology, Haifa, Israel, 1996.
- [41] A. Lewko and B. Waters, “New proof methods for attribute-based encryption: achieving full security through selective techniques,” in *Proceedings of the Annual Cryptology Conference*, pp. 180–198, Santa Barbara, CA, USA, August 2012.
- [42] B. Lynn, *The Pairing-Based Cryptography Library* Stanford, CA, USA, 2006.

Research Article

A Lightweight Intelligent Intrusion Detection Model for Wireless Sensor Networks

Jeng-Shyang Pan ¹, Fang Fan ^{1,2}, Shu-Chuan Chu ^{1,3}, Hui-Qi Zhao ²,
and Gao-Yuan Liu ²

¹College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, 266590 Shandong, China

²College of Intelligent Equipment, Shandong University of Science and Technology, Taian, 271019 Shandong, China

³College of Science and Engineering, Flinders University, 1284 South Road, Clovelly Park SA 5042, Australia

Correspondence should be addressed to Shu-Chuan Chu; scchu0803@gmail.com

Received 9 February 2021; Revised 21 April 2021; Accepted 23 April 2021; Published 3 May 2021

Academic Editor: Kuo-Hui Yeh

Copyright © 2021 Jeng-Shyang Pan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The wide application of wireless sensor networks (WSN) brings challenges to the maintenance of their security, integrity, and confidentiality. As an important active defense technology, intrusion detection plays an effective defense line for WSN. In view of the uniqueness of WSN, it is necessary to balance the tradeoff between reliable data transmission and limited sensor energy, as well as the conflict between the detection effect and the lack of network resources. This paper proposes a lightweight Intelligent Intrusion Detection Model for WSN. Combining k-nearest neighbor algorithm (kNN) and sine cosine algorithm (SCA) can significantly improve the classification accuracy and greatly reduce the false alarm rate, thereby intelligently detecting a variety of attacks including unknown attacks. In order to control the complexity of the model, the compact mechanism is applied to SCA (CSCA) to save the calculation time and space, and the polymorphic mutation (PM) strategy is used to compensate for the loss of optimization accuracy. The proposed PM-CSCA algorithm performs well in the benchmark functions test. In the simulation test based on NSL-KDD and UNSW-NB15 data sets, the designed intrusion detection algorithm achieved satisfactory results. In addition, the model can be deployed in an architecture based on cloud computing and fog computing to further improve the real-time, energy-saving, and efficiency of intrusion detection.

1. Introduction

Wireless sensor networks (WSN) provide the necessary underlying support for the Internet of Things and also build a landing platform for artificial intelligence (AI). Both of them have achieved deep integration and active promotion in WSN. The research and application of WSN have been involved in many fields, from the initial military reconnaissance to many aspects of social life, such as smart city, medical health, industrial production, environmental monitoring, and disaster warning [1]. WSN is a kind of wireless communication network that is composed of a large number of sensor nodes in a certain topological structure through self-organization. The sensor node monitors the

target area or object and transmits the collected sensor data to the user along the network route [2]. WSN can break through the limitations of traditional monitoring methods, which not only significantly reduces the cost of detection, but also greatly simplifies the cumbersome process. With the rapid development of sensor technology, wireless communication technology, big data, computing intelligence, etc., the low-cost and easy-to-deploy WSN can satisfy our urgent desire to learn more about the surrounding environment or ourselves. This technology will greatly enhance the breadth and depth of our perception of the world [3].

The application scenarios of WSN are complex and changeable. Compared with the traditional wired network, it faces many unique problems and challenges. First of all, the

computing power and storage capacity of a single sensor node are quite limited, and the communication ability between nodes is weak. Furthermore, the sensor nodes are often scattered in a wide range or in a complex or even harsh physical environment, which makes it difficult or impossible to perform maintenance tasks such as energy supply. In addition, it is an open network with dynamic and random topology. So, it is necessary to carry out a series of targeted research to ensure the real-time, energy-saving, reliability, and other operational requirements of WSN [4]. As a data-centric network, more and more sensitive data are collected, stored, transmitted, and processed in WSN. Its security problem has become increasingly serious [5]. Due to the limitations and characteristics of WSN itself, the data is easy to be destroyed, stolen, or tampered with. How to protect network security effectively in the face of various network attacks is an important research topic. Unfortunately, passive defense only through firewalls, access control, and other means is not enough to prevent all the network attacks. Intrusion detection is a proactive security protection technology that can monitor the operating status of network systems and detect intrusions such as internal attacks, external attacks, or misoperations, so that the network system can intercept and respond as necessary [6]. Wired network intrusion detection technology has been relatively mature and can be divided into two types: misuse-based and anomaly-based. The prerequisite of misuse detection is that the knowledge of attack method has been acquired, and the intrusion mode has been defined in advance. Intrusion is detected by judging whether the collected data characteristics match the intrusion pattern database. Therefore, it only has a high detection rate for specific attack methods and is invalid for unknown attacks. In order to cope with the endless emergence of various attacks, anomaly detection method can be considered. This method assumes that cyber attacks are uncommon compared to normal behaviors. By comparing the captured network behavior with normal patterns, it can be judged whether an intrusion has occurred. Anomaly detection can deal with unpredictable attacks, but it needs to learn a lot of historical data for training [7]. In order to improve the detection efficiency, the introduction of AI is expected. Many scholars have tried to apply artificial neural network [8, 9], machine learning [10], evolutionary computing [11–13], etc. to the field of intrusion detection and have achieved constructive research results [14]. However, WSN has its own characteristics and limitations in terms of network scale, computing power, storage space, energy supply, communication bandwidth, and networking mode, which makes it impossible to directly use the traditional intrusion detection system (IDS) architecture. AI technology generally requires high computing power and consumes relatively large amounts of running time, storage resources, and energy. Therefore, it is necessary to make modifications and adjustments to the WSN intrusion detection model according to the actual application scenarios and user requirements and seek the balance between security, energy consumption, real-time, and other objectives [15, 16].

Obviously, WSN intrusion detection is a technical problem with multiple constraints. How to provide a feasible and effective solution is an important issue to be solved urgently. Many scholars have done fruitful work in this field

[17]. Feature selection is an important and practical strategy for lightweight intrusion detection. Dimension reduction can improve the generalization performance and detection efficiency of intrusion detection. Literature [18] proposed a novel feature selection algorithm named DRFSA, combining an intelligent extension to the decision tree algorithm and convolution neural networks, to classify large volume of data in WSN. This model provides better intrusion detection accuracy, packet delivery ratio, and network throughput, while it reduces the network delay and false negative rate. The researchers also introduced a cryptographic mechanism to ensure the confidentiality and integrity of the data in the WSN and achieved encouraging results [19]. Literature [20] proposed a detection scheme for SQL injection attacks, which does not require access to the source code of the application, so it can be directly applied to the cloud environment. Literature [21] proposed a certificate-based aggregate signature scheme in WSN, which can resist forgery attacks. In addition, various machine learning and deep learning technologies are increasingly used to solve the WSN intrusion detection problem [22, 23].

This paper proposes a lightweight intelligent intrusion detection model for WSN. This model implements detection based on abnormal traffic data and can quickly and accurately discover attack behaviors in WSN. The k -nearest neighbors algorithm (kNN) is selected as the classifier. kNN is simple to implement and easy to understand. It supports nonlinear problems well and can provide relatively robust recognition results. The time complexity of the kNN is lower than that of the support vector machine (SVM) [24, 25]. Compared with naive Bayes algorithm [26], kNN has no hypothesis on data and is not sensitive to outliers. Therefore, compared with other machine learning algorithms, kNN meets the requirements of lightweight data classification. In order to further improve the classification effect, this paper uses evolutionary algorithm to optimize kNN. The selected evolutionary algorithm is the sine cosine algorithm (SCA). Among many metaheuristic optimization algorithms, SCA has low computational complexity, simple parameters, and good optimization performance. Taking into account the many limitations of WSN intrusion detection, the compact mechanism is applied to SCA (CSCA), which greatly reduces the time and space occupied in the optimization process. In order to ensure that the accuracy requirements are met, a polymorphic mutation strategy (PM) is designed, and an improved version of SCA is proposed (PM-CSCA). The organic combination of kNN and PM-CSCA constitutes a lightweight intelligent intrusion detection model for WSN. On the one hand, the intelligent detection is realized by means of evolutionary computation and machine learning; on the other hand, the computational burden of evolutionary algorithm is greatly reduced, so as to ensure the lightweight of the designed intrusion detection model.

This article is organized as follows: the second part is related work, introducing the SCA and kNN used in the intrusion detection algorithm proposed in this paper. The third part introduces the architecture of the intrusion detection system. The fourth part is the design of intrusion detection algorithm, including the improvement of SCA,

and how to combine it with kNN. The fifth part is the simulation results and discussion. The last part is the conclusion and future work.

2. Related Works

2.1. Sine Cosine Algorithm (SCA). SCA is a metaheuristic swarm intelligence optimization algorithm. The algorithm has a concise structure, has fewer parameters, and is easy to understand and implement. The search trajectory for the optimal solution is mainly affected by the sine and cosine functions [27–29].

The algorithm first initializes the population X , that is, to create N random candidate solutions $X_i (i = 1, 2, \dots, N)$. They are then guided to move through the search space using mathematical models based on sine and cosine functions. The optimization process is divided into two stages: global exploration and local exploitation. The formula for updating the position of the solution is as follows:

$$\begin{cases} X_i^{t+1} = X_i^t r_1 * \sin(r_2) * |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5, \\ X_i^{t+1} = X_i^t r_1 * \cos(r_2) * |r_3 P_i^t - X_i^t|, & r_4 < 0.5, \end{cases} \quad (1)$$

where t is the current number of iterations, P_i^t is the position of the current optimal solution in the i -th dimension, and $|\cdot|$ represents the absolute value. There are only four parameters involved here: r_1, r_2, r_3 and r_4 . $r_2 \in [0, 2\pi]$, which controls the distance the solution moves each time. $r_3 \in [0, 1]$, which gives a random weight to the current optimal solution. $r_4 \in [0, 1]$, which controls the switching between the sine and cosine update modes to ensure the same probability of using both. The above three parameters are random numbers that obey a normal distribution within their respective ranges. The parameter r_1 determines the direction of movement. When $r_1 < 1$, the solution will move to the area between the current position and the target position to exploit the local potential space. When $r_1 > 1$, the solution is to move away from the current optimal position to explore a larger search space. r_1 decreases linearly as the number of iterations increases, realizing the transition from exploration to exploitation. The updated formula of r_1 is shown in equation (2). Generally, $a = 2$, and T represents the maximum number of iterations.

$$r_1 = a - t \frac{a}{T}. \quad (2)$$

2.2. The k -Nearest Neighbors Algorithm (kNN). kNN algorithm is commonly used in data mining and machine learning. As one of the simplest classification algorithms, kNN is widely used in many fields. The core idea is that, in the feature space, if most of the k samples closest to a sample belong to a certain category, then this sample also belongs to this category and has all its characteristics. So, only the category of the k most similar samples is used to determine the category of the pending sample when making a classification decision [30, 31]. The implementation method is that all samples are mapped to points in D -dimensional space; k known samples nearest to the unknown sample are

selected as reference, and the distances between them are calculated, respectively; according to the majority voting rule, the unknown sample is classified into the category of most of its k -nearest neighbors. Obviously, kNN algorithm mainly considers three elements: the value of K , the way of distance measurement, and classification decision rules. The majority voting method is usually used to make decisions. The focus is usually on the choice of k value and the measurement of distance.

As the only parameter, the value of k has a crucial impact on the prediction results of kNN [32]. If k is relatively small, the approximate error of learning will decrease, but the estimation error will increase, and it is easy to learn noise. In severe cases, the model becomes complicated, and overfitting occurs. Similarly, if the k is large, the model will become too simple and underfit, which will also lead to inaccurate predictions. In actual engineering practice, k is generally selected by cross-validation. There is no fixed experience to guide the setting of k [33]. This has caused inconvenience in using the kNN algorithm.

We also need to pay attention to the distance measurement in the sample space. The shorter the distance, the higher the similarity between the two sample points, and conversely, the lower the similarity. The commonly used distance measurement methods are Minkowski Distance, Euclidean Distance, Manhattan Distance, Chebyshev Distance, Mahalanobis Distance, etc.

Suppose that there are two samples x_i and x_j in the D -dimensional feature space, which are expressed as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and $x_j = (x_{j1}, x_{j2}, \dots, x_{jD})$. The distance between the two samples is denoted as $d(x_i, x_j)$. kNN classifiers generally use Euclidean distance to measure the similarity between samples, as shown in

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^D (x_{ik} - x_{jk})^2}. \quad (3)$$

But in the process of classification, the importance of features is often different. Some features are strongly correlated with the classification results, some are weakly correlated, and some are even negatively correlated. If the distance between samples is largely dominated by weakly correlated or irrelevant features, it will easily lead to confusion in classification. To solve this problem, a certain weight $w_k (k = 1, 2, \dots, D)$ can be assigned to each feature dimension to express its importance. So, the distance between samples can be transformed into the following formula:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^D w_k (x_{ik} - x_{jk})^2}. \quad (4)$$

As a popular machine learning algorithm, kNN has been successfully applied in many fields [34, 35]. Some literatures try to improve it, mostly around the adjustment of parameter k [36, 37]. In fact, there is no universal experience in the determination of k , the selection of distance function, or

the setting of distance weight. All of these should be based on the distribution of samples, the characteristics of data, and the needs of analysis. This can be regarded as a typical optimization problem. With the help of the optimization ability of metaheuristic algorithm, a more reasonable and effective kNN classification model can be constructed [38].

3. WSN Intrusion Detection System Architecture

Intrusion detection is a security mechanism that collects information from several key nodes in the network system and analyzes it to try to find out whether there is any behavior that violates the security policy or signs of being attacked. The data in WSN shows an explosive growth trend. This requires high data processing capabilities, and intrusion detection also requires sufficient computing power.

The cloud computing platform has powerful computing and storage capabilities, as well as open, flexible, and shared characteristics, which provides a new research idea for WSN to break through the bottleneck restricting its development. In order to reduce the burden of importing and exporting data from the cloud and relieve the pressure of bandwidth shortage, fog computing can be further introduced. As a new generation of distributed computing, fog computing is closer to the edge of the network, providing space for a wider range of nodes to access. Comprehensive utilization of cloud computing and fog computing can achieve efficient collaborative computing. The powerful data processing and storage capabilities of the cloud computing platform provide technical support for big data analysis of WSN.

The intrusion detection system designed in this paper is deployed in the network architecture that combines cloud computing and fog computing, which can give full play to its advantages and better meet the data security requirements of WSN. The intrusion detection model can be deployed on the cloud server. Fog computing can be implemented by sink nodes with rich resources, which can independently assist the cloud to complete data processing, storage, and other tasks. WSN generally adopts hierarchical network structure and is divided into several clusters. The common sensor nodes in the cluster collect data and send it to the cluster heads, which transmit the data to the fog computing virtual network composed of sink nodes in a multihop manner. Figure 1 shows the architecture of the above WSN intrusion detection system.

4. Proposed Works

4.1. The Improvement of SCA. SCA is less computationally expensive compared with many other optimization algorithms. It is a reasonable choice for solving optimization problems that require low computational complexity and high real-time performance. In order to further improve the convergence speed of SCA, this paper uses the compact mechanism to make the algorithm more lightweight. Compact SCA (CSCA) can greatly reduce the computing load, but it will inevitably lose optimization accuracy to a certain extent. To solve this problem, a polymorphic

mutation strategy (PM) is proposed to enrich the diversity of population and compensate for the loss of precision. The framework structure of PM-CSCA is shown in Figure 2. In this part, the main ideas and implementation schemes of the proposed PM-CSCA are described in detail.

4.1.1. Compact SCA (CSCA). Compact is an optimization mechanism of swarm intelligence algorithm. After compact processing, the memory requirement of the algorithm will be significantly reduced [39, 40]. Because this technology will greatly alleviate the computational burden of the population-based metaheuristic algorithm, it is particularly suitable for devices with limited computing power and scarce storage space, such as sensor nodes, wearable devices, and embedded devices. SCA is an intelligent optimization algorithm based on population. The optimization process is as follows: N solutions are randomly generated in the D -dimensional space, and the positions of the solutions are constantly updated in the iterative process to realize the evolution of the population and finally find the global optimal solution. When the number of solutions is large, or the dimensionality is high, this calculation mode consumes more computing power. In application scenarios with high real-time requirements or limited storage space, the optimization algorithm needs to make necessary adjustments. The main idea of compact technology is to transform the original population into the form of a probability model that reflects its distribution characteristics. All operations on the original population are also transferred to its probability model [41, 42]. Since the number of variables and storage space required by the probabilistic model are far less than the original population, the algorithm runs more efficiently in time and space. The data structure of perturbation vector (PV) is usually used to describe the macroscopic probability distribution of the population: $PV^t = [\mu^t, \sigma^t]$. Here, μ and σ are the mean and standard deviation of PV, respectively, and t represents the current iteration number. Each pair of μ and σ in PV corresponds to a probability density function (PDF) [43] and is updated with the iteration of the algorithm. Generally, PDF is a truncated normal distribution in the interval $[-1, 1]$, and the calculation formula is as follows:

$$PDF_i(x) = \frac{\sqrt{2/\pi} e^{-(x-\mu_i)^2/2\sigma_i^2}}{\delta(\text{erf}(\mu_i + 1/\sqrt{2}\sigma_i) - \text{erf}(\mu_i - 1/\sqrt{2}\sigma_i))}. \quad (5)$$

It can be seen that PDF is a function of μ and σ . Among them, $x \in [-1, 1]$, erf represents error function, and i means dimension. Next, the cumulative distribution function (CDF) corresponding to the PDF can be obtained. The calculation method is as follows:

$$\begin{aligned} CDF &= \int_{-1}^x PDF dx \\ &= \int_{-1}^x \frac{\sqrt{2/\pi} e^{-(x-\mu)^2/2\sigma^2}}{\sigma(\text{erf}(\mu + 1/\sqrt{2}\sigma) - \text{erf}(\mu - 1/\sqrt{2}\sigma))} dx. \end{aligned} \quad (6)$$

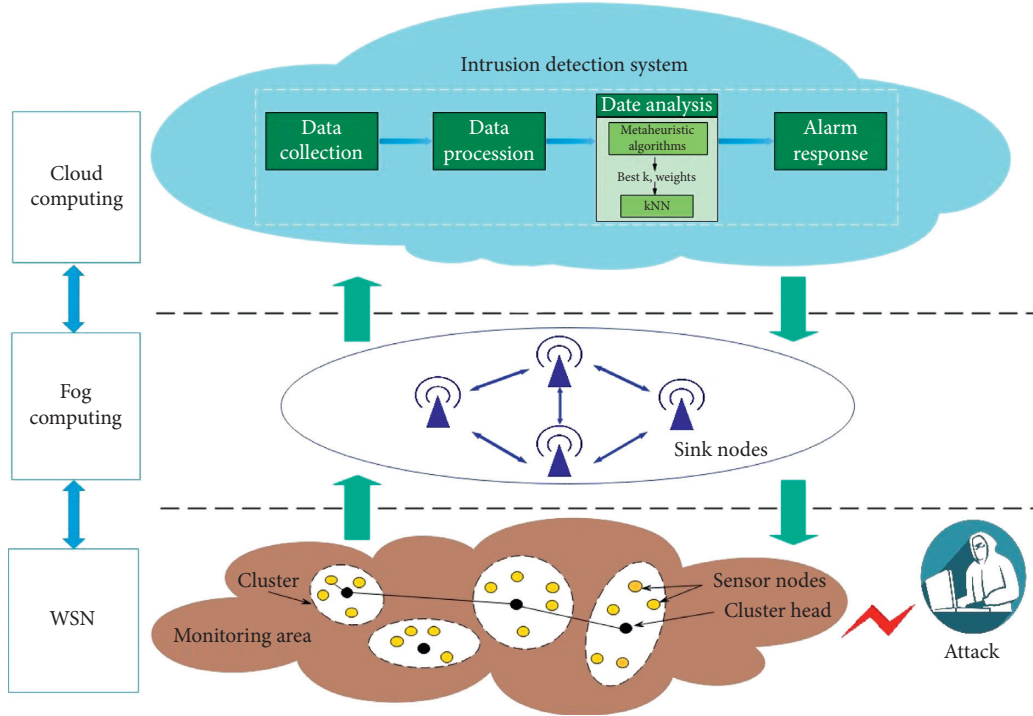


FIGURE 1: WSN intrusion detection system.

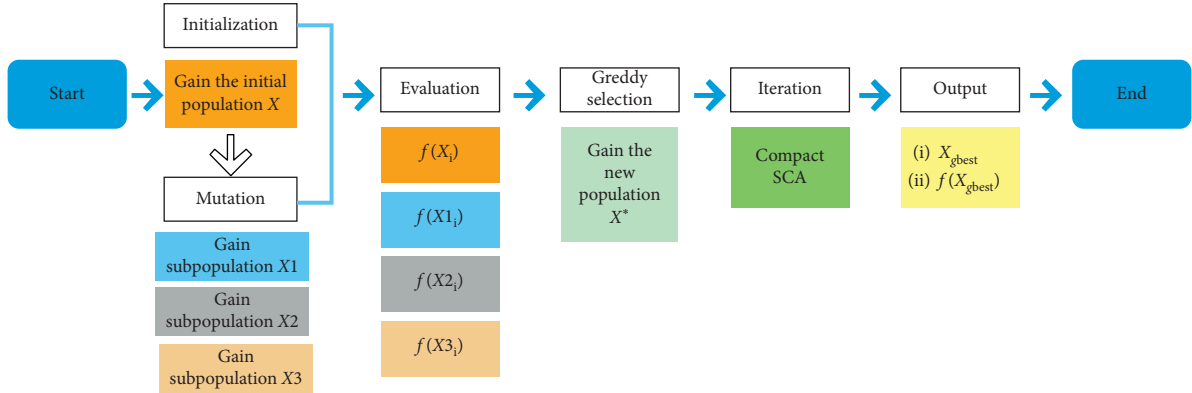


FIGURE 2: Framework of PM-CSCA.

Since PDF is a truncated normal distribution in the interval $[-1, 1]$, the CDF range is from 0 to 1. With the inverse function of CDF, a virtual solution y can be obtained by using PV:

$$y = \sqrt{2}\sigma \operatorname{erf}^{-1} \left(-\operatorname{erf} \left(\frac{\mu + 1}{\sqrt{2}\sigma} \right) - x \operatorname{erf} \left(\frac{\mu - 1}{\sqrt{2}\sigma} \right) + x \operatorname{erf} \left(\frac{\mu + 1}{\sqrt{2}\sigma} \right) \right) + \mu, \quad (7)$$

where $y \in [-1, 1]$, erf^{-1} is the inverse function of erf, and x is a random number between $[0, 1]$. It is necessary to map the virtual solution y to the solution y_{ds} of the decision space. Assuming that, in the D -dimensional decision space, the upper and lower limits of a certain dimension are ub and lb , respectively. y can be mapped to y_{ds} using

$$y_{ds} = y \times \frac{1}{2} (ub - lb) + \frac{1}{2} (ub + lb), \quad (8)$$

y_{ds} then attempts to move using equation (1). Evaluate the quality of the position before and after the movement, and record them as winner and loser, which are used to update the PV. Please see equations (9) and (10) for details.

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p} (\text{winner}_i - \text{loser}_i), \quad (9)$$

$$\sigma_i^{t+1} = \sqrt{(\sigma_i^t)^2 + (\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p} (\text{winner}_i - \text{loser}_i)^2}. \quad (10)$$

Among them, N_p is the number of solutions in the virtual population. In the process of updating PV, the global

optimal position is updated synchronously, and then the next iteration is carried out. With the help of compact mechanism, the original population is greatly reduced in size, and considerable benefits are achieved in both time and space [44–46]. However, due to the use of approximate probability distribution to simulate the real distribution of data, it is inevitable to bring the risk of loss of optimization accuracy, resulting in the occurrence of local traps or missing the global optima.

4.1.2. Polymorphic Mutation Strategy (PM). In order to make up for the possible loss of precision in compact SCA, a polymorphic mutation strategy (PM) is proposed. Based on the SCA initial population, a variety of distribution functions are introduced to realize polymorphic variation, and then the population with better quality is obtained through greedy selection. This can effectively increase the diversity of the population and create more opportunities for covering potential search areas, thereby improving the optimization accuracy. Three distribution functions are used here: Gaussian distribution, Cauchy distribution, and Levy' distribution. Gaussian distribution is a kind of thin-tailed distribution, which is an important probability distribution in statistics. It is often used to represent an uncertain random variable. Cauchy distribution belongs to fat-tailed distribution, and the possibility of extreme values is greater than that of Gaussian distribution. Among all the distributions, the generalized Cauchy distribution has the largest spreading characteristic. Levy' distribution can be approximated as heavy-tailed distribution. It can be used to generate Levy' flight, that is a random walk with relatively high probability of having a larger stride. So, the search efficiency of Levy' flight is better in the unknown environment or in large space [47].

In PM strategy, the population X initialized by SCA is randomly divided into three subpopulations: X_1, X_2, X_3 . Generate three variables between $[0,1]$: G, C, L , which obey different probability distributions: $G \sim N(\mu, \sigma^2)$, $C \sim C(\mu, \sigma^2)$, $L \sim \text{Levy}'(\lambda)$ ($\text{Levy}' \sim u = t^{-\lambda}$, $1 < \lambda \leq 3$). Perform mutation based on Gaussian distribution on X_1 to obtain a new subpopulation X_G , as shown in equation (11). In the same way, mutations based on Cauchy distribution and Levy' distribution are applied to X_2 and X_3 , respectively; and X_C and X_L are obtained according to equations (12) and (13).

$$X_G = X_1 + X_1 \otimes G, \quad (11)$$

$$X_C = X_2 + X_2 \otimes C, \quad (12)$$

$$X_L = X_3 + X_3 \otimes L, \quad (13)$$

Here, $G \sim N(0, 1)$, $C \sim C(1, 0)$, $L \sim \text{Levy}'(\lambda)$ ($\text{Levy}' \sim u = t^{-1.5}$). The product \otimes means entry-wise multiplications. According to the fitness value obtained by the evaluation function $f(\cdot)$, all solutions from the population X, X_1, X_2 and X_3 are sorted, and the better population X^* is obtained by greedy selection.

The computational complexity of the proposed PM-SCA depends on the following processes: initial population, polymorphic mutation, fitness evaluation, greedy selection, update population, and compact mechanism. Suppose that the number of solutions is n , the dimension is d , and the number of iterations is t . The computational complexity of initializing n d -dimensional solutions is $O(n \times d)$. The computational complexity of evaluating all solutions is $O(t \times n)$. The complexity of greedy selection is $O(n \times \log n)$. The computational complexity of updating all solutions is $O(t \times n \times d)$. Among them, the computational complexity of polymorphic mutation is $O(1)$, and the compact mechanism hardly brings about an increase in computational complexity. In general, the computational complexity of PM-SCA is the same as that of original SCA.

The pseudocode of PM-CSCA is shown in Algorithm 1. When the maximum number of iterations `max_iter` is reached, or other termination conditions are met, the global optimal solution x_{gbest} and its corresponding fitness value f_{gbest} are output.

4.1.3. Experiment Results. In order to test the performance of the algorithm, this part uses benchmark functions to carry out comparative experiments in the five algorithms of PM-CSCA, CSCA, SCA, Particle Swarm Optimization (PSO), and Whale Optimization Algorithm (WOA). 12 typical benchmark functions are selected here, including 3 unimodal functions ($F_1 \sim F_3$), 3 multimodal functions ($F_4 \sim F_6$), and 6 complex functions ($F_7 \sim F_{12}$), as shown in Table 1.

For the purpose of measuring the performance of the algorithm in a comprehensive and objective way, the algorithm runs independently 30 times in each experiment, recording the best value, average value (Avg), and standard deviation (Std), respectively. Please refer to Table 2 for specific data, and the best results have been marked in bold. The convergence curves of the benchmark functions are shown in Figure 3.

In the test of the three types of benchmark functions, PM-CSCA has achieved an absolute advantage in the algorithms participating in the comparison. The performance is particularly prominent in the optimization of complex functions. All indicators of the 6 complex functions ($F_7 \sim F_{12}$) have got the first place. PM-CSCA shows good optimization strength and reliable stability.

4.2. Combination of PM-CSCA and kNN. kNN parameter k and distance weight w_k determine the classification effect to a large extent. However, these aspects usually depend on the subjective decision of users, which brings great uncertainty to the performance of the algorithm. The PM-CSCA proposed in this article can be used to optimize the relevant parameters of kNN to obtain the best or approximately best configuration of the classifier.

The samples in the D -dimensional feature space correspond to the N solution vectors of the evolutionary algorithm: $X_i (i = 1, \dots, N)$, the specific form is shown in equation (14). The first dimension represents the


```

Initialize the parameters related to the algorithm: ub, lb, Dim, max_iter, PV( $\mu, \sigma$ );
Generate initial population  $X$  containing  $N$  individual  $X_i (i = 0, 1, 2, 3, \dots, N)$ ;
Divide  $X$  into three subpopulations  $X1, X2, X3$ ;
Realize the mutation of three subpopulations by using equations (11)–(13), respectively;
Evaluate each individual by the objective function;
Greedy selection: select  $N$  individuals from  $X, X1, X2$  and  $X3$  using greedy strategy, and get new population  $X^*$ ;
Do
  Update SCA parameter:  $r_1, r_2, r_3$  and  $r_4$ ;
  Get  $y_1$  from PV by equations (5)–(8);
  Update the  $y_1$  by SCA to get  $y_2$ ;
  Evaluate  $y_1$  and  $y_2$  by the objective function to get [winner, loser];
  for  $i = 1:Dim$ 
    Update PV via by equations (9) and (10);
    if  $f_{winner} < f_{gbest}$ 
      Update the best solution obtained so far;
    end
  while ( $t < max\_iter$ ) or (get the expected function value);
  Return the best solution obtained so far as the global optimum;

```

ALGORITHM 1: Pseudocode of PM-CSCA.

TABLE 1: Benchmark functions for testing.

Function	Dimension	Range	F_{min}
$F_1(x) = \sum_{i=1}^n x_i^2$	20	$[-100, +100]$	0
$F_2(x) = \max_i\{ x_i , 1 \leq x \leq n\}$	20	$[-100, +100]$	0
$F_3(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	20	$[-1.28, +1.28]$	0
$F_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	20	$[5.12, +5.12]$	0
$F_5(x) = (1/4000) \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	20	$[-32, +32]$	0
$F_6(x) = (\sum_{i=1}^5 i * \cos(i+1)x_1 + i) * (\sum_{i=1}^{25} i * \cos((i+1)x_2) + i)$	20	$[-5.12, +5.12]$	0
$F_7(x) = ((1/500) * \sum_{i=1}^{25} (1/i + \sum_{j=1}^2 (x_j - x_{ij})))$	20	$[-65, 65]$	0
$F_8 = 4 * x_1^2 - 2.1 * (x_1^6/3 + x_1 * x_2) - 4 * x_2^2 + 4 * x_2^4$	2	$[-5, +5]$	0
$F_9(x) = [1 + (x_1 + x_2 + x_3)^2 * (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$	2	$[-2, +2]$	3
$F_{10}(x) = -\sum_{i=1}^4 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[-10, +10]$	-10.1532
$F_{11}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[-10, +10]$	-10.4028
$F_{12}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[-10, +10]$	-10.5363

parameter K of kNN, which can be set as a random integer within a certain range as required. $w_{ij} \in [0, 1]$, the random number represents the j -th distance weight in the i -th solution. Evolutionary algorithm will continuously search and iterate under the guidance of the objective function and finally output the optimal solution or the approximate best [48–51], that is, the most suitable related parameters of kNN.

$$X_i = [k_i, w_{i1}, w_{i2}, \dots, w_{ij}, \dots, w_{iD}], \quad i = 1, \dots, N, j = 1, \dots, D. \quad (14)$$

5. Simulation Results and Discussion

Machine learning usually uses the following four criteria to evaluate the performance of the model: the true positive (TP), true negative (TN), false positive (FP), and the false negative (FN). In the field of intrusion detection, their specific meanings are as follows: TP is the number of actual attack records classified as attacks, TN is the number of actual normal records classified as normal, FP is the number

of actual normal records classified as attacks, and FN is the number of actual attack records classified as normal. They are also used to calculate a variety of performance evaluation indicators, such as detection rate (DR), false alarm rate (FAR), and accuracy rate (ACC). The calculation methods are as shown in the equations (15)–(17).

$$DR = \frac{TP}{(TP + FN)}, \quad (15)$$

$$FAR = \frac{FP}{(FP + TN)}, \quad (16)$$

$$ACC = \frac{(TP + TN)}{(TP + FN + FP + TN)}, \quad (17)$$

DR represents the probability of positive prediction among samples with normal real value. FAR is the probability of positive prediction among samples with abnormal real values. ACC is to divide the number of samples with correct prediction by the total number of samples, indicating the

TABLE 2: Results of PM-CSCA, CSCA, SCA, PSO, and WOA on 12 benchmark functions.

Function	Algorithm	Best value	Avg	Std
F_1	PM-CSCA	1.49E-95	1.62E-94	3.62E-21
	CSCA	6.02E-19	4.07E-03	9.37E+02
	SCA	7.04E-17	1.70E-19	4.03E-19
	PSO	201.2388	8.30E+01	3.00E+01
	WOA	7.91E-20	7.41E-19	1.03E-17
F_2	PM-CSCA	2.79E-08	5.91E-08	3.93809E-07
	CSCA	1.70E-06	3.22E-01	7.330382517
	SCA	0.00010521	4.39E-07	3.81943E-07
	PSO	6.0981	5.82E+00	1.426012414
	WOA	10.1124	9.35E-02	0.004428296
F_3	PM-CSCA	0.0024941	0.000918692	0.000285841
	CSCA	0.0030472	0.811201	0.429978878
	SCA	0.010223	0.000496352	0.000630681
	PSO	0.076345	0.01382486	0.002997254
	WOA	0.0032755	0.000493716	0.007144592
F_4	PM-CSCA	1.97E-11	1.74E-04	4.49035E-05
	CSCA	4.58E-09	6.23E-02	20.2012566
	SCA	3.52E+01	3.72E+00	0.40291051
	PSO	42.6913	2.75E+01	8.681105181
	WOA	0	0.00E+00	0.000095952
F_5	PM-CSCA	0	0.070084961	0.234385246
	CSCA	0.55431	0.4014294	16.96738349
	SCA	0.35735	0.018086667	0.171485471
	PSO	3.0755	2.14161	0.388616247
	WOA	0.12531	0.0712398	0.019007278
F_6	PM-CSCA	0.022866	0.0694455	0.015782176
	CSCA	0.10923	0.115816364	1738716.553
	SCA	0.10679	0.0784325	0.022381946
	PSO	8.9794	4.19622	1.311501322
	WOA	0.14293	0.001379359	0.000136212
F_7	PM-CSCA	0.99867	0.998402	0.59335955
	CSCA	1.0924	1.70102	0.966604051
	SCA	2.9821	1.401698	0.792971048
	PSO	1.993	0.998402	9.2957E-05
	WOA	2.9821	1.791716	0.907953884
F_8	PM-CSCA	0.00076939	0.001097476	0.000323742
	CSCA	0.0015264	0.00461873	0.003881268
	SCA	0.0015936	0.000929303	0.00040039
	PSO	0.001016	0.001423611	0.0004534
	WOA	0.0014995	0.001104929	0.000151675
F_9	PM-CSCA	3	3	2.22045E-16
	CSCA	3.0003	3.88066	0.007128226
	SCA	3.0001	3.0003	4.58258E-05
	PSO	3.0033	3.00784	0.000652993
	WOA	3.0001	3	3.68258E-05
F_{10}	PM-CSCA	-3.8499	-3.85357	0.000224499
	CSCA	-3.8544	-3.80696	0.854785298
	SCA	-3.8317	-3.83598	0.000801249
	PSO	-3.6506	-3.79914	0.006526132
	WOA	-3.8074	-3.75664	0.001567945
F_{11}	PM-CSCA	-4.9998	-4.35345	0.000961301
	CSCA	-2.9376	-3.68698	0.077142766
	SCA	-4.5372	-3.8552	0.002753834
	PSO	-1.9555	-3.81817	0.046047219
	WOA	-3.7214	-3.86085	0.010410014

TABLE 2: Continued.

Function	Algorithm	Best value	Avg	Std
F_{12}	PM-CSCA	-4.9514	-4.80574	0.201138842
	CSCA	-0.94657	-1.89185	0.998811381
	SCA	-4.7207	-3.995813	1.204583359
	PSO	-1.4388	-2.313657	0.934009458
	WOA	-2.4202	-2.24961	0.706751911
Statistics of the number of wins	PM-CSCA	11	10	10
	CSCA	0	0	0
	SCA	0	0	0
	PSO	0	0	0
	WOA	1	2	2

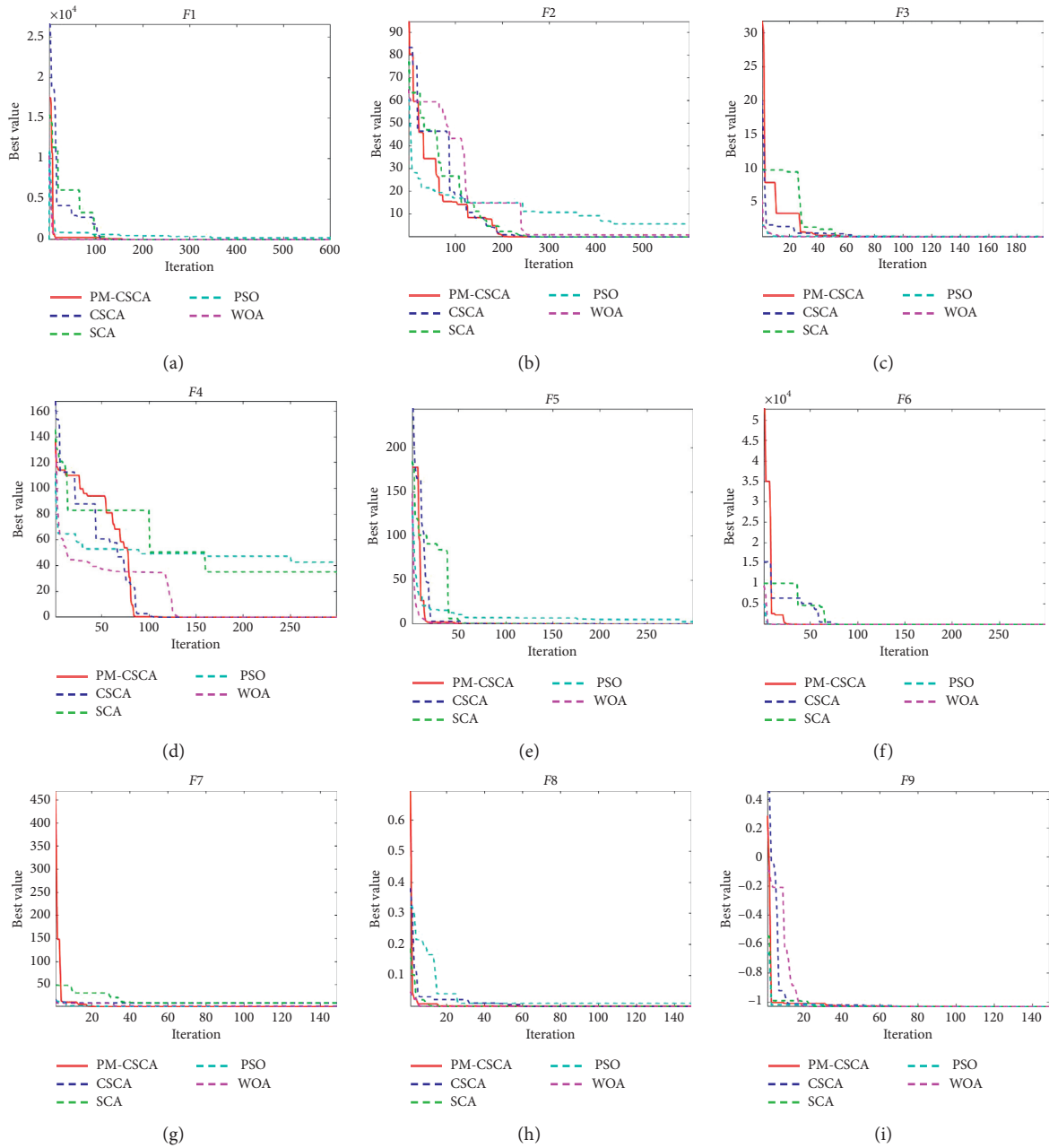


FIGURE 3: Continued.

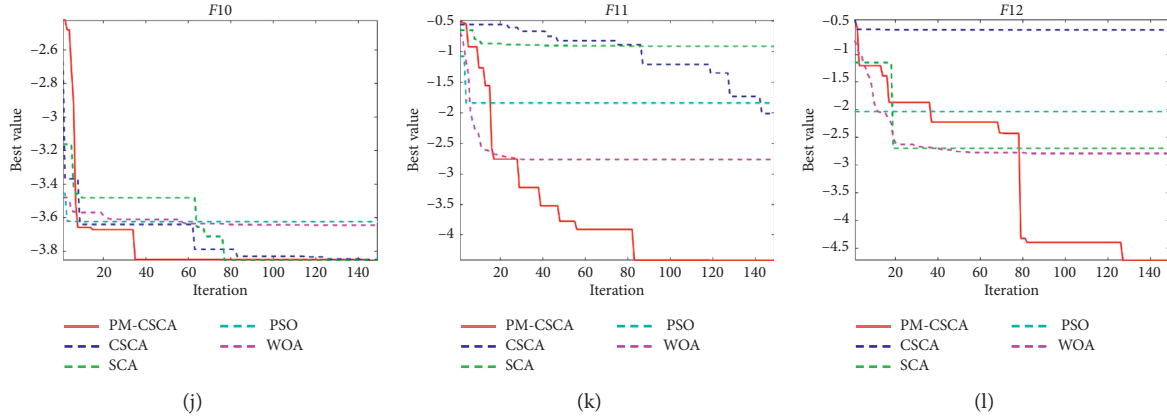


FIGURE 3: Convergence curves of 12 benchmark functions.

accuracy of prediction results. Obviously, the DR and ACC of intrusion detection should be high enough, while the FAR should be as low as possible. This article uses the ACC indicator as the fitness function $\text{fit}(\cdot)$, as shown in

$$\text{fit} = \frac{\text{TP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (18)$$

In order to verify the performance of the intrusion detection model, this paper used the NSL-KDD and UNSW-NB15 datasets commonly used in WSN intrusion detection to conduct simulation experiments. Each sample in the NSL-KDD dataset consists of 34 numerical features, 7 symbol features, and one-dimensional labels. There are five types of samples including normal data and 4 types of attack data. The four types of attacks are denial of service (DoS), sniffing (Probe), illegal access to superuser privileges by ordinary users (U2R), and illegal access from remote machines (R2L). NSL-KDD includes two training data sets (KDDTrain+, KDDTrain+_20%) and one test data set (KDDTest+). The training data set contains 21 types of attacks, and the test set adds 17 new attacks.

UNSW-NB15 is a more recent dataset than NSL-KDD, so it is more representative of real network traffic. It includes 100 GB of original network traffic and a total of 2540044 data samples. The features of this dataset are different from NSL-KDD and are more in line with the current network protocol model. It contains 10 categories, a normal category and 9 attack categories (i.e., Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worm).

Before the implementation of the algorithms, the datasets are preprocessed, including numerical, normalization, and other operations. The detection performance of five intrusion detection models was tested, respectively (SVM, kNN, PSO+kNN, SCA+kNN, and PM-CSCA+kNN). The experimental results are shown in

Table 3 and the average results of 10 independent experiments are recorded. The population size of the three evolutionary algorithms of PSO, SCA, and PM-CSCA is set to 30, and the number of iterations is 120. The model PM-CSCA+kNN achieved the best results on the three indicators of ACC, DR, and FAR (indicated in bold), which means that the model can identify most WSN attack behaviors and distinguish different types.

This paper introduces evolutionary algorithms in the intrusion detection model. Figure 4 shows the iterative process of the four optimization schemes. It was found that the result of optimizing kNN by SCA is always better than that of PSO; although CSCA has a great advantage in convergence speed, the accuracy is not stable, and sometimes it will fall into the local optimum; PM-CSCA has the best optimization effect on kNN, showing strong competitiveness both in accuracy and speed.

The confusion matrix is used to evaluate the accuracy of the four detection models on NSL-KDD, as shown in Figure 5. The horizontal axis represents the predicted value, and the vertical axis represents the true value, which visually shows the misclassification of each category. It can be seen that PM-CSCA+kNN has the best detection effect.

For WSN intrusion detection systems, reducing the false alarm rate is a challenge. We conducted five independent experiments ($E1 \sim E5$) on two data sets. Figure 6 Intuitively shows the comparison result of the false alarm rate of four different detection algorithms. It can be seen that the false alarm rate of PM-CSCA+kNN is extremely stable at a low level. For the convenience of showing the relationship between DR and FAR, the Receiver Operating Characteristics (ROC) curves based on two datasets are drawn, as shown in Figure 7. The ROC curves corresponding to the algorithm proposed in this article are all closest to the upper left boundary, so the effect of this prediction model is the best.

TABLE 3: Performance indicators comparison of five intrusion detection models (SVM, kNN, PSO + kNN, SCA + KNN, and PM-CSCA + kNN) on NSL-KDD and UNSW-NB15 datasets.

Model	NSL-KDD			UNSW-NB15		
	ACC (%)	DR (%)	FAR (%)	ACC (%)	DR (%)	FAR (%)
SVM	92.116	92.459	9.3684	92.6	91.82	8.73
kNN	94.100	95.370	8.1300	86.64	85.35	11.48
PSO + kNN	95.890	96.078	4.2105	90.64	89.86	10.08
SCA + kNN	97.952	97.321	1.6575	93.84	93.28	7.95
PM-CSCA + kNN	99.327	99.206	0.5848	98.27	97.94	5.82

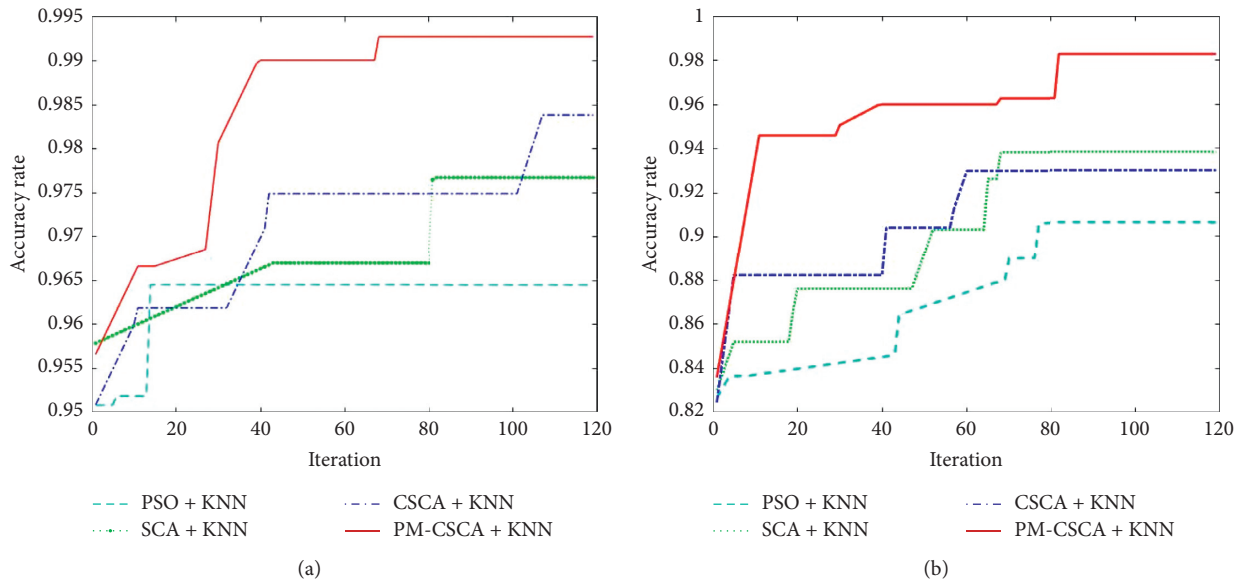


FIGURE 4: Comparison of the convergence curves of PSO, SCA, CSCA, and PM-CSCA. (a) Based on NSL-KDD dataset. (b) Based on UNSW-NB15 dataset.

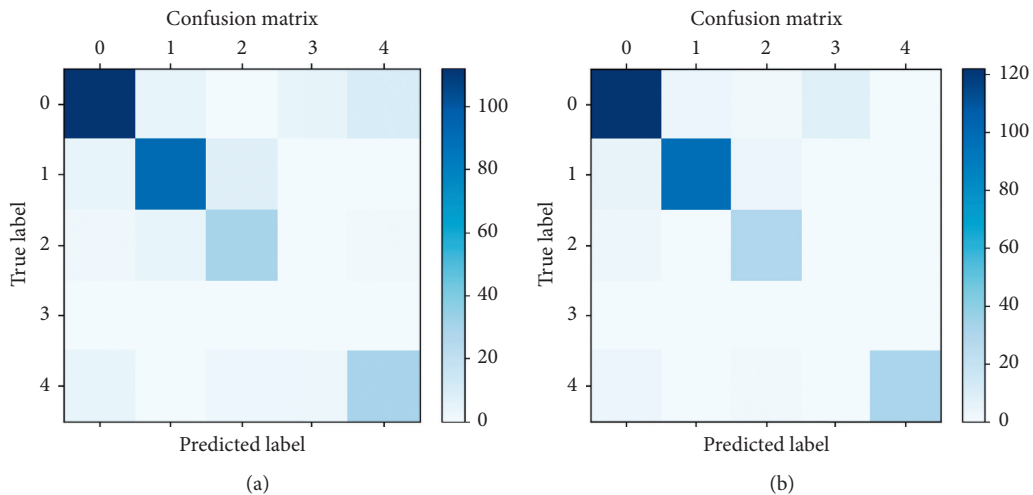


FIGURE 5: Continued.

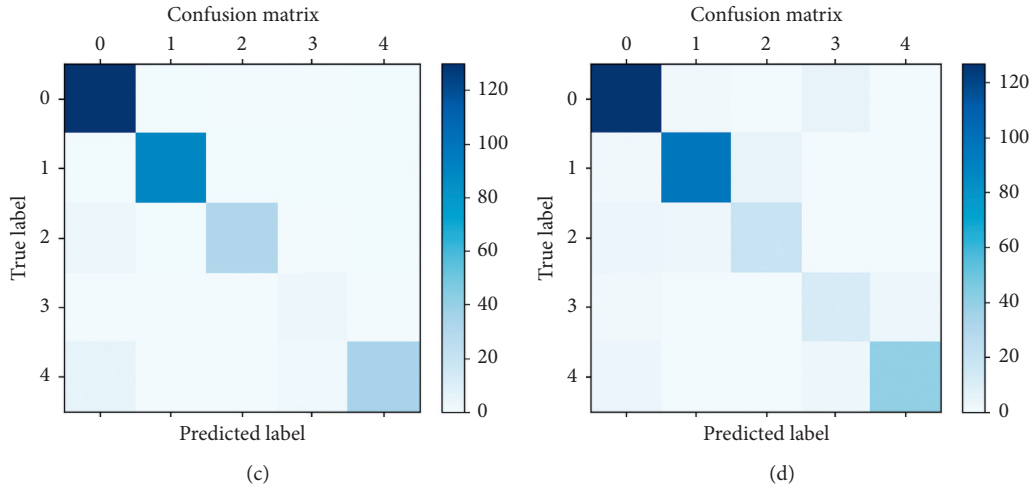


FIGURE 5: Confusion matrices of four intrusion detection models on NSL-KDD. (a) kNN. (b) PSO + kNN. (c) SCA + kNN. (d) PM-CSCA + kNN.

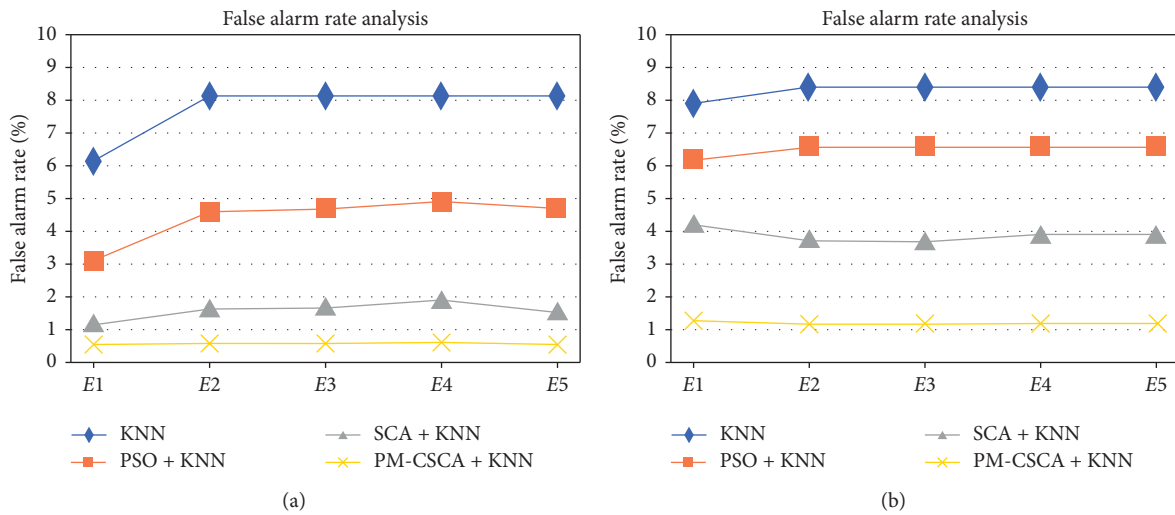


FIGURE 6: Comparison of the false alarm rate of kNN, PSO + kNN, SCA + kNN, and PM-CSCA + kNN. (a) Based on NSL-KDD dataset. (b) Based on UNSW-NB15 dataset.

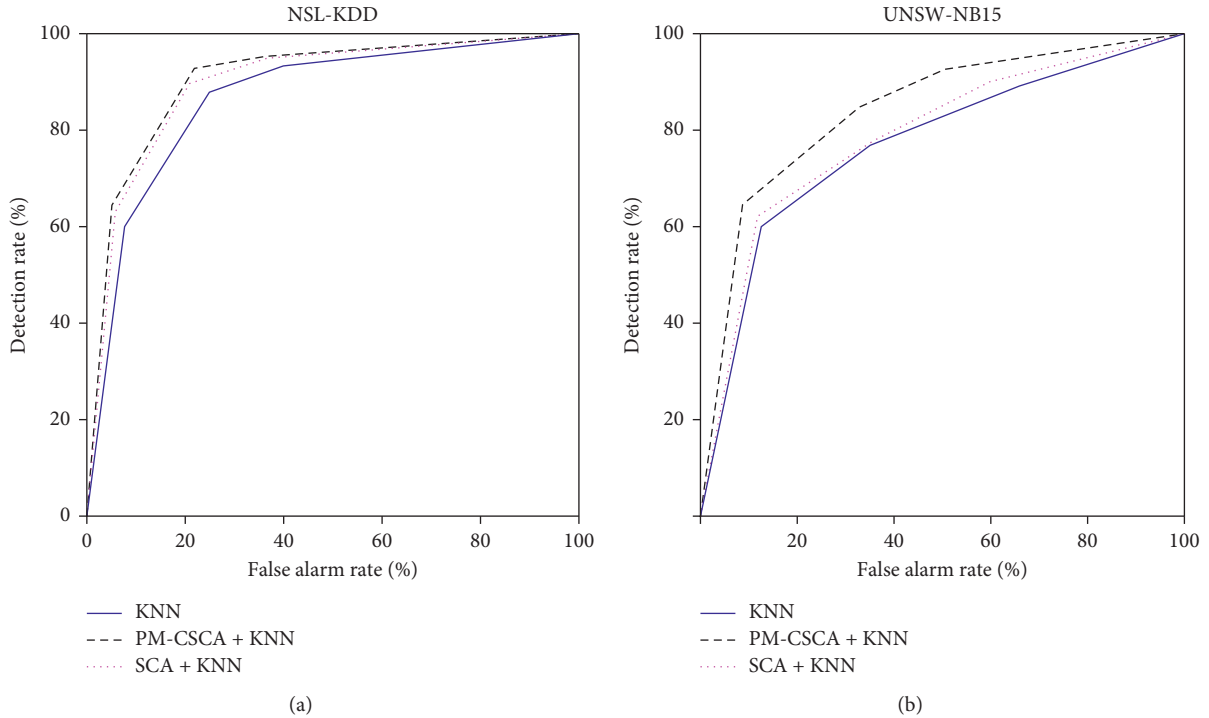


FIGURE 7: ROC curves of three classification algorithms on two datasets.

6. Conclusion and Future Works

Intrusion detection is one of the key issues that need to be solved urgently in practical applications of WSN. With the continuous expansion of the service area and the rapid rise of data volume, the threat and consequences of network attacks in WSN cannot be ignored. Most of the existing intrusion detection systems can only deal with specific types of attacks, and they are powerless against unknown attacks [52]. And while protecting the network security, it inevitably increases the energy consumption and transmission delay. These problems need to be paid more attention in WSN. This paper proposes a lightweight and intelligent intrusion detection model for WSN, which comprehensively considers security, energy saving, and real-time. Intelligent anomaly detection is realized through the joint use of kNN and SCA. The introduction of evolutionary algorithms makes the kNN classifier change from lazy learning to active optimization in the setting of its parameters, which significantly improves the detection accuracy. kNN and SCA are both algorithms with less computation and easy implementation, which meet the requirements of lightweight model. In order to be more efficient, this article proposes an improved version of SCA: PM-CSCA. Two technologies are integrated: compact mechanism greatly reduces the time and space required for algorithm, and PM strategy ensures the optimization accuracy, and these have been verified in tests based on benchmark functions. PM-CSCA shows good optimization ability in the benchmark function test. In simulation experiments on public data set, the intrusion detection model

proposed in this paper has also been proved to be feasible and effective. In addition, the intrusion detection system for WSN is deployed in the hybrid computing mode. Cloud computing, fog computing, and AI work together to provide a feasible and efficient solution for maintaining data security in WSN.

We will do further research on the lightweight and intelligent WSN intrusion detection model, for example, how to use unsupervised machine learning techniques to deal with unpredictable cyber attacks [53]. Furthermore, more core technologies of evolutionary computing can be applied to solve big data or large-dimensional problems encountered in intrusion detection [54, 55].

The following abbreviations are used in this manuscript:

Abbreviations

WSN:	Wireless sensor networks
kNN:	k-nearest neighbor algorithm
SCA:	Sine cosine algorithm
CSCA:	Compact SCA
PM:	Polymorphic mutation
AI:	Artificial intelligence
IDS:	Intrusion detection system
SVM:	Support vector machine
PV:	Perturbation vector
PDF:	Probability density function
CDF:	Cumulative distribution function
PSO:	Particle swarm optimization
WOA:	Whale optimization algorithm.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] H. Ghayvat, S. Mukhopadhyay, X. Gui, and N. Suryadevara, "WSN- and IOT-based smart homes and their extension to smart buildings," *Sensors*, vol. 15, no. 5, pp. 10350–10379, 2015.
- [2] Q.-W. Chai, S.-C. Chu, J.-S. Pan et al., "Applying adaptive and self-assessment fish migration optimization on localization of wireless sensor network on 3-D terrain," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 2, pp. 90–102, 2020.
- [3] D. Ciuonzo, P. S. Rossi, and P. K. Varshney, "Distributed detection in wireless sensor networks under multiplicative fading via generalized score-tests," *IEEE Internet of Things Journal*, vol. 1, p. 1, 2021.
- [4] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "Energy efficient routing algorithm with mobile sink support for wireless sensor networks," *Sensors*, vol. 19, no. 7, p. 1494, 2019.
- [5] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. De Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.
- [6] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," 2018, <https://arxiv.org/abs/1802.09089>.
- [7] I. Butuan, S. D. Merger, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 266–282, 2013.
- [8] T. Ma, F. Wang, J. Cheng, Y. Yu, and X. Chen, "A hybrid spectral clustering and deep neural network ensemble algorithm for intrusion detection in sensor networks," *Sensors*, vol. 16, no. 10, p. 1701, 2016.
- [9] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udipi, India, December 2017.
- [10] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
- [11] K. A. P. Costa, L. A. M. Pereira, R. Y. M. Nakamura, C. R. Pereira, J. P. Papa, and A. Xavier Falcão, "A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks," *Information Sciences*, vol. 294, pp. 95–108, 2015.
- [12] Y. Xue, W. Jia, X. Zhao et al., "An evolutionary computation-based feature selection method for intrusion detection," *Security and Communication Networks*, vol. 2018, Article ID 2492956, 10 pages, 2018.
- [13] C. Umarani and S. Kannan, "Intrusion detection system using hybrid tissue growing algorithm for wireless sensor network," *Peer-to-Peer Networking and Applications*, vol. 13, no. 3, pp. 752–761, 2019.
- [14] Z. Sun, Y. Xu, G. Liang et al., "An intrusion detection model for wireless sensor networks with an improved V-detector algorithm," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 1971–1984, 2017.
- [15] A. A. Aburomman and M. B. Ibne Reaz, "A novel SVM-kNN-PSO ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360–372, 2016.
- [16] M. H. Ali, B. A. D. Al Mohammed, A. Ismail, and M. F. Zolkipli, "A new intrusion detection system based on fast learning network and particle swarm optimization," *IEEE Access*, vol. 6, pp. 20255–20261, 2018.
- [17] G. Bovenzi, G. Aceto, D. Ciuonzo et al., "A hierarchical hybrid intrusion detection approach in iot scenarios," 2020.
- [18] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. Santhosh Kumar, M. Selvi, and K. Arputharaj, "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Communications*, vol. 14, no. 5, pp. 888–895, 2020.
- [19] C.-M. Chen, B. Xiang, T.-Y. Wu, and K.-H. Wang, "An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks," *Applied Sciences*, vol. 8, no. 7, p. 1074, 2018.
- [20] T.-Y. Wu, C.-M. Chen, X. Sun, S. Liu, and J. C.-W. Lin, "A countermeasure to SQL injection attack for cloud environment," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5279–5293, 2017.
- [21] J. N. Chen, F. M. Zou, T. Y. Wu et al., "A new certificate-based aggregate signature scheme for wireless sensor networks," *J. Inf. Hiding Multimedia Signal Process*, vol. 9, no. 5, pp. 1264–1280, 2018.
- [22] D. S. Vijayakumar and S. Ganapathy, "Machine learning approach to combat false alarms in wireless intrusion detection system," *Computer and Information Science*, vol. 11, no. 3, pp. 67–81, 2018.
- [23] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," *Soft Computing*, vol. 24, no. 22, pp. 17265–17278, 2020.
- [24] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Computers & Security*, vol. 77, pp. 304–314, 2018.
- [25] M. Safaldin, M. Otair, and L. Abualigah, "Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1–18, 2020.
- [26] Z. Nannan, W. Lifeng, Y. Jing et al., "Naive bayes bearing fault diagnosis based on enhanced independence of data," *Sensors*, vol. 18, no. 2, p. 463, 2018.
- [27] S. Mirjalili, "SCA: a Sine Cosine Algorithm for solving optimization problems," *Knowledge-based Systems*, vol. 96, pp. 120–133, 2016.
- [28] S. Gupta, K. Deep, S. Mirjalili, and J. H. Kim, "A modified sine cosine algorithm with novel transition parameter and mutation operator for global optimization," *Expert Systems with Applications*, vol. 154, Article ID 113395, 2020.
- [29] A. Bhadoria, S. Marwaha, and V. K. Kamboj, "An optimum forceful generation scheduling and unit commitment of thermal power system using sine cosine algorithm," *Neural Computing and Applications*, vol. 32, no. 1, pp. 1–30, 2020.
- [30] Y. L. Qiao, J.-S. Pan, and S. H. Sun, "Improved K nearest neighbor classification algorithm," in *Proceedings of the 2004*

- IEEE Asia-Pacific Conference on Circuits and Systems*, Tainan, Taiwan, December 2004.
- [31] Y. Chen, X. Hu, W. Fan et al., "Fast density peak clustering for large scale data based on kNN," *Knowledge-Based Systems*, vol. 187, Article ID 104824, 2020.
- [32] S. Zhang, D. Cheng, Z. Deng, M. Zong, and X. Deng, "A novel k NN algorithm with data-driven k parameter computation," *Pattern Recognition Letters*, vol. 109, pp. 44–54, 2018.
- [33] S. Zhang, X. Li, M. Zong, X. Zhu et al., "Efficient knn classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2017.
- [34] Z. Deng, X. Zhu, and D. Cheng, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, 2016.
- [35] H. Yang, S. Liang, J. Ni et al., "Secure and efficient kNN classification for industrial Internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 11, 2020.
- [36] K. Wang, X. Yu, Q. Xiong et al., "Learning to improve WLAN indoor positioning accuracy based on DBSCAN-KRF algorithm from RSS fingerprint data," *IEEE Access*, vol. 7, pp. 72308–72315, 2019.
- [37] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "K-nearest neighbor classification over semantically secure encrypted relational data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1261–1273, 2014.
- [38] Z. Y. Meng, J.-S. Pan, and L. P. Kong, "Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution," *Knowledge -Based Syst*, vol. 141, pp. 92–112, 2018.
- [39] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.
- [40] E. Mininno, F. Cupertino, and D. Naso, "Real-valued compact genetic algorithms for embedded microcontroller optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 203–219, 2008.
- [41] E. Mininno, F. Neri, F. Cupertino et al., "Compact differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 32–54, 2010.
- [42] F. Neri, E. Mininno, and G. Iacca, "Compact particle swarm optimization," *Information Sciences*, vol. 239, pp. 96–121, 2013.
- [43] C. H. Chen, F. Song, F. J. Hwang et al., "A probability density function generator based on neural networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 541, Article ID 123344, 2020.
- [44] X. S. Xue and J. F. Chen, "Using compact evolutionary tabu search algorithm for matching sensor ontologies," *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.
- [45] X. S. Xue and J. F. Chen, "Optimizing sensor ontology alignment through compact co-firefly algorithm," *Sensors*, vol. 20, no. 7, pp. 1–15, 2020.
- [46] X. S. Yang and S. Deb, "Cuckoo search via lévy flights," in *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, December 2009.
- [47] P. C. Song, J.-S. Pan, and S.-C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Applied Soft Computing*, vol. 94, Article ID 106443, 2020.
- [48] H. Wang, W. Wang, Z. Cui et al., "A new dynamic firefly algorithm for demand estimation of water resources," *Information Sciences*, vol. 438, pp. 95–106, 2018.
- [49] X. Wang, J.-S. Pan, and S. C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32018–32030, 2020.
- [50] J.-S. Pan, X. X. Sun, S.-C. Chu et al., "Digital watermarking with improved SMS applied for QR code," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104049, 2021.
- [51] T. B. Jiang and S.-C. Chu, "Parallel charged system search algorithm for energy management in wireless sensor network," in *Proceedings of the 2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, Shenyang, China, October 2020.
- [52] O. A. Osanaiye, A. S. Alfa, and G. P. Hancke, "Denial of service defence for resource availability in wireless sensor networks," *IEEE Access*, vol. 6, pp. 6975–7004, 2018.
- [53] H. Qu, Z. Qiu, X. Tang et al., "Incorporating unsupervised learning into intrusion detection for wireless sensor networks with structural co-evolvability," *Applied Soft Computing*, vol. 71, pp. 939–951, 2018.
- [54] S. F. Qin, C. L. Sun, G. C. Zhang et al., "A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems," *Complex & Intelligent Systems*, vol. 6, no. 2, pp. 263–274, 2020.
- [55] H. Wang, M. G. Liang, C. L. Sun et al., "Multiple-strategy learning particle swarm optimization for large-scale optimization problems," *Complex & Intelligent Systems*, vol. 23, 2020.

Review Article

A Bibliometric Analysis of Edge Computing for Internet of Things

Yiou Wang,^{1,2} Fuquan Zhang ,³ Junfeng Wang,² Laiyang Liu,² and Bo Wang⁴

¹Beijing Institute of Science and Technology Information, Beijing 100044, China

²School of Computer Science & Technology, Beijing Institute of Technology, Beijing 100081, China

³Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350117, China

⁴Internet Finance Department, Huaxia Bank, Beijing 100020, China

Correspondence should be addressed to Fuquan Zhang; 8528750@qq.com

Received 25 February 2021; Revised 19 March 2021; Accepted 31 March 2021; Published 9 April 2021

Academic Editor: Shehzad Ashraf Chaudhry

Copyright © 2021 Yiou Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, with the emergence of many Internet of Things applications such as smart homes, smart city, and connected vehicles, the amount of network edge data increases rapidly. Now, edge computing for Internet of Things has attracted the research interest of many researchers. Then, a thorough analysis of the current body of knowledge in edge computing for Internet of Things is conducive to a comprehensive understanding of the research status and future trends in this field. In this paper, a bibliometric analysis of edge computing for Internet of Things was performed using the Web of Science (WoS) Core Collection dataset. The relevant literature studies published in this field were quantitatively analyzed based on a bibliometric analysis method combined with VOSviewer software, and the development history, research hotspots, and future directions of this field were studied. The research results show that the number of literature studies published in the field of edge computing for Internet of Things is on the rise over time, especially after 2017, and the growth rate is accelerating. China and USA take the lead position in the number of literature studies published. Zhang is the most productive author, and Satyanarayanan is the most influential author. IEEE Access and IEEE Internet of Things Journal are the main journals in this field. Beijing University of Posts Telecommunications has published most literature studies. Research hotspots of edge computing for Internet of Things mainly include specific problem research such as resource management, architecture research, application research, and fusion research of this field with some other fields such as artificial intelligence and 5G.

1. Introduction

At present, the global Internet of Things (IoT) has entered the third wave of development. In 2018, and the number of IoT connections around the world had reached about 8 billion [1]. Artificial Intelligence (AI) technologies [2], such as deep learning, provide technical supports for intelligent analysis of massive data in the IoT. With the rapid development of IoT and AI, more and more intelligent applications appear in front of people, such as smart home, smart city, industrial IoT, and Internet of vehicles. These applications are usually resource-intensive [3], which bring great challenges to resource-constrained terminal devices. In the traditional cloud computing architecture, data need to be transferred centrally to the cloud for processing. However,

such a large amount of data transmitted in the IoT will increase the network load, resulting in transmission congestion and data processing delay. At the same time, the transmission of these massive data in the IoT will also increase the risk of data leakage, which puts the privacy and confidentiality of data at risk. Therefore, the processing of data information relying only on traditional cloud computing cannot be completed effectively. At this time, edge computing arises at the historical moment.

Pang et al., based on cloud computing, put forward the concept of edge computing [4] to improve the overall availability and extensibility of the system by pushing data to the edges of the Internet. Edge computing can effectively solve the problem of big data processing in the IoT [5]. Edge computing is a kind of computing mode [6]. Compared with

the data-centralized cloud computing model, edge computing processes data at the network edges. Functional entities with the capabilities of application, storage, and computing between data sources and cloud data centres can serve as network edges. Qian et al. [7] proposed a workflow-aided Internet of Things (WIoT) paradigm with intelligent edge computing (iEC) to automate the execution of IoT applications with dependencies. Their design primarily targeted at reducing the latency from two perspectives, including IoT application perspective and edge computation perspective. Liao et al. [8] provided a promising paradigm to support the implementation of industrial IoT by offloading computational-intensive tasks from resource-limited machine-type devices (MTDs) to powerful edge servers. In addition, Bonomi et al. [9] proposed the concept of fog computing which bridges the gap between the cloud and IoT devices by enabling computing, storage, networking, and data management to the network nodes within the close vicinity of IoT devices. Compared with fog computing, edge computing lays more emphasis on the coordination of resources among edge devices. Edge computing is a technology for processing upstream data of cloud or downstream data of IoT [5]. Edge computing provides a theoretical basis for the implementation of fog computing services [10].

Edge computing for Internet of Things (EC-IoT) is an emerging research field in recent years. Now, it is in a period of rapid development, and its hotspots are constantly emerging. Li et al. [11] proposed a design scheme of intelligent building gateway based on edge computing and priority classification for the problems of various types of equipment, inconsistent communication protocols, large data communication volume, and poor real-time performance in the field of large buildings. Zhang et al. [12] established an industrial edge network model and proposed a new cache replacement algorithm based on combing Persistence Prediction and Size Caching Strategy (PPPS). This algorithm effectively improved the hit ratio and cache utilization efficiency of edge cache and reduced the delay of user request files. Du et al. [13] proposed optimization strategies in NOMA-based vehicle edge computing network, which could effectively reduce the cost of task processing under the premise of guaranteeing the maximum delay. Yi et al. [14] proposed a vehicle-adaptive interest packet routing scheme based on relationship strength and interest degree for both inter-/intracommunities which combined the idea of edge computing. Although some breakthroughs have been made in some application fields, the development of EC-IoT has been hampered by some problems such as fragmentation, insecurity, and increased network load. Therefore, it is urgent to conduct a systematic analysis of the knowledge system of EC-IoT, analyze the key technologies, and discuss the future development trends and hotspots.

In this paper, a bibliometric study of EC-IoT literature was conducted with the aim of revealing some valuable insights to the active scholars and practitioners in the EC-IoT field. In terms of scientific impact, bibliometrics research has been widely used to analyze trends and identify emerging scientific fields [15–19].

The results extracted from the bibliometric study presented in this paper included (1) the top 10 most cited literatures; (2) the most popular journals; (3) the most productive authors; (4) annual literatures and citation trends; (5) main research institutions; (6) the most published countries/regions; and (7) hotspots.

This paper is organized as follows: in Section 2, the data sources, main methods, and research questions in this paper were introduced. In Section 3, core literatures, core journals, core authors, overall growth, main research institutions, notable countries/regions, and hot spots related to EC-IoT from WoS core collections were studied in detail. In Section 4, a brief conclusion was given, and a brief summary of the future development of this field was made.

2. Data and Methods

2.1. Description of Data Source. The literature data of this paper came from the Sci-Expanded database of Web of Science (WoS) Core Collection of the Institute for Scientific Information (ISI). The knowledge development system of EC-IoT was expected to be fully understood, and this field has been around for about 20 years, so the time frame is limited to 2000–2020, which covers almost the entire period of large-scale scientific production in this field. By December 16, 2020, 13,498 references were retrieved in WoS Core Collection datasets under the theme of “edge computing” or “mobile computing.” The retrieved literature studies were then further refined under the theme “IoT” or “Internet of things” or “smart Home” or “Smart City” or “Industrial Automation” or “Connected vehicles.” At this time, a total of 2732 literature studies were retrieved. The complete records and references of these literature studies were exported as the dataset.

2.2. Research Methods. Based on bibliometric analysis, the main literature studies, journals, authors, research institutions, countries/regions, and keywords of EC-IoT were statistically studied. Bibliometric analysis is based on mathematics and statistics to quantitatively analyze scientific literature studies published in a specific field of knowledge [20]. The status quo and development trend of science and technology are described, evaluated, and predicted to a certain extent, and the current research status and frontiers of the discipline are reflected [21]. Through bibliometric analysis, the development history, research hotspots, and future directions of EC-IoT were explored.

In recent years, knowledge mapping tools are used in bibliometrics research to transform the table analysis of written data into visual maps that are more visual and easier to read. Visualization of Similarities Viewer (VOSviewer) is a kind of bibliometric analysis and knowledge visualization software jointly developed by N. J. Van Eck and L. Waltman from the Science and Technology Research Centre of Leiden University in the Netherlands in 2010 [22]. In this paper, the clustering algorithm in VOSviewer was used to carry out co-occurrence analysis of published countries and high-frequency keywords. In addition, cocitation networks of

cited literature, journals, and authors were built for visual analysis of the knowledge maps.

2.3. Research Questions. The current situation in the world raises many questions that need to be answered. In this paper, the following questions need to be answered that will help to identify the dynamics of EC-IoT and provide a holistic means for future research in field. These questions are addressed as follows:

RQ1: what are the most influential literatures of EC-IoT?

RQ2: which journals are the most popular in the EC-IoT field?

RQ3: which authors are leading the EC-IoT study?

RQ4: what is the evolution of EC-IoT research field?

RQ5: what are the main research institutions?

RQ6: what is the research status of EC-IoT in countries/regions around the world?

RQ7: what are the EC-IoT hot spots?

3. Bibliometric Analysis of EC-IOT Literatures

The publication status of literatures is usually regarded as an important indicator to measure the development level of a discipline and the level of scientific research achievements and contributions [23]. Trends in EC-IoT research were studied in this paper using statistical literature studies and the frequencies of citations over time.

3.1. Analysis of Core Literature Studies. The main literature studies on EC-IoT were highlighted. The top 10 most cited literatures in the world are listed in Table 1. These literature studies have had the widest influence in EC-IoT.

On top of the list, Shi et al. [5] first proposed the definition of edge computing in 2016. They analyzed the applications of edge computing in smart homes and cities through case studies and pointed out the opportunities and challenges of edge computing. Their paper had the highest total citations, with a value of 1342. And, the value of its average citations per year is also the highest. This shows that their paper has a strong impact. In the second place of the list, Mao et al. [24] provided a comprehensive survey of the state-of-the-art mobile edge computing (MEC) research. They discussed a series of issues, challenges, and future research directions for MEC research, such as MEC system deployment and cache-enabled MEC. Chiang et al. [25] expounded the opportunities and challenges of fog computing in view of the IoT network context. Stankovic et al. [26] presented a vision for how IoT could change the world in the distant future and enumerated eight key research topics. Lin et al. [27] conducted an overview of IoT with respect to system architecture, enabling technologies, security and privacy issues, and presented the integration of fog/edge computing and IoT and applications. Satyanarayanan et al. [28] introduced the emergence of edge computing. Gu et al. [29] proposed a Service-Oriented

Context-Aware Middleware (SOCAM) architecture, aiming to solve the context-aware problems of wireless network and mobile computing. Abbs et al. [30] provided the definition of mobile edge computing, its advantages, architecture, application areas, and future directions. Dastjerdi et al. [31] introduced the advantages of fog computing. Shi and Dustdar [32] analyzed the promise of edge computing.

EC-IoT is an emerging field of research. Edge computing was not formally defined until 2016. Therefore, the majority of the most cited literature studies in the world are summary articles, as shown in Table 1. These articles mainly analyzed and discussed the problems such as the concept, application, challenge, and development prospects of EC-IoT.

3.2. Analysis of Core Journals. In the development of EC-IoT, journals play an important role as the main disseminators of the process of studies. The top 7 leading journals with the most published literatures in the EC-IoT field are listed in Table 2.

According to Table 2, IEEE Access and IEEE Internet of Things Journal are the most popular journals in the EC-IoT field, and they have the greatest number of literature studies published. Among them, IEEE Access has published 279 literature studies in this field, ranking first. Meanwhile, the total number of literature studies published by the journals listed in Table 2 reaches 832, accounting for about 30% of all literature studies retrieved. These journals provide significant supports for research and development in the EC-IoT field.

Next, the journal citation totals [33, 34] were studied, that is, the most cited journal and the journal frequently cited by the same source. The minimum number of citations of journals was set 50 in VOSviewer. A visualization of journal cocitation network is shown in Figure 1. In Figure 1, the sizes of dots and words represent the cited times. The larger the dots and words are, the more times they are cited. It can be seen from Figure 1 that IEEE Access and IEEE Internet of Things are cited the most times, indicating that these two journals have very strong influences in the EC-IoT field. Besides, the sensor with the third place in Table 2 has been cited much less often.

3.3. Analysis of Core Authors. The core authors are the most productive authors. They are researchers who have published many literature studies in a certain research field. Studying the core authors is conducive to analyzing and finding authoritative EC-IoT experts. Price's law in bibliometrics can be used to determine the core author in a research field [5]. Place's law pointed out the core authors, which can be expressed as follows:

$$M = 0.749\sqrt{N_{\max}}, \quad (1)$$

where N_{\max} is the maximum number of literature studies published by the same author and M is the minimum number of literature studies published by the core authors. The authors who published more than M literature studies are the coauthors. As shown in Table 3, the maximum

TABLE 1: The top 10 most cited literatures.

Number	Literature title	Total citation	Average citation per year
1	Edge Computing: Vision and Challenges [5]	1342	268.4
2	A Survey on Mobile Edge Computing: The Communication Perspective [24]	836	209
3	Fog and IoT: An Overview of Research Opportunities [25]	801	160.2
4	Research Directions for the Internet of Things [26]	796	113.71
5	A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications [27]	598	149.5
6	The Emergence of Edge Computing [28]	442	110.5
7	A Service-Oriented Middleware for Building Context-Aware Services [29]	438	27.38
8	Mobile Edge Computing: A Survey [30]	417	139
9	Fog Computing: Helping the Internet of Things Realize Its Potential [31]	342	68.4
10	The Promise of Edge Computing [32]	325	65

TABLE 2: The top 7 leading journals with the most published literatures.

Number	Journal title	Number of literatures published	Impact factor (2019)
1	IEEE Access	279	3.74
2	IEEE Internet of Things Journal	251	9.51
3	Sensors	98	3.27
4	Future Generation Computer Systems: The International Journal of eScience	72	5.38
5	IEEE Transactions on Industrial Informatics	50	9.11
6	IEEE Network	44	7.50
7	IEEE Communications Magazine	38	11.05

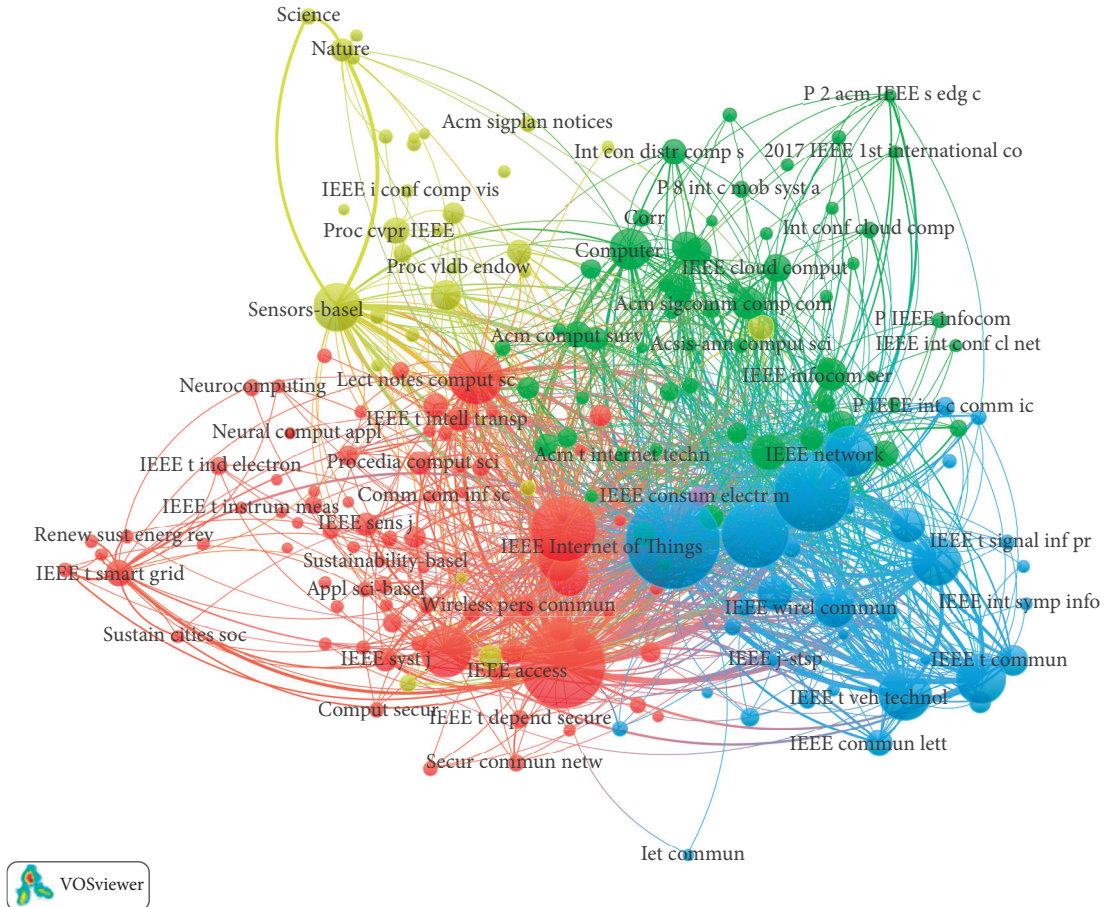


FIGURE 1: A visualization of journal cocitation network.

TABLE 3: The top 10 authors with the most published literature studies.

Number	Author name	Number of literature studies published
1	Y. Zhang	27
2	J. Zhang	23
3	Y. Chen	22
4	X. Chen	21
5	J. J. Liu	19
6	J. J. P. C. Rodrigues	18
7	S. Dustdar	17
8	Y. Jararweh	16
9	Y. Liu	16
10	M. Villari	15

number of literature studies published by the same author in the EC-IoT field is 27. Then, $M = 4$. Therefore, the authors who have published more than 4 literature studies are the coauthors in this field, totaling 466. The number of literature studies published by these 466 core authors accounted for 88.61% of all published literature studies.

According to Table 3, Zhang is the most productive author. However, scientometrics have done a great deal of work on how to meaningfully quantify the publication of academic results. They believe that counting the number of literatures is one way, and that counting the total number of citations is considered the other way that is more meaningful [35].

The minimum number of citations of the authors was set 50 in VOSviewer. Then, of the 36638 authors, 155 meet this threshold. A density visualization of author cocitation network is shown in Figure 2. If the color is lighter and the words are larger, the author's number of citations is higher. As can be seen in Figure 2, M. Satyanarayanan, W. Shi, F. Bonomi, and X. Chen have the highest authors' number of citations. This indicates that their work is recognized by many researchers and has a strong impact in the EC-IoT field.

3.4. Analysis of the Overall Growth Trend. When downloading data from the WoS Core Collection database, the time range was set from 2000 to 2020. However, the first EC-IoT literature retrieved was in 2005 [29]. Therefore, the data we used were from 2005 to 2020. Number of literature studies and total number of citations by year are shown in Figure 3.

As can be seen from Figure 3, number of literature studies and total number of citations generally show an upward trend in the EC-IoT field. In particular, their growth accelerated rapidly after 2017, almost exponentially. This may be because the definition of edge computing had not been proposed before 2016, and the research on EC-IoT was still on the exploration stage. Shi et al. [5] defined edge computing for the first time in 2016, which attracted the attention of many researchers. Therefore, EC-IoT entered a period of rapid development after 2017. The number of EC-IoT literature studies reached a maximum of 947 in 2019 but fell slightly to 783 in 2020. This may be due to the impact of COVID-19 on a global scale. Nevertheless, the total number of citations in EC-IoT has been on the rise. As such, EC-IoT

is in the explosion stage, and future research in this field is likely to continue for a long time. With the booming development of edge computing, EC-IoT will be urgently needed to be applied in various fields such as smart home, smart city, industrial automation, and connected vehicles. At the same time, the rapid development of IoT is also driven by the practical problems in various industries.

3.5. Analysis of Main Research Institutions. From 2005 to 2020, Beijing University of Posts Telecommunications has published 115 literature studies, accounting for 4.21% of all literature studies in this field. Beijing University of Posts Telecommunications ranks first in the number of published literature studies of EC-IoT. Xidian University ranks second, with 52 published literature studies. It was followed by University of Electronic Science and Technology of China and King Saud University. The top 10 research institutions with the most published literature studies are shown in Figure 4.

As can be seen from Figure 4, the top 10 research institutions are dominated by universities, and Chinese research institutions are the majority. China's research and development in EC-IoT has reached a certain scale. Besides, King Saud University in Saudi Arabia and the University of Messina in Italy are also among the top 10 research institutions. This indicates that these two research institutions are also concerned about EC-IoT.

3.6. Analysis of Notable Countries/Regions. Considerable efforts have been made to promote the development of EC-IoT to generate knowledge that can be used to solve problems encountered in practical applications of IoT. The top 10 most published countries are shown in Figure 5. It can be seen from Figure 5 that China is the country with the highest productivity, which produced 931 literature studies in total, accounting for 34.08% of the total. It is followed by USA, which produced 576 literature studies, accounting for 21% of the total. Then comes Italy, England, and South Korea. Many literature studies in the EC-IoT field have been published by 10 countries in Figure 5, which provide a good foundation for this work. This gives these countries a leading position in research and a better opportunity and development prospect in the future development process of IoT applications.

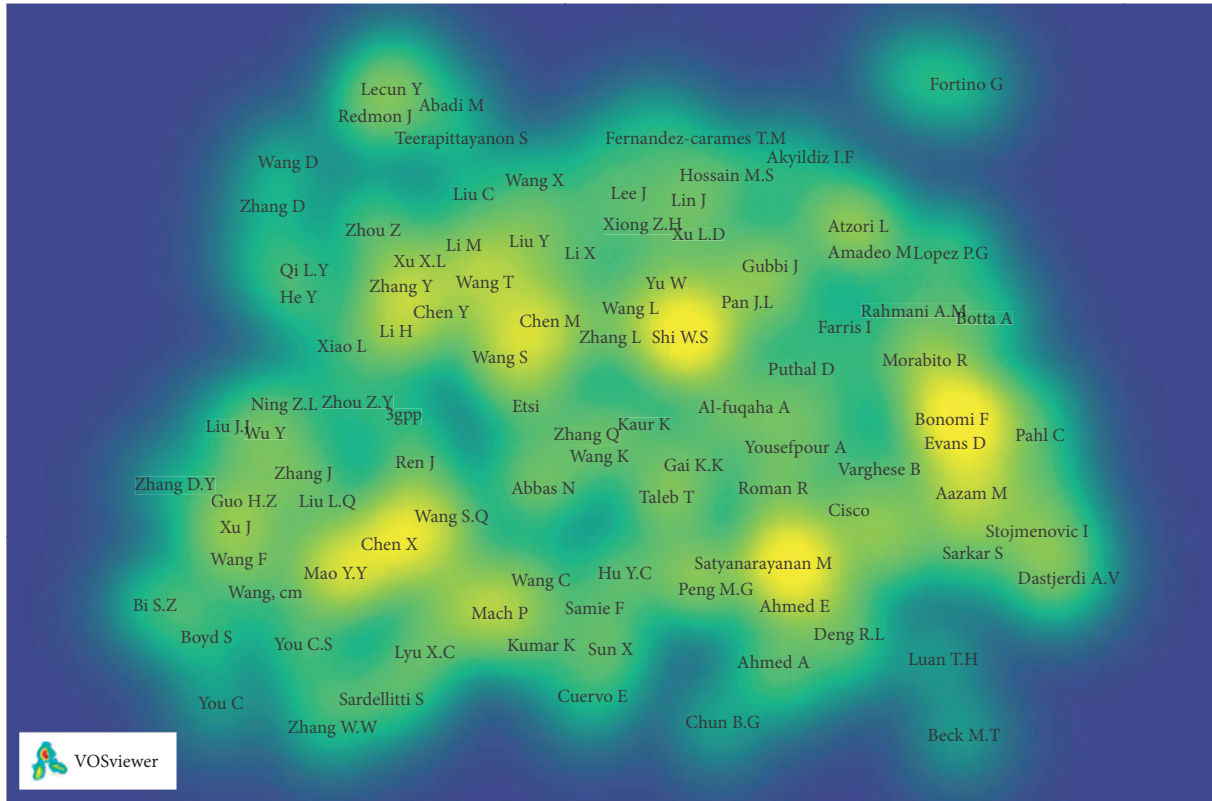


FIGURE 2: A density visualization of author cocitation network.

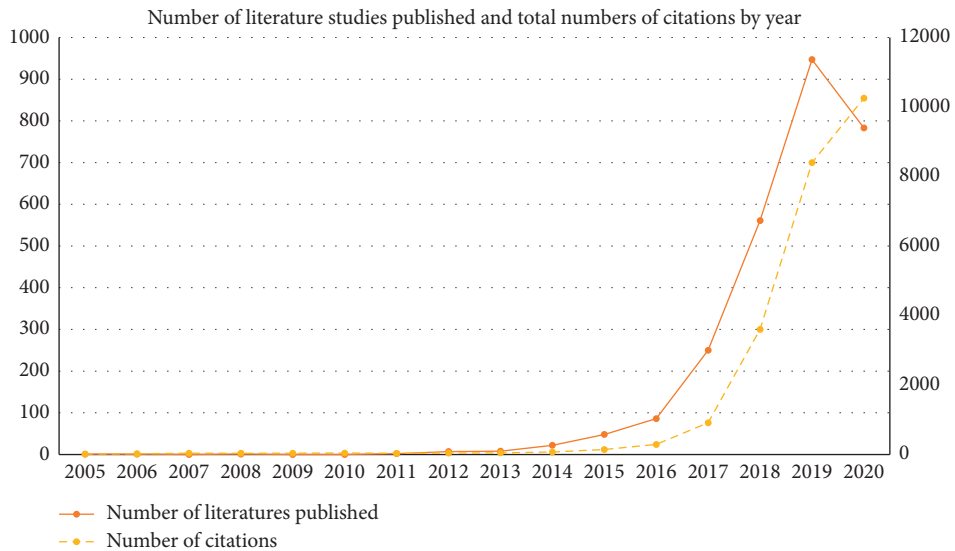


FIGURE 3: Number of literature studies published and total number of citations by year.

Next, the spatial distribution of the literature studies published is discussed. A geographic visualization of the literatures published is conducive to the clear understanding of the geographic output distribution of literature studies, so as to help researchers further understand the overview of scientific research achievements of EC-IoT. A geographic visualization of research co-occurrence network is shown in Figure 6. According to Figure 6, countries all over the world

attach great importance to EC-IoT, and there are three research intensive regions in the world, namely, Europe, southeast North America, and Southeast Asia.

3.7. Analysis of Hotspots. Keywords are an important part of the literature, which highly condense the content of the literature. The co-occurrence network analysis of keywords

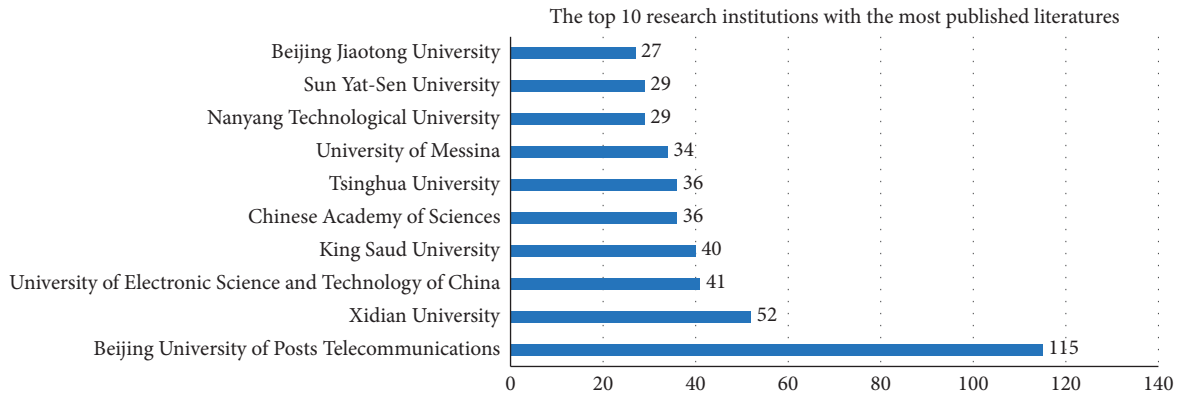


FIGURE 4: The top 10 research institutions with the most published literature studies.

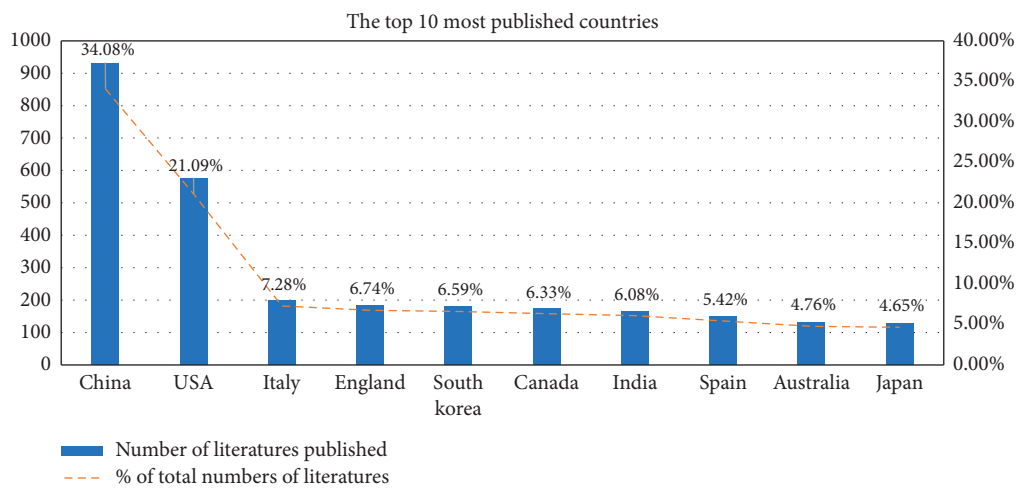


FIGURE 5: The top 10 most published countries.

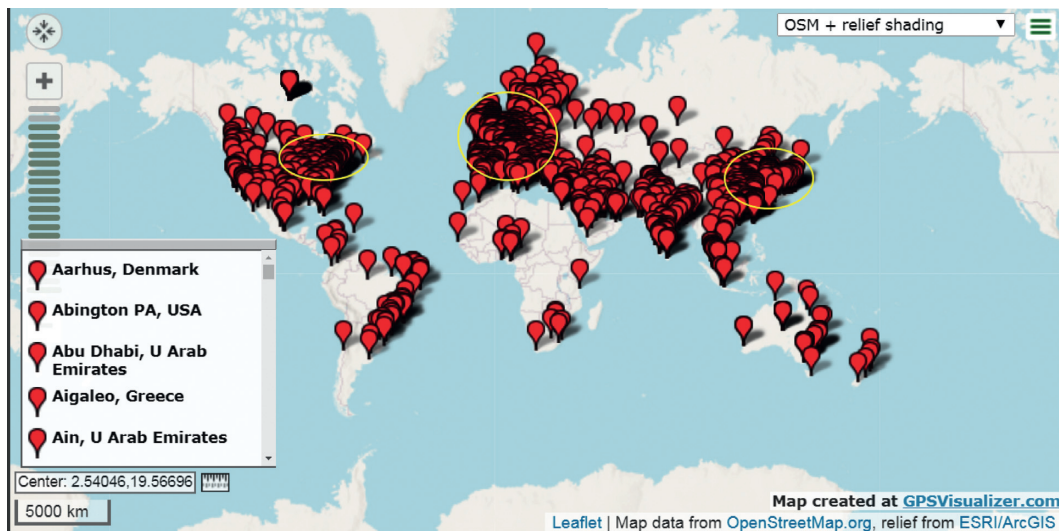


FIGURE 6: A geographic visualization of the research co-occurrence network.

can effectively reflect the research hotspots in the subject area. The minimum number of occurrences of a keyword was set 40 in VOSviewer. After merging EC-IoT and its

synonyms, four clusters of high-frequency keywords were obtained, whose nodes of the same color belong to the same cluster. A visualization of keywords co-occurrence network

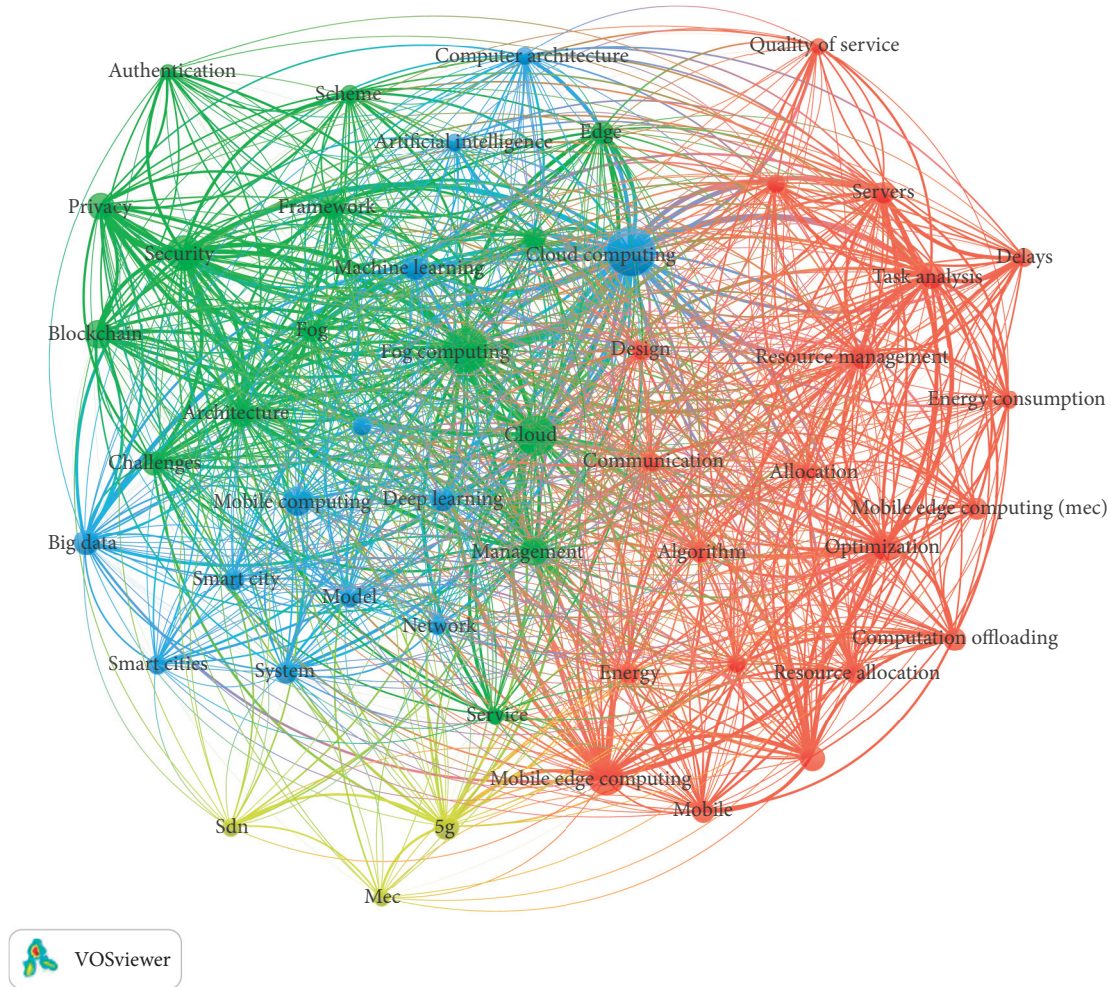


FIGURE 7: A visualization of keywords co-occurrence network.

is shown in Figure 7. Keywords of “edge computing,” “Internet of Things,” “Internet,” and “IoT” are not shown because literature studies retrieved all related to “edge computing” and “Internet of Things.”

Cluster 1: cluster 1 is the study of specific difficult problems of EC-IoT, as shown in the red node region. The specific problems of EC-IoT include resource management, resource allocation, computation offloading, energy consumption, and delay.

Cluster 2: cluster 2 is the study of the overall architecture of EC-IoT, as shown in the green node region. Through the optimization and innovation of the overall architecture, the protection of privacy and security will be further increased.

Cluster 3: cluster 3 is the study of EC-IoT applications, as shown in the blue node region. Mobile computing is combined with big data technology and artificial intelligence algorithms to make EC-IoT better applied in smart homes, smart city, and some other fields.

Cluster 4: cluster 4 is the study of mobile edge computing (MEC) with 5G, as shown in yellow node area. The development of 5G ensures high bandwidth and

low latency in the transmission process, which provides the network security for EC-IoT. With the continuous maturity of 5G technology, the study of MEC will also usher in rapid development.

Research hotspots of EC-IoT are understood through the statistics of keywords with frequent co-occurrence. In summary, it includes four aspects: specific difficult problems, overall architecture, applied research and joint research with 5G.

The literature keyword analysis not only provides an effective way for the knowledge structure of the research field but also provides an effective way for the exploration of the development trend in the field. Therefore, it can be speculated that problems of EC-IoT related to resource management, resource allocation, computation offloading, energy consumption, and delay will be further studied in the future. Meanwhile, with the gradual maturity of EC-IoT technology and 5G, EC-IoT will be further promoted and popularized in smart cities, smart transportation, and some other applications.

4. Conclusions

Significant influential aspects of EC-IoT literatures were studied in this paper. It can be summarized as follows: the

most influential literature in the world was written by Shi who first proposed the definition of edge computing. IEEE Access and IEEE Internet of Things are the two leading journals that have published most literature studies and are cited most times. According to Price's Law, there are 466 EC-IoT core authors in the world, among which Zhang is the most productive author, and Satyanarayanan is the most cited author. On the whole, the number of EC-IoT literature studies published and literature studies cited are on the rise. In particular, they showed an exponential growth trend after 2017. The future research of EC-IoT will continue for a long time. Beijing University of Posts Telecommunications ranks first in the number of published literature studies of EC-IoT. Countries all over the world attach great importance to EC-IoT, and there are three research intensive regions in the world, namely, Europe, southeast North America, and Southeast Asia. China and USA are the most published countries. In addition, the research hotspots of EC-IoT mainly focus on four aspects, including specific difficult problems, overall architecture, applied research, and joint research with 5G. In the future, with the constant maturity of EC-IoT technology and 5G technology, EC-IoT will be further promoted and popularized in smart cities, smart transportation, and some other applications.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Research Program Foundation of Minjiang University (no. MJW201831408).

References

- [1] J. Tian, "Analysis on application of internet of things in Chengdu broadcasting network," *Telecom World*, vol. 26, no. 5, pp. 35–36, 2019.
- [2] M. H. Alshaif, A. H. Kelechi, K. Yahya, and S. A. Chaudhry, "Machine learning algorithms for smart data analysis in Internet of things environment: taxonomies and research trends," *Symmetry*, vol. 12, no. 1, p. 88, Jan. 2020.
- [3] Z. Li and X. Zhang, "Resource allocation and offloading decision of edge computing for reducing core network congestion," *Computer Science*, vol. 48, no. 3, pp. 281–288, 2020.
- [4] H. H. Pang and K. L. Tan, "Authenticating query results in edge computing," in *Proceedings. 20th International Conference on Data Engineering*, pp. 560–571, IEEE, Boston, MA, USA, April 2004.
- [5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [6] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [7] Y. Qian, L. Shi, J. Li et al., "A workflow-aided Internet of things paradigm with intelligent edge computing," *IEEE Network*, vol. 34, no. 6, pp. 92–99, Dec. 2020.
- [8] H. Liao, Z. Zhou, X. Zhao et al., "Learning-based context-aware resource allocation for edge-computing-empowered industrial IoT," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4260–4277, May 2020.
- [9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, pp. 13–15, Association for Computing Machinery, Helsinki, Finland, August 2012.
- [10] L. Wang, C. Wu, and W. Fan, "A survey of edge computing resource allocation and task scheduling optimization," *Journal of System Simulation*, vol. 33, no. 3, pp. 509–520, 2020.
- [11] L. Li, H. Jin, X. Sun, and L. Li, "Design of intelligent building gateway based on edge computing and priority classification," *Modern Electronics Technique*, vol. 44, no. 6, pp. 67–71, 2021.
- [12] L. Zhang, L. Li, H. Chen, and B. Daniel, "A cache replacement algorithm for industrial edge computing application," *Journal of Computer Research and Development*, vol. 1, 2021.
- [13] J. Du, N. Xue, Y. Sun et al., "Optimization strategies in NOMA-based vehicle edge computing network," *Chinese Journal on Internet of Things*, vol. 5, no. 1, pp. 1–8, 2021.
- [14] B. Yi, X. Wang, and M. Huang, "Content delivery enhancement in Vehicular Social Network with better routing and caching mechanism," *Journal of Network and Computer Applications*, vol. 177, 2021.
- [15] T. U. Daim, G. Rueda, H. Martin, and P. Gerdri, "Forecasting emerging technologies: use of bibliometrics and patent analysis," *Technological Forecasting and Social Change*, vol. 73, no. 8, pp. 981–1012, 2006.
- [16] L. Huang, Y. Zhang, Y. Guo, D. Zhu, and A. L. Porter, "Four dimensional science and technology planning: a new approach based on bibliometrics and technology roadmapping," *Technological Forecasting and Social Change*, vol. 81, pp. 39–48, Jan. 2014.
- [17] A. Moro, E. Boelman, G. Joanny, and J. L. Garcia, "A bibliometric-based technique to identify emerging photovoltaic technologies in a comparative assessment with expert review," *Renewable Energy*, vol. 123, pp. 407–416, 2018.
- [18] M. Sergio, L. M. Lopez, L. R. Africa, C. Joaquin, M. P. Alexis, and C. Manuel, "Analysis of new technology trends in education: 2010–2015," *IEEE Access*, vol. 6, pp. 36840–36848, 2018.
- [19] K. Hu, H. Wu, K. Qi et al., "A domain keyword analysis approach extending Term Frequency-Keyword Active Index with Google Word2Vec model," *Scientometrics*, vol. 114, no. 3, pp. 1031–1068, 2018.
- [20] S. Pablo, J. C. Manuel, D. L. H. Carlos, I. P. Jose, and H. Enrique, "Opinion mining, sentiment analysis and emotion understanding in advertising: a bibliometric analysis," *IEEE Access*, vol. 8, pp. 134563–134576, 2020.
- [21] Y. Song, Y. Wu, and D. Fan, "Knowledge mapping of three-dimensional printing in biomedical field based on VOSviewer," *China Academic Journal Electronic Publishing House*, vol. 25, no. 15, 2021.
- [22] J. Li, *Overview of VOSviewer and CitNetExplorer*, in *Principles and Applications of Mapping Knowledge Domains*, Higher Education Press, Beijing, China, 1st edition, 2020.

- [23] H. Chen and Z. Deng, "Bibliometric analysis of the application of convolutional neural network in computer vision," *IEEE Access*, vol. 4, pp. 1–12, 2016.
- [24] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [25] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
- [26] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.
- [27] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of things: architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 125–1142, 2017.
- [28] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
- [29] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," *Journal of Network and Computer Applications*, vol. 28, no. 1, pp. 1–18, 2005.
- [30] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
- [31] A. V. Dastjerdi and R. Buyya, "Fog computing: helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.
- [32] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
- [33] F. J. Martínez-López, J. M. Merigó, L. Valenzuela-Fernández, and C. Nicolás, "Fifty years of the European journal of marketing: a bibliometric analysis," *European Journal of Marketing*, vol. 52, no. 1/2, pp. 439–468, 2018.
- [34] A. Tur-Porcar, A. Mas-Tur, J. M. Merigó, N. Roig-Tierno, and J. Watt, "A bibliometric history of the journal of psychology between 1936 and 2015," *The Journal of Psychology*, vol. 152, no. 4, pp. 199–225, 2018.
- [35] B. Wu, F. Ou, Y. Deng et al., "E-Index-a bibliometric index of research efficiency," *IEEE Access*, vol. 6, pp. 51355–51364, 2018.

Research Article

Task Priority-Based Cached-Data Prefetching and Eviction Mechanisms for Performance Optimization of Edge Computing Clusters

Ihsan Ullah ¹, Muhammad Sajjad Khan ^{2,3}, Marc St-Hilaire ⁴, Mohammad Faisal ⁵,
Junsu Kim ³ and Su Min Kim ³

¹Advanced Technology Research Center, Korea University of Technology and Education, Cheonan, Republic of Korea

²Department of Electrical Engineering, International Islamic University, Islamabad, Pakistan

³Department of Electronics Engineering, Korea Polytechnic University, Siheung, Republic of Korea

⁴School of Information Technology and Department of Systems and Computer Engineering Carleton University, Ottawa, Canada

⁵Department of Computer Science and Information Technology, University of Malakand, Chakdara, Pakistan

Correspondence should be addressed to Su Min Kim; suminkim@kpu.ac.kr

Received 24 January 2021; Revised 3 March 2021; Accepted 14 March 2021; Published 24 March 2021

Academic Editor: Chien-Ming Chen

Copyright © 2021 Ihsan Ullah et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The rapid evolution of the Internet of Things (IoT) and the development of cloud computing have endorsed a new computing paradigm called edge computing, which brings the computing resources to the edge of the network. Due to low computing power and small data storage at the edge nodes, the task must be assigned to the computing nodes, where their associated data is available, to reduce overheads caused by data transmissions in the network. The proposed scheme named task priority-based data-prefetching scheduler (TPDS) tries to improve the data locality through available cached and prefetching data for offloading tasks to the edge computing nodes. The proposed TPDS prioritizes the tasks in the queue based on the available cached data in the edge computing nodes. Consequently, it increases the utilization of cached data and reduces the overhead caused by data eviction. The simulation results show that the proposed TPDS can be effective in terms of task scheduling and data locality.

1. Introduction

Edge computing is a paradigm to extend cloud computing services to those at edge nodes in networks. Thus, it brings the computing services near to Internet of things (IoT) devices [1]. Putting resources at the edge of the network enables achieving low latency processing. However, since the enormous number of IoT devices generates a high volume of data, transmitting them to the cloud yields high computational processing. In general, the cloud contains distributed computing resources and processes the data using a group of servers in parallel and distributed way. Sending all the data and tasks to the cloud for processing makes the core network congested and yields a huge load to the cloud servers. To minimize the workload of the core network and the cloud, novel paradigms such as edge

computing and fog computing are developed [2–7] to bring computational resources to the edge of the network and offer services near to each IoT device as shown in Figure 1. Due to low computing power and limited data storage, the edge nodes are clustered to perform computation and the huge tasks are distributed to the edge nodes. To distribute the tasks resourcefully and efficiently to the edge nodes based on task-associated data, an efficient task scheduling strategy is required. In other words, a cost-effective task scheduler is needed to assign the tasks closer to the data on a cluster node and bring the resources near to computation nodes while improving the overall system performance.

In cloud computing systems, complicated tasks and data are collected to the cloud for computing processes [8, 9]. All these data and tasks are generated by IoT devices which are connected to the cloud by a middle layer, i.e., IoT edge

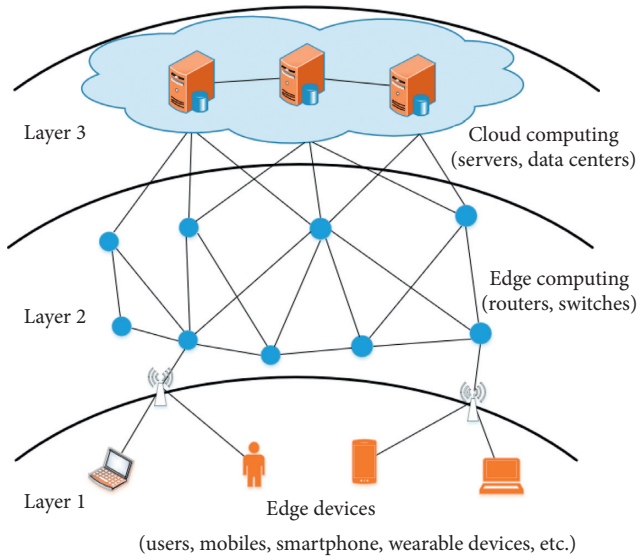


FIGURE 1: The structure of edge computing network.

nodes. Thousands of IoT devices are connected to the cloud which can yield a heavy load to the core network and the cloud system. This increases the frequency of communication exchanges and causes a long latency to the end-users. Consequently, there are resource limitations in a cloud computing layer to incline many researchers toward the computation at edge devices. Data generated by IoT devices can be processed by middle layer devices such as IoT nodes and base stations. The nodes at the edge level retain low processing power and limited resources and, thus, they cannot handle such heavy and complicated tasks. Therefore, a cost-effective task management strategy is needed to distribute the complicated tasks to the edge nodes in an efficient way.

In a cloud computing cluster, a task manager predicts the amount of data at the computing node and assigns the tasks to appropriate targeted nodes to guarantee data locality [10]. Based on this prediction, each node tries to bring and preload the data from other locations. How much the preloaded data match the task depends on the result of the prediction. The wrong prediction will yield the preloaded data, which is not useful for running the task. It implies the wastes of communication bandwidth and system resources. Yet, the preloaded data can be exploited by fetching the associated task from the queue. So far, several scheduling schemes have been proposed to balance the workload in the network based on the amount of available resource and data [11–20]. All these schemes are prefetching the data but they do not consider tasks' priorities concerning the available cached or stored data. On the contrary, our approach in this paper assigns a priority to a task according to availability if the required data can be obtained from the cached-data queue. Consequently, it can reduce the overhead required for task eviction.

On the other hand, in distributed systems [21, 22], fetching a computation task near to data is cheaper than fetching the data near to the computation task. Bringing the

computation task close to the required data is called data locality in cloud computing environments. It is impossible to guarantee 100% data locality but it somehow can be improved with the existing data at the edge level by minimizing unnecessary data transmissions. For quick access, used data are kept in the cache memory for iterative processes. The cache memory contains two different types of data: static data, which is not changeable and can be used in the next round of task execution, and dynamic data, which is changeable and useable in the next round. Due to limited memory capacity, it is impossible to keep all the data needed for the task in the cache memory of the computing node, since data swap-out and swap-in require frequent processes in the cache and storage memories. Loading data from the storage memory to the cache memory is an expensive process in terms of data processing and transferring. If the cache memory becomes full and the system cannot store more data in the cache memory, least recently used (LRU) and first-in-first-out (FIFO) eviction techniques [23] can be applied to swap out unnecessary old data from the cache memory.

In this paper, we extend the idea from our earlier work [11] to utilize the existing preloaded data effectively based on a cost-effective scheduling strategy, named task priority-based data-prefetching scheduler (TPDS), which distributes the tasks to the computing nodes logically. The proposed TPDS tries to match the task in the queue with the cached-data at the computing node. It generates a priority for a task and allocates the task to a proper edge node based on task-associated data in the cache. With this technique, the frequency of data swapping in the cache can be significantly reduced and the data utilization can be improved for available tasks. If there is no task in the queue for the cached-data, the data is swapped out and replaced by the required new data. In this paper, we employ the multi-server queuing theory [24] to evaluate the performance of the proposed scheduling strategy. The proposed TPDS achieves better performance in terms of data locality, task distribution, and reduction of system overheads caused by unnecessary evictions and data exchanges. The main contributions of this paper are summarized as follows:

- (i) Dynamic workload scheduling considering queue-wise job priorities is proposed based on data locality of the cache memory in order to maximize the resource efficiency and the data utilization of a cloud cluster.
- (ii) In the cloud cluster, our proposed scheme prefetches and evicts the cached-data from the computing node based on task priority. It is able to avoid blind eviction of the cached-data and reduce the system overhead. Hence, it improves the resource efficiency at each node.
- (iii) Through assigning a task to the computing node based on the data locality, we can minimize the average completion and waiting time for each task.
- (iv) A multi-server queuing model applicable to the proposed TPDS scheme is developed in order to

improve schedulability of the tasks under different constraints and requirements.

The rest of the paper is organized as follows. In Section 2, we review the related previous work concerning scheduling considering prefetching and data locality. We propose a scheduling strategy based on priority-based data-prefetching in Section 3. In Section 4, we evaluate the performance of the proposed strategy, compared to the conventional ones. Finally, this paper is concluded in Section 5.

2. Related Work

Many data locality schemes for task scheduling have been developed to improve the performance of the computing system regarding task execution. The data locality enables avoiding unnecessary data transmissions for the task in cloud computing. In distributed cloud system, tasks are assigned to the nodes in the network based on the prediction of associated data [25].

In [26], a new caching algorithm, called similarity-aware popularity-based caching (SAPoC), is proposed to promote the performance of wireless edge-caching by utilizing the similarity among contents in dynamic scenarios. It is developed for dynamic wireless edge-caching scenarios, where both mobile devices and contents arrive and leave dynamically. In SAPoC, a content's popularity is determined by not only its history of the requests but also its similarity with existing ones to enable a quick-start of newly arrived contents. It aims to devise an efficient edge-caching strategy considering the dynamic nature of wireless edge computing systems.

In [10], data locality aware workflow scheduling (D-LAWS), which focuses on data locality, data transfer time based on network bandwidth, virtual machine (VM) consolidation, and fairness of workflow scheduling at the node level, is proposed. The D-LAWS maximizes resource utilization and parallelism of tasks and analytically formulates data transfer time between VMs. It combines VMs and considers task parallelism by using data flow while planning task executions for a data-intensive scientific workflow. Moreover, it reflects more complex workflow models and the data locality regarding data transfer before task executions. In [27], the authors proposed a novel scheduling scheme for real-time bag-of-tasks jobs that arrive dynamically at a hybrid cloud. It takes into account end-to-end deadlines of the jobs, as well as monetary cost required for use of the complementary public cloud resources. In [28], a novel hierarchical architecture for multiple cloudlets is proposed for mobile edge clouds. In this work, the authors target improving the efficiency of cloud resource utilization by organizing the edge cloud servers into a hierarchical architecture. Instead of serving mobile users directly using a flat collection of edge cloud servers, the basic idea of the proposed scheme is to opportunistically aggregate the mobile loads and send the peak loads exceeding the capacities of edge cloud servers at lower tiers to other servers at higher tiers in the edge cloud hierarchy. They developed analytical models to compare the performance between flat and hierarchical designs of edge computing in terms of resource utilization efficiency. Also, they provided theoretical

results that show the advantages of the proposed hierarchical edge cloud architecture.

In [29], Raicu et al. implemented regulating data locality and resource utilization. In [30], the authors proposed a cache-aware task scheduling (CATS) technique that finds suitable resources for executing the data-intensive workload. The proposed model minimizes energy consumptions for both core network and cache accesses. The CATS model brings good tradeoffs between energy minimization and execution time reduction by employing accurate analytical models. Similarly, to enhance the data locality and replication technique, a delay scheduling scheme, called delay scheduling based replication algorithm (DSBRA), is presented in [31]. The DSBRA tries to replicate and de-replicate blocks of the data based on prior information taken from the scheduler. This algorithm focuses on block-level replication but some blocks are stored on the least loaded nodes and some blocks are stored on the heavily loaded nodes. In [32], a locality-based data scheduling algorithm 1 is proposed. It allocates the input data blocks to proper nodes based on their processing capacity in order to enhance the performance of MapReduce in heterogeneous Hadoop clusters.

The prefetching technique is a smart approach for reducing the extra-overhead of data traffic in distributed computing systems. By applying this technique, the delay consumed for task execution can be reduced due to the presence of preloaded data for the task. However, prefetching and predicting data to be preloaded based on the scheduled tasks become a great challenge. In [31, 32], the authors present how to enhance the prefetching techniques and also focus on task scheduling for TaskTracker based on the data. The above-mentioned prefetching strategy maximizes the data locality in distributed computing environments.

Our approach in this paper is based on these previous studies which take into account prefetching to efficiently reuse existing cache data. The main focus in the proposed approach is data eviction and confirmation before task assignment. Our goal is to improve the data locality and to guarantee the resourceful task scheduling in edge computing environments. In the next section, we present the proposed scheduling strategy which enhances the performance of data preloading for tasks and reduces the frequency of cached-data removal blindly. According to our proposed approach, the task scheduler tries to select the most appropriate node in the edge computing cluster from the perspective of the data locality and to assign the task to the selected node. It is able to increase the cached-data utilization and enhance the swapping process for minimizing the overall system overhead.

3. Proposed Task Priority-Based Data-Prefetching Scheduler (TPDS)

In this section, the proposed TPDS is presented for edge computing clusters. The TPDS tries to avoid unnecessary eviction of data in order to improve the operation process of task scheduling and data caching. Since the costs of data transfer and eviction result in a great impact on system

performance, the proposed TPDS attempts to reduce the costs for data transfer and eviction, while it tries to improve the task execution procedure.

3.1. Design Goals. The design goals of the proposed strategy are (i) prioritization of tasks based on the existing data in the cache memory of the computing node, (ii) improvement of awareness between the computing nodes and a task manager regarding data and task to increase hit ratio of the cached-data, and (iii) speeding up the execution of tasks by reducing the waiting time of jobs and increasing the utilization of the cached-data. Let us consider a set of tasks $T = \{t_1, t_2, t_3, \dots, t_n\}$ with the associated data set $D = \{d_1, d_2, d_3, \dots, d_n\}$ and edge computing nodes $E = \{e_1, e_2, e_3, \dots, e_n\}$, which contain different data blocks d_n in the cache memory, C , or storage, S . Based on the traditional data locality scheme, the task $t_n \in T$ will be assigned to the computing node $e_n \in E$ which contains its required data, d_n . Then, task allocation to the node can be expressed as

$$t_n \longrightarrow e_n \begin{cases} S_{e_n \in E} \exists d_n, \\ or \\ S_{e_n \in E} \leftarrow R_{L_n} \exists d_n, \end{cases} \quad (1)$$

where R_{L_n} denotes any remote location, which contains data d_n near to node e_n .

We assume that five tasks arrive in the system as shown in Figure 2. The details of task allocation to the computing node, $e_n \in E$, are given in Table 1. The task t_1 is assigned to the computing node e_1 since the cached-data of the node, $C_{e_1 \in E}$, have the data block d_8 that is needed for the processing of t_1 . Similarly, the task t_2 needs d_2 which is unavailable in the cache of the n -th node, $C_{e_n \in E}$, but available in the storage $S_{e_2 \in E}$ of the node e_2 . By the LRU cache replacement policy, d_0 , which is the old data block, is swapped with d_2 . Similarly, the task t_3 needs the data block d_3 , which considers an old data block is replaced with d_1 as shown in Figure 2.

In Figure 2, it is noted that there are two data blocks d_0 and d_3 which are replaced with d_2 and d_1 by the LRU policy for the tasks t_2 and t_3 , respectively, due to the limited capacity of the cache memory. After finishing, the tasks t_2 and t_3 , and the data blocks d_0 and d_3 will shift again to the cache $C_{e_n \in E}$ for the tasks t_4 and t_5 , which require them. Therefore, the proposed scheduling strategy avoids such unnecessary eviction and swapping of data by prioritizing the tasks based on the available cached-data in the computing node $C_{e_n} \exists d_n$ as shown in Table 2 and Figure 3. Equations (2) and (3) express the computing node and task allocation based on the availability of cached-data.

$$e_n \in E = \forall e_n (C_{e_n \in E}, S_{e_n \in E}), \quad (2)$$

$$\forall t_n \longrightarrow \forall e_n \begin{cases} C_{e_n \in E} \exists d_n, \\ S_{e_n \in E} \exists d_n, \\ S_{e_n \in E} \leftarrow R_{L_n} \exists d_n. \end{cases} \quad (3)$$

4. Performance Evaluation Model

In this section, a theoretical model of the proposed TPDS is formulated and derived. We employ an $M/M/c$ queuing model to evaluate the performance of the proposed TPDS. Suppose that there are n number of tasks denoted by $T = \{t_1, t_2, t_3, \dots, t_n\}$ with a set of data blocks denoted by $D = \{d_1, d_2, d_3, \dots, d_n\}$ and a set of computing nodes denoted by $E = \{e_1, e_2, e_3, \dots, e_m\}$. Here, e denotes the computing nodes, m represents the total number of computing nodes, D represents the set of data blocks, and d_n denotes the specific data block required for a task. If all tasks arrive in the system, the total number of data blocks contained in the cache for all computing nodes can be expressed as

$$T = \sum_{i=1}^n t_i, \quad (4)$$

$$D = \sum_{i=1}^m \sum_{j=1}^n d_j. \quad (5)$$

According to the proposed TPDS, before eviction of the data $d_n \in D$ from the cache memory $C_{e_n \in E}$, the computing node sends a request to the task manager in order to know if there is any task $t_n \in T$ in the queue for this eviction of the data $d_n \in D$. If there is a task in the queue of the task manager, then it gives a priority to the task and assigns that task to the node $e_m \in E$. Otherwise, the data d_n is evicted and swapped in the cache memory. To estimate and optimize the probabilistic performance of edge computing nodes, the notations are defined in Table 3.

In this model, we consider two types of tasks: high priority task and low priority task, based on the cached-data as shown in Figure 3. The high priority tasks are the tasks whose required data are already available in the cache memory and low priority tasks are the tasks whose required data are not available in the cache memory of the edge node $e_n \in E$ as follows:

$$t_n = \begin{cases} t_{n < C_{e_n \in E} \exists d_n} > \text{high priority} \\ or \\ t_{n < C_{e_n \in E} \nexists d_n} > \text{low priority} \end{cases} \quad (6)$$

We consider all tasks arriving at the edge computing nodes with the rate of $\lambda \in T$. We assume that the arrival of a

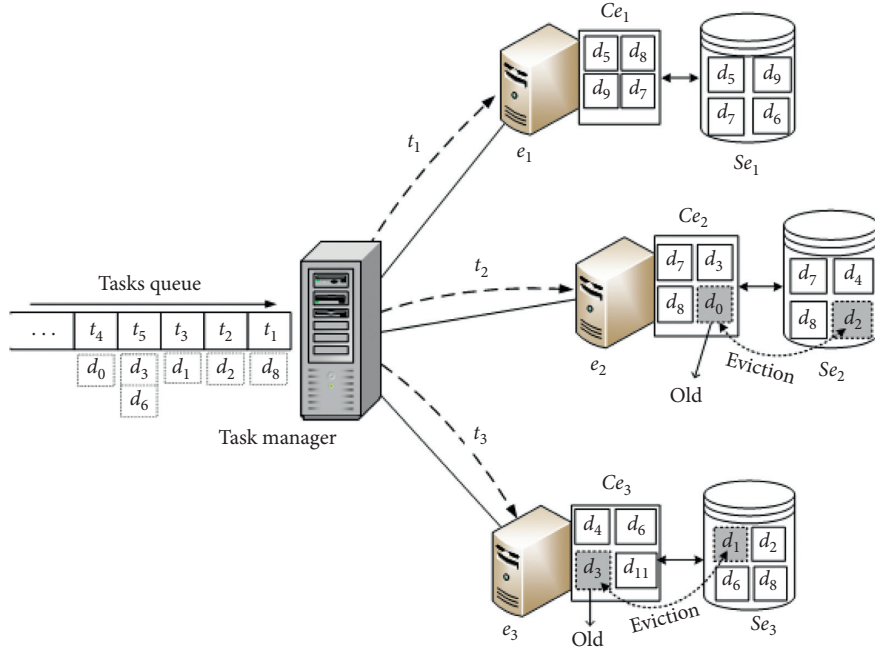


FIGURE 2: Data prefetching and eviction process without task priority.

```

(1) Initialize the values
(2) QT: Queue of tasks in TaskManager.
(3) QD: Record of cached-data in edge node.
(4) QE: List of nodes
(5) Qtn.dn: List of data needed for task execution.
(6) Procedure:
(7) Check status of nodes
(8)  $e_n \leftarrow Idle$ 
(9)  $e_n \leftarrow busy$ 
(10) While (QT is not empty) do
(11)   if ( $e_n$  is idle) then
(12)     for all tasks in queue do
(13)       if  $tn.dn \in C_{e_n \in E}$  then
(14)          $e_n \in E \leftarrow tn \langle h \rangle$ 
(15)       else
(16)         if ( $C_{e_n \in E}$  need eviction) then
(17)            $evcit \leftarrow C_{e_n \in E}.old\_data$ 
(18)            $C_{e_n \in E} \leftarrow S_{e_n \in E} d_n$ 
(19)            $e_n \in E \leftarrow t_n$ 
(20)         end if
(21)       end if
(22)     end for
(23)    $busy \leftarrow e_n$ 
(24) end if
(25) end while

```

ALGORITHM 1: Task priority-based data-prefetching.

task follows a Poisson process and each arrival is transferred to different nodes in the cluster of edge computing nodes. Let $\rho = \lambda/\mu$ be the traffic strength regarding the tasks with different priorities based on the available cached-data, where λ and μ are the arrival rate and the service rate, respectively.

The parameters for task requests in the queuing model are N_s , W_Q , and T_s . Among these three parameters, W_Q affected by the number of tasks being served plays the primary role in the performance. As shown in Figure 4, the scheduling policy is based on $M/M/(e_n \in E)$. According to $M/(e_n \in E)$

TABLE 1: An example of assigning tasks without considerations of priority and data locality.

Arrival of tasks	Required data	Computing nodes
t_1	d_8	$e_1 \ni d_8$
t_2	d_2	$e_2 \ni d_2$
t_3	d_1	$e_3 \ni d_1$
t_4	d_0	$e_2 \ni d_0$
t_5	$d_3 + d_6$	$e_2 \ni d_{3+6}$

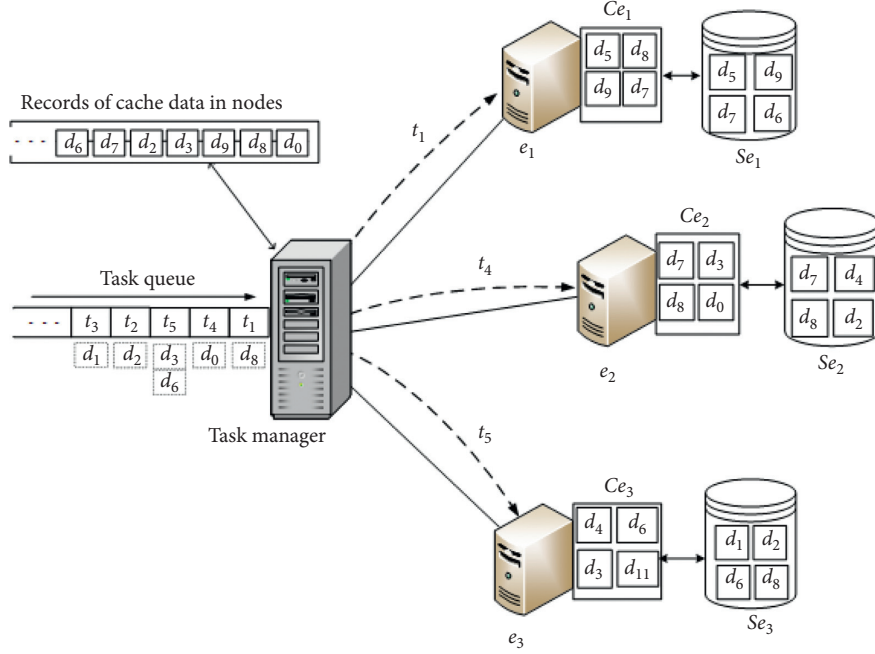
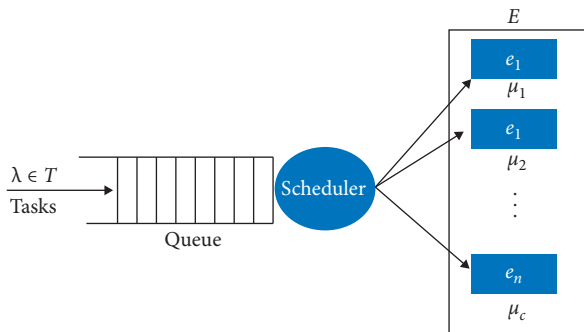


FIGURE 3: Data prefetching and eviction process based on task priority.

TABLE 2: An example of assigning tasks based on priority and data locality.

Arrival of tasks	Prioritized tasks	Required data	Computing nodes
t_1	t_1	d_8	$e_1 \ni d_8$
t_2	t_4	d_0	$e_2 \ni d_0$
t_3	t_5	$d_3 + d_6$	$e_2 \ni d_{3+6}$
t_4	t_2	d_2	$e_2 \ni d_2$
t_5	t_3	d_1	$e_3 \ni d_1$

FIGURE 4: Edge computing queue model $M/M/(e_n \in E)$.

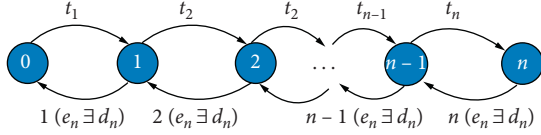
queuing model, remaining time, waiting time, and service time of the tasks in edge computing are mathematically evaluated.

As the requests to edge nodes come from the end devices like smartphones, tablets, and wearable devices, the pool of the tasks and the size of the queue are considered to be limitless in the task manager at the cluster of edge nodes. The state transition diagram of $M/M/(e_n \in E)$, which can be denoted through balance equations, is shown in Figure 5. When the number of tasks $t_n \in T$ is less than the number of computing nodes $e_n \ni d_n$, only n out of the nodes e_n are busy and the mean service rate is equal to n . From (4), we can obtain

$$P_n = P_0 \left[\frac{(e_n \ni d_n \rho)^n}{(n)!} \right] \quad (1 \leq n \leq e_n). \quad (7)$$

If the number of tasks is greater than or equal to $e_n \ni d_n$, i.e., $n \geq e_n \ni d_n$, all the nodes are busy and the effective service rate is equal to $\mu(e_n \ni d_n)$. Thus,

$$P_n = P_0 \left[\frac{(e_n \ni d_n \rho)^n}{(e_n \ni d_n)^{n-P} (e_n \ni d_n)!} \right] \quad \text{for } n > (e_n \ni d_n). \quad (8)$$

FIGURE 5: State transition diagram of $M/M/1 (e_n \in E)$.

Here, $\rho = \lambda/\mu (e_n \exists d_n)$ where ρ must be less than 1 for system stability. Note that the expected number of busy nodes is equal to $\rho (e_n \exists d_n) = \lambda/\mu$. To obtain P_0 , both sides of (7) and (8) are summed up. Since $\sum_{n=0}^{\infty} P_n = 1$, P_0 is derived as

$$P_0 = \left[\sum_{n=1}^{e_n-1} \frac{(e_n \exists d_n \rho)^n}{n!} + \frac{\rho^{e_n}}{(e_n \exists d_n)!(1-\rho)} \right]^{-1}. \quad (9)$$

The proposed TPDS is an efficient scheduling strategy that minimizes the costs of data transfer and execution latency. For evaluation of the system performance, it is necessary to calculate the total number of tasks in the queue, the total waiting time, the service time of the jobs, and the total number of tasks in the system. If the number of incoming tasks is less than the number of nodes in the cluster as represented in (7), the system is under the stable condition. Thus, it is expected that all tasks can be completed on time with no extra waiting in the queue. Otherwise, as in (8), it is highly probable that some tasks wait for long time and never get a service. The proposed TPDS tries to optimally minimize unnecessary eviction and improve the data locality for the tasks. As discussed earlier, when $n > e_n \exists d_n$, some tasks must wait in the queue. Thus, the estimated number of tasks in the queue is given by

$$N_Q = P_0 \frac{(\rho)^{e_n \exists d_n}}{(e_n \exists d_n - 1)!(\mu e_n \exists d_n - \lambda)^2}. \quad (10)$$

To evaluate the system performance by applying Little's law, it is necessary to obtain the total waiting time of tasks before service, the total number of tasks in the queue, and the total time spent by a single task in the cluster of edge computing nodes.

$$W_Q = P_0 \frac{\mu (\rho)^{e_n \exists d_n}}{(e_n \exists d_n - 1)!(\mu e_n \exists d_n - \lambda)^2}, \quad (11)$$

$$T_s = P_0 \left[\frac{\mu (\rho)^{\mu e_n \exists d_n}}{(e_n \exists d_n - 1)!(\mu e_n \exists d_n - \lambda)^2} \right] + \frac{1}{\mu}, \quad (12)$$

$$N_s = P_0 \left[\frac{\lambda \mu (\rho)^{e_n \exists d_n}}{(\mu e_n \exists d_n - 1)!(\mu e_n \exists d_n - \lambda)^2} \right] + \frac{\lambda}{\mu}. \quad (13)$$

The probability that all nodes are busy in edge computing clusters can be derived from (14) and (15).

$$P_B = P_0 \sum_{n=e_n}^{\infty} \left(\rho^n \frac{(e_n \exists d_n)^{e_n}}{(e_n \exists d_n)!} \right), \quad (14)$$

$$P_B = P_0 \frac{(e_n \exists d_n)^{e_n}}{(e_n \exists d_n - 1)!(\mu e_n \exists d_n - \lambda)}. \quad (15)$$

5. Performance Evaluation

In this section, the proposed TPDS is evaluated through computer simulations. The job completion time and node utilization under data locality in the cache memory are estimated by Cloudsim [33]. Cloudsim includes a broker (task manager node) and client nodes (number of machines) entities. The results of the proposed TPDS are compared with the existing scheduling and eviction schemes: FIFO, LRU, and HPSO [23, 34]. The efficiency of the proposed TPDS strategy is evaluated in terms of hit ratio of cached-data, task execution time, task waiting time, and data locality. The parameter details for the Cloudsim simulator are given in Table 4.

Figure 6 shows the used ratio of data for the proposed TPDS, compared with the three conventional schemes. The proposed TPDS maximizes utilization of the cached-data by using it for the incoming task in the queue. The proposed TPDS is not blindly swapping out the old data without knowing the incoming task in the queue. Thus, it is noted that the hit ratio is higher in the proposed TPDS than the conventional FIFO, LRU, and HPSO schemes. Particularly, when the number of data blocks increases, the time consumed for completing the task for all the schemes also increases due to swapping out data blocks from the cache memory without checking the queue of the task manager. This causes lower hit ratios in the cached-data as the number of data blocks increases.

Figure 7 shows the execution times of tasks of the proposed TPDS and three conventional schemes. As shown in Figure 8, the execution time of the task for the proposed TPDS is always smaller than those of the conventional FIFO, LRU, and HPSO schemes. It is because the more swapping out of data gives the longer waiting time to the task to update the associated data for the incoming task. Pre-existing data for tasks will execute the task quickly and there is no need to wait to bring the related data.

Similarly, Figure 8 shows the average waiting time of tasks. From the figure, it is observed that the waiting time of the proposed TPDS is smaller than those of the conventional FIFO, LRU, and HPSO schemes. Compared to the conventional schemes, the proposed TPDS consistently allows shorter average waiting time in the whole range of the number of tasks. The number of tasks is varying from 200 to 2200 and the same distribution of job sizes is maintained throughout the simulation test. The proposed TPDS significantly outperforms the other conventional schemes in terms of average execution time as the number of tasks increases.

TABLE 3: The variables used in the performance model.

Variable	Description
N_Q	The number of tasks waiting in the queue for service
N_s	Total number of tasks in the edge computing system
T_s	The total time spent by the task in the edge computing system
W_Q	The total time of the task waiting in the queue for service
P_n	The probability that the system has “ n ” number of tasks
P_b	The probability that all nodes are busy in the edge computing system
$e_n \exists d_n$	The computing node that contains the related data, d_n , for the task

TABLE 4: Cloudsim simulator parameters.

Entity	Parameters	Values
Cloudlets/task	Length of the task (expected number of instructions)	50–2000
	Number of tasks for six times running	200 to 2200
	The priority of tasks based on the cached-data blocks	High, medium, and low
Virtual machine	Number of VMs	50 in ache data center
	VM memory	1 GB to 2 GB
	Bandwidth	500–1000
	Number of CPUs	1 to 3
Data center	Number of data centers	5
	Number of hosts	2 to 3

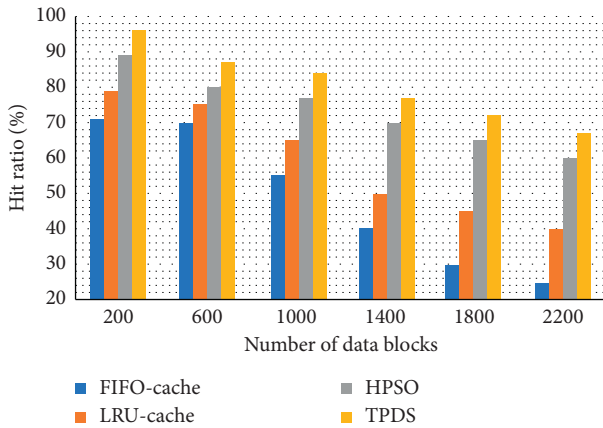


FIGURE 6: The comparison of hit ratio.

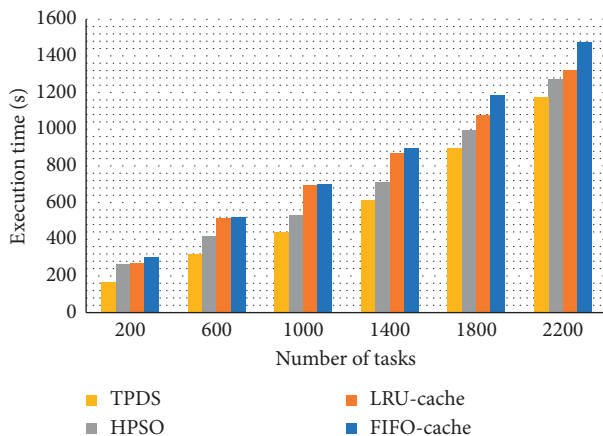


FIGURE 7: The comparison of task execution time.

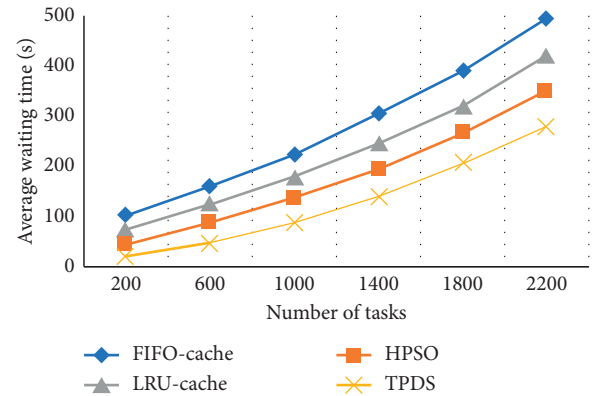


FIGURE 8: The comparison of average waiting time.

Another feature of the proposed TPDS is the priority scheduling of tasks as shown in Figure 9. It is noted that, with help of this scheduling strategy, the jobs achieve nearly the best data locality, which is helpful to improve the performance of distributed systems. The proposed TPDS takes advantage of cached-data locality to accelerate the computation of the task and minimize the CPU usage and data transfer load in terms of swapping out and swapping in data from the cache memory. It significantly improves the performance of the computing nodes and the execution of tasks. It is shown that the proposed TPDS also always outperforms the conventional schemes in terms of data locality.

In Figure 10, the average execution time of the proposed TPDS is compared with the conventional schemes. We use six different types of workloads as different numbers of data blocks (200 to 2200). Compared to the conventional schemes, the proposed TPDS reduces the average execution time by 8.5% to 10.2% for six different workloads,

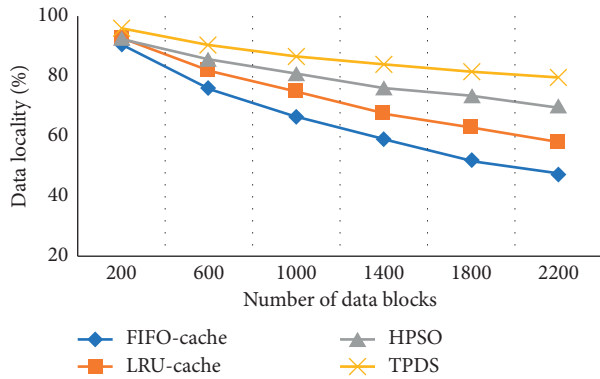


FIGURE 9: The comparison of data locality.

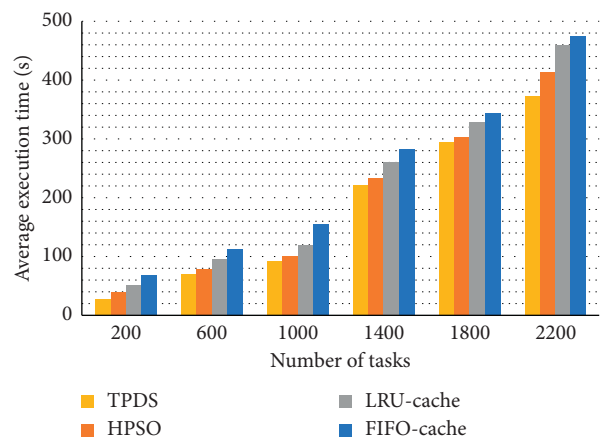


FIGURE 10: The comparison of average execution time.

respectively. This demonstrates that the proposed TPDS performs data locality more efficiently than the existing schemes due to the availability of data blocks in the cache memory.

6. Conclusion

As the number of IoT devices and the scale of cloud computing grow in popularity, many edge computing and distributed systems have emerged in recent years. In general edge computing architecture, computing power, bandwidth, and data at the edge are scarce resources. To improve system performance, a task scheduling strategy must be efficient. In this paper, we proposed a cache data locality scheduler for edge-computing cluster environments. The proposed strategy schedules tasks by taking a broad view and adjusts data for tasks dynamically according to data in cache memory. Especially in an edge computing cluster environment, where the number of resources is limited, our proposed approach tries its best to enhance task execution under limited resources and reduce the extra flow of data in the cluster network. When the computing cluster is overloaded, the proposed strategy takes the advantage of data in the cache and brings the task first which finds the needed data in the cache of the node. The simulation results show that the proposed strategy exhibits some improvements

which can also work in a busy network and cluster. As future work, we plan to improve the proposed task scheduling strategy based on available resources. We will consider the aspects that may affect the performance including data distributions and replication in a heterogeneous system. Edge computing and distributed technologies are growing up due to massive data volume generated by a large number of IoT devices. Accordingly, it is essential to keep update and development on scheduling strategies and efficient algorithms for tasks to manage resources in edge computing environments.

Data Availability

The data used to support the findings of this study are included within this article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2018-0-01426) supervised by the IITP (Institute for Information and Communication Technology Planning & Evaluation) and in part by the National Research Foundation (NRF) funded by the Korea Government (MSIT) (no. 2019R1F1A1059125).

References

- [1] B. Liu, H. Xu, and X. Zhou, "Resource allocation in wireless-powered mobile edge computing systems for internet of things applications," *Electronics*, vol. 8, no. 2, p. 206, 2019.
- [2] Z. Xu, W. Liu, J. Huang, C. Yang, J. Lu, and H. Tan, "Artificial intelligence for securing IoT services in edge computing: a survey," *Security and Communication Networks*, vol. 2020, 2020.
- [3] P. Garcia Lopez, A. Montesor, D. Epema et al., "Edge-centric computing," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.
- [4] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, "Fog computing: focusing on mobile users at the edge," 2015, <https://arxiv.org/abs/1502.01815>.
- [5] I. Ullah and H. Y. Youn, "Task classification and scheduling based on K-means clustering for edge computing," *Wireless Personal Communications*, vol. 113, no. 4, pp. 2611–2624, 2020.
- [6] J. Bellendorf and Z. Á. Mann, "Classification of optimization problems in fog computing," *Future Generation Computer Systems*, vol. 107, pp. 158–176, 2020.
- [7] S. A. Chaudhry, K. Yahya, F. Al-Turjman, and M.-H. Yang, "A secure and reliable device access control scheme for IoT based sensor cloud systems," *IEEE Access*, vol. 8, pp. 139244–139254, 2020.
- [8] L. M. Dang, M. J. Piran, D. Han, K. Min, and H. Moon, "A survey on internet of things and cloud computing for healthcare," *Electronics*, vol. 8, no. 7, p. 768, 2019.

- [9] D. C. Marinescu, *Cloud Computing: Theory and Practice*, Morgan Kaufmann, Burlington, MA, USA, 2017.
- [10] J. Choi, T. Adufu, and Y. Kim, "Data-locality aware scientific workflow scheduling methods in HPC cloud environments," *International Journal of Parallel Programming*, vol. 45, no. 5, pp. 1128–1141, 2017.
- [11] I. Ullah, M. S. Khan, M. Amir, J. Kim, and S. M. Kim, "LSTPD: least slack time-based preemptive deadline constraint scheduler for Hadoop clusters," *IEEE Access*, vol. 8, pp. 111751–111762, 2020.
- [12] C.-H. Chen, T.-Y. Hsia, Y. Huang, and S.-Y. Kuo, "Scheduling-aware data prefetching for data processing services in cloud," in *Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications*, pp. 835–842, Taipei, Taiwan, March 2017.
- [13] M. Sun, H. Zhuang, C. Li, K. Lu, and X. Zhou, "Scheduling algorithm based on prefetching in MapReduce clusters," *Applied Soft Computing*, vol. 38, pp. 1109–1118, 2016.
- [14] C. Li, J. Zhang, Y. Chen, and Y. Luo, "Data prefetching and file synchronizing for performance optimization in Hadoop-based hybrid cloud," *Journal of Systems and Software*, vol. 151, pp. 133–149, 2019.
- [15] C.-H. Chen, T.-Y. Hsia, Y. Huang, and S.-Y. Kuo, "Data prefetching and eviction mechanisms of in-memory storage systems based on scheduling for big data processing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 8, pp. 1738–1752, 2019.
- [16] H. Cui, X. Liu, T. Yu, H. Zhang, Y. Fang, and Z. Xia, "Cloud service scheduling algorithm research and optimization," *Security and Communication Networks*, vol. 2017, 2017.
- [17] A. Samanta, Z. Chang, and Z. Han, "Latency-oblivious distributed task scheduling for mobile edge computing," in *Proceedings of the 2018 IEEE Global Communications Conference*, pp. 1–7, Abu Dhabi, United Arab Emirates, December 2018.
- [18] Y. Deng, Z. Chen, X. Yao, S. Hassan, and J. Wu, "Task scheduling for smart city applications based on multi-server mobile edge computing," *IEEE Access*, vol. 7, pp. 14410–14421, 2019.
- [19] T. Zhu, T. Shi, J. Li, Z. Cai, and X. Zhou, "Task scheduling in deadline-aware mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4854–4866, 2018.
- [20] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, "An improved anonymous authentication scheme for distributed mobile cloud computing services," *Cluster Computing*, vol. 22, no. 1, pp. 1595–1609, 2019.
- [21] M. Wang and Q. Zhang, "Optimized data storage algorithm of IoT based on cloud computing in distributed system," *Computer Communications*, vol. 157, pp. 124–131, 2020.
- [22] L. M. Haji, S. Zeebaree, O. M. Ahmed, A. B. Sallow, K. Jacksi, and R. R. Zeabri, "Dynamic resource allocation for distributed systems and cloud computing," *TEST Engineering & Management*, vol. 83, pp. 22417–22426, 2020.
- [23] K. Lu, D. Dai, X. Zhou, M. Sun, C. Li, and H. Zhuang, "Unbinds data and tasks to improving the hadoop performance," in *Proceedings of the 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 1–6, Melbourne, Australia, August 2014.
- [24] L. Guo, T. Yan, S. Zhao, and C. Jiang, "Dynamic performance optimization for cloud computing using M/M/m queueing system," *Journal of Applied Mathematics*, vol. 2014, Article ID 756592, 18 pages, 2014.
- [25] G. Chen, S. Wu, R. Gu et al., "Data prefetching for scientific workflow based on Hadoop," *Computer and Information Science 2012*, vol. 429, pp. 81–92, 2012.
- [26] X. Wei, J. Liu, Y. Wang, C. Tang, and Y. Hu, "Wireless edge caching based on content similarity in dynamic environments," *Journal of Systems Architecture*, vol. 115, Article ID 102000, 2021.
- [27] G. L. Stavrinides and H. D. Karatza, "Dynamic scheduling of bags-of-tasks with sensitive input data and end-to-end deadlines in a hybrid cloud," *Multimedia Tools and Applications*, pp. 1–23, 2020.
- [28] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.
- [29] I. Raicu, I. T. Foster, Y. Zhao et al., "The quest for scalable support of data-intensive workloads in distributed systems," in *Proceedings of the 18th ACM international symposium on High performance distributed computing-HPDC '09*, pp. 207–216, Munich, Germany, June 2009.
- [30] S. N. Prasad, S. Kulkarni, and P. Venkatarreddy, "Cache aware task scheduling algorithm for heterogeneous cloud computing environment," in *Proceedings of the Fifth International Conference on Research in Computational Intelligence and Communication Networks*, pp. 154–158, Kolkata, India, March 2020.
- [31] S. Suresh, N. Gopalan, and N. P. Gopalan, "Delay scheduling based replication scheme for Hadoop distributed file system," *International Journal of Information Technology and Computer Science*, vol. 7, no. 4, p. 73, 2015.
- [32] N. S. Naik, A. Negi, T. B. B.R., and R. Anitha, "A data locality based scheduler to enhance MapReduce performance in heterogeneous environments," *Future Generation Computer Systems*, vol. 90, pp. 423–434, 2019.
- [33] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.
- [34] M. Sun, H. Zhuang, X. Zhou, K. Lu, and C. Li, "HPSO: prefetching based scheduling to improve data locality for mapreduce clusters," *Algorithms and Architectures for Parallel Processing, Lecture Notes in Computer Science*, vol. 8631, pp. 82–95, 2014.
- [35] B. Jiang, J. Wu, X. Shi, and R. Huang, "Hadoop scheduling base on data locality," 2015, <https://arxiv.org/abs/1506.00425>.

Research Article

S-DPS: An SDN-Based DDoS Protection System for Smart Grids

Hassan Mahmood,^{1,2} Danish Mahmood ^{1,2}, Qaisar Shaheen ³, Rizwan Akhtar,⁴ and Wang Changda ¹

¹School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China

²Department of Computer Science, SZABIST, Islamabad, Pakistan

³Department of Computer Science, Superior College, Lahore, Pakistan

⁴Department of IT and Computer Science, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur, Pakistan

Correspondence should be addressed to Danish Mahmood; dr.danish@szabist-isb.edu.pk and Wang Changda; changda@ujs.edu.cn

Received 31 December 2020; Revised 19 February 2021; Accepted 8 March 2021; Published 22 March 2021

Academic Editor: Shehzad Chaudhry

Copyright © 2021 Hassan Mahmood et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Information Communication Technology (ICT) environment in traditional power grids makes detection and mitigation of DDoS attacks more challenging. Existing security technologies, besides their efficiency, are not adequate to cater to DDoS security in Smart Grids (SGs) due to highly distributed and dynamic network environments. Recently, emerging Software Defined Networking- (SDN-) based approaches are proposed by researchers for SG's DDoS protection; however, they are only able to protect against flooding attacks and are dependent on static thresholds. The proposed approach, i.e., Software Defined Networking-based DDoS Protection System (S-DPS), is efficiently addressing these issues by employing light-weight Tsallis entropy-based defense mechanisms using SDN environment. It provides early detection mechanism with mitigation of anomaly in real time. The approach offers the best deployment location of defense mechanism due to the centralized control of network. Moreover, the employment of a dynamic threshold mechanism is making detection process adaptive to the changing network conditions. S-DPS has demonstrated its effectiveness and efficiency in terms of Detection Rate (DR) and minimal CPU/RAM utilization, considering DDoS protection focusing smurf attacks, socket stress attacks, and SYN flood attacks.

1. Introduction

There is a drastic increase in energy dependence from very minute to huge activity especially the cloud-based data centers and Internet of Things (IoT), which have a dire need of availability, reliability, and efficiency of power systems. This requirement paved path towards the Smart Grid (SG) paradigm that ensures two-way communication within power systems. It holds the capability to remove the constraints of a traditional grid infrastructure and provide power systems that are scalable, dynamic, situation-aware, and flexible. On the other hand, such facilities give birth to complexity, heterogeneity, and interconnectivity of diverse ICT requirements due to which the existing network paradigms and security strategies are marked as ineffective

[1, 2]. Moreover, the IP-enabled communication infrastructure in SGs raises the likelihood of malicious activities and attacks. Such attacks may result in wrong smart meter readings or incorrect demands or responses to or from electricity company or they can be severe for power generation systems [3].

Millions of consumers are serviced by SG. The service provided by SG is crucial and the availability of such a service is extremely important. The SG makes up a cyber-physical system (CPS) and a single error in any part of the system can lead to a direct or indirect catastrophic effect on human life [4]. Distributed Denial of Service (DDoS) attacks are also making more frequent appearances and are becoming more sophisticated and severe because of the fact that the existing protection mechanisms are not capable to deal with such threats. Hence,

detection, mitigation, and prevention of DDoS attacks are now on the top most priority of engineering industries and research arena. Researches have come up with SDN-based approaches to handle DDoS attacks in SG [5]. However, still experimental validation of the proposed approaches is lagging. Also, SGs are safeguarded against High-Rate (HR)-DDoS attacks only during their detection and mitigation approach [1] [6]. This level of safety is not enough for a sensitive ICT infrastructure such as SG that carries mission critical information. Because of these reasons, there exists a desperate need for further research regarding SDN-based security protocols in SG to ensure a safe and light-weight mechanism against DDoS, having a capability to detect in the early stages and mitigation of varied level of DDoS attacks.

The remainder structure of the paper is organized as follows: Section 2 discusses related work with critical evaluation of literature, motivation with problem statement, and approach with contribution. Section 3 discusses system model with implementation constraints. Experimental setup with evaluation criteria is discussed in Section 4. Results and discussion are presented in Section 5. Section 6 discusses performance evaluation of the approach. Finally, Section 7 presents conclusion with future work.

2. Literature Review

Considering the wide spread of ICT and upcoming IoT devices, applications, and scenarios in almost every field of life, the authors in [7] showcased the vulnerabilities that may attract negative attentions. Moreover, the authors discussed state-of-the-art work regarding mitigation of such malicious activities. Researchers from academia and industry have shown interest and utilize new network paradigm, i.e., SDN to deal with underlying security risks in SG communication network [5]. The authors in [8] presented a comprehensive survey focusing SG communication security measures and privacy breaches. Major emphasis is given to privacy handling within SG communication networks. In [9], the authors presented a taxonomy of network attacks focusing fog-based smart grid SCADA systems. The authors in this study classify intrusion detection systems (IDSs) as major solution for the attacks; however, they focused mainly on machine learning approaches which at times are more time consuming and compel in comparison to entropy-based IDSs.

The authors in [10] used blockchain for securing the data and SDN to deal with control issues and scalability. Furthermore, the authors in [11] also used blockchain to secure energy sector, mainly authentication and privacy issues.

SDN is recently used in SG to provide a resilient SDN-based security framework/simulators and communication architecture. Researchers have utilized either single or multicontroller architecture to establish the underlying network infrastructure [12] which is considered to be the first SG-enabled simulator which is resilient and secured. The proposed security module is able to detect and resolve DoS attack within 60 seconds with no impact on bus system. However, maximum power capacity allowed on each bus/branch is not mentioned to address how much additional load other branches can bear in case of failure. Static threshold of 40% for number of packets/sec is used in

detection mechanism. Similarly, in [1], a novel SDN-based communication architecture for resiliency and security of microgrid operations is proposed. They have used three applications, i.e., self-healing mechanism, network verification, and intrusion detection. Self-healing mechanism uses rapid network configuration changes to mitigate further penetration by doing traffic isolation. Network verification is implemented using consistent updates feature, to avoid network instability by ensuring consistency of packets. Specification-based intrusion detection system is proposed; however, experimental validation is missing in the work.

Entropy-based approaches have been widely in traditional networks to provide DDoS protection. These approaches have proved to be useful in SDN environment as well, providing better detection efficiency [13]. The authors in [14] used open flow SDN controller to detect DDoS attacks on SG. As this is basic feature of SDN framework, the proposed methodology is also situation aware; however, for anomaly detection, an entropy-based mechanism is proposed which not only detects but also mitigates the attacks. However, the authors have not enhanced their proposed model to adoptively change according to situation and environment.

The authors in [15] presented a DDoS traceback mechanism under the umbrella of SDN architecture. The authors established an anomaly tree by analysing the communication flow changes via base station nodes. Once the anomaly tree is formulated, traceback scheme calls out any of different DDoS protection algorithms depending upon the nature and severeness of the attack. The authors claim that proposed scheme is better than the state-of-the-art frameworks regarding detection and trace back time with minimal usage of resources. In the future, the authors intend to optimize this approach by making it adoptive such that it can detect most types of the DDoS attacks.

A scheduling algorithm based on two levels is proposed in [16] to make sure better QoS regarding power services and communication network. For this purpose, an SDN controller is utilized and in first level, a scheduling mechanism is devised focusing priority in terms of delay constrained power services. Once priority-based services are schedules, then congestion and queueing control mechanism follows which ensures minimal delay with respect to the priority assigned. The authors used Mininet and Ryu controllers for simulation purposes. The proposed approach reduced delay and packet loss ratio with respect to state-of-the-art work. An elliptic curve cryptography (ECC) is presented in [17] and the proposed scheme which is based on mutual authentication by using biometric system. The authors claim to eliminate many authentication attacks.

Moreover, ECC technique supersedes other state-of-the-art protocols considering the performance metrics of communication and computational time considering SG environment. The authors in [18] proposed multilevel autoencoders-based IDS for DDoS attacks in SGs. The authors claim to have better accuracy in predictive analysis with other state-of-the-art methods. In [19], the authors presented a novel SDN-based IDS for SG. Basic feature of SDN, i.e., centralized controller in control

plane, is made distributed by using blockchain approach. The proposed model is simulated using AnyLogic and results declare it as more effective in terms of DDoS detection with state-of-the-art frameworks. Moreover, this approach also reduces the controller overhead; however, the delay in decision making is the trade-off that is not required in demand responsiveness feature of SG.

The authors in [20] present a novel entropy-based statistical approach in multicontroller SDN environment approach which is proposed for early detection and mitigation of DDoS attack. Apart from early detection, it is able to identify the attack path as well to apply the mitigation strategy instantly after detection. Shannon entropy with experimental static threshold against “DA” is used as detection mechanism, whereas Drop/block ports mechanism is used for mitigation purpose. Experimental validation for backup controller functionality, in case of primary controller failure, is missing. Threshold mechanism should have been adaptive rather than static, considering dynamic nature of modern networks. The authors have not addressed the efficacy of approach in protecting against LR-DDoS attacks. Further, approach should have been validated using performance metrics like DR and FPR.

Apart from SDN-based approaches for DDoS protection, it is important to discuss existing entropy-based approaches that have been successful in detecting DDoS attacks in traditional networks. Different variants of entropy are available for detecting DoS/DDoS attacks, i.e., generalized entropy, Tsallis entropy, and normalized entropy. Each of them can achieve varying DR and FPR for HR and LR-DDoS attacks. Some are able to detect both types of DDoS attacks with better DR and FPR, where some are best suited for a single type only. These solutions depend on the traffic features and perform statistical procedures on normal and attack traffic to do the comparison in order to find the anomaly. The authors in [21] present a generalized entropy-based feature selection technique which is used to detect network anomalies from real-life WAN traffic data with a high DR and low FPR. An outlier score function is used to detect the anomalies. The algorithm was evaluated against other techniques like LOF and ORCA using dataset Zoo. They achieved DR of 94.11% and FPR of 2.38%, higher than the other two approaches. However, user-defined parameters for threshold values are used. Although these values are set after conducting training on datasets, but still, it poses a limitation with respect to dynamic nature of networks and underlying attacks. The approach did not directly work on categorical and mixed types data.

A variant of Renyi entropy is proposed in [22], as a light-weight detection system utilizing extended entropy-based metric to detect HR-DDoS flooding attack and IP traceback. The proposed approach is evaluated against other entropy metrics like Shannon entropy and Kullback–Leibler divergence using both simulated and real-time DDoS datasets. Another important variant of parameterized entropy, i.e., Tsallis entropy, is utilized by researchers for anomaly detection. A feature-based Anomaly Detection System (ADS) using Tsallis entropy at device level is proposed in [23] and is capable of detecting and classifying known and unknown anomalies with additional information regarding network usage. Primitive properties of

flows like SA, DA, SP, and DP and derived flow properties at device and network level, i.e., out-degree, in-degree, per flow, per packet, per byte, packet per sample (pps), etc., are used in flow extraction process. Based on the discussion above, it can be observed that the authors have not highlighted the capability of their approach to detect both HR and LR-DDoS attacks. Moreover, static thresholds based on experiments are utilized in their approaches which cannot prevail in dynamic and complex environments like SG. Real dataset for DDoS attacks based on SG networks are not publically available easily [24]; hence, researchers have used simulated datasets for validation of their work.

Tsallis entropy metric has performed well, as per validation metrics, compared to other entropic metrics in detecting varying number of DDoS attacks, i.e., both LR-DDoS and HR-DDoS attacks. For effective DDoS defense mechanism, mitigation strategies should also be incorporated with intrusion detection system. Placement of detection mechanism is way important for efficient detection and in-time capitalization of DDoS attack. Such fact has not been addressed by many of the researchers. Finally, Tsallis entropy metric, besides its efficiency with respect to DR and FPR, has not been tested in an SDN environment. Utilizing SDN for securing SGs is in focus for energy engineering industries and research arena as well. The authors in [25] orchestrate a strategic connection, monitoring SDN controllers and sources of new flow requests that are threatening for DDoS attack. Compromised switches are identified and a noncooperative game is orchestrated using dynamic Bayesian network. The authors in [26] proposed a DDoS detection mechanism in SGs using Convolutional Neural Network. Variance fractal dimension trajectory is used as a preprocessing tool, whereas postprocessing of data is conducted by employing support vector machine. The authors claimed to achieve 87.35% accuracy in DDoS detection. Critical evaluation of literature is given in Tables 1 and 2.

2.1. Motivation and Problem Statement. In light of the above discussion, it can be concluded that SDN significantly addresses the deployment locality requirement to its centralized controller architecture. Further, entropy-based techniques used by the researchers rely on experimental-based thresholds and do not adapt to changing network conditions. Therefore, it necessitates developing an adaptive light-weight entropy-based defence mechanism using SDN environment for SG, providing early detection and mitigation of anomaly in real time. Real-time reconfiguration based on network conditions is required to change static thresholds and also to make it appropriate for high Detection Rate (DR) and low False Positive Rate (FPR). Here DR measures portion of the attacks that are detected correctly by the system and FPR provides the percentage of events that are reported as negative events where actually they are positive events. This factor makes it highly inappropriate for SG due to dynamic nature and heterogeneity.

With the advent of IoT, security concerns related to user and network resources have become even more critical and prone to attacks. SG being one important application of IoT also shares the same security threats that exist in traditional IoT environment. However, protection of DDoS attacks in

TABLE 1: SDN-based security approaches for DDoS protection.

Approach	Security parameter	Network/dataset	Experimental setup	Tools/simulators	Parameters/approach for intrusion detection	Limitations
DSSnet: microgrid simulator [1]	DoS and resilience	IEEE-13 bus power distribution system with two subsystems, i.e., wind turbine and energy storage system	Developed a simulator for evaluation of microgrid operation. Applications: (i) Self-healing network management (ii) Communication network verification (iii) Specification-based intrusion detection	OpenDSS, Mininet, virtual time system (Linux-based kernel)	(i) Network slicing (ii) Traffic isolation	A little literature on specification-based intrusion detection provided experimental validation of intrusion detection is not provided.
PYGRID: SG simulator [12]	DoS protection and resilience	Simulated IEEE-14 bus power system	Scenarios: normal operation, bus failure, and bus attack Result: successfully mitigated DDoS attack	Mininet, PYPOWER	(i) Number of packets/second = 40% threshold (ii) Flows count	(i) Maximum power capacity allowed on each bus/branch is not mentioned; rationale for using fixed threshold limit for number of packets/sec is missing. (ii) All traffic flows are being monitored for rapid detection. Computation overhead cost is associated with the approach since all flows will go to application layer. (i) Experimental validation for backup controller functionality, in case of primary controller failure, was missing. (ii) Threshold value should have been changed dynamically as per the changing network environment. (iii) The authors did not address the efficacy of approach in protecting against LR-DDoS attacks (iv) Flash crowds may be detected by the algorithm as an attack, resulting in extra FPR. (v) The proposed approach should have been validated against performance metrics like DR, FPR, etc.
Multicontroller-based SDN [20]	UDP/TCP/ICMP flood attacks	Simulated	Design components entropy-based DDoS detection algorithm (i) Virtualized network environment of 3 switches and 32 hosts (ii) Set of mitigation actions (block traffic/ports) (iii) UDP flood attack simulated	POX controller, Mininet 2.0, and Scapy tool for traffic generation	Analysis metric (i) Destination IP address entropy	

TABLE 2: Entropy-based approaches.

Year	Technique	Anomalies addressed	Dataset	Data source	Source tool	Flow properties for anomaly detection	Comparison	Validation metrics (%)	Conclusion
2016 [21]	Generalized entropy	DDoS, probe	Real: KDDcup99, NSL-KDD, UCI machine learning repository datasets	IP packet/ IP flow	Netflow data	Dynamic selection of features through mutual information and GE	LOF for $\tau = 0.58$ at dataset Zoo	DR = 82.35 FPR = 19.04	Proposed approach achieved better DR and FPR metrics compared to other outlier approaches
			Simulated: Testbed dataset (TUIDS) for DDoS and probe attacks				ORCA	DR = 88.23 FPR = 13.09	
2015 [22]	Extended entropy	DDoS, port scan, network scan, DoS, worm, and spam	Legitimate traffic from tsinghua University Campus network	IP flow	Netflow	Source IP address, source port, destination IP, address, destination port, flow byte, flow direction, protocol number, and TCP control bit	—	DR = 93.46 FPR = 5	2015
			Real and simulated versions: Toledo Campus and FISTSC/GW campus				Tsallis entropy	DR = 100 FPR = 1	
2017 [23]	Tsallis entropy	DDoS, alpha flow, port scan, network scan	Real Campus network data, i.e., UTFPR/Toledo Campus and FISTSC/GW campus	IP flow	Netflow v9	Source address, destination address, source port, destination port, number of packets, number of flows, number of bytes, in-degree	Shannon entropy	DR = 25 FPR = 2.2806	Achieved better DR and FPR compared to Shannon entropy validation metrics dropped a little with sampling effects
							After incorporating sampling effects in technique	DR = 99.45 FPR = 0.12	

SG has grabbed more attention of researchers. The reason is the occurrence of massive DDoS attack on Ukraine power grid in 2015. Existing security protocols/techniques provide network protection at Internet edge only and are not sufficient enough to prevent dynamic attacks, considering borderless architecture of IoT. Additionally, current approaches of security, i.e., firewall zoning and intrusion detection and prevention system (IDPS) are too constrained by traditional network architecture. They are computationally heavy when considering the increase in network devices [27]. If appropriate security actions are not taken, then attacks like DDOS, service unavailability, and most importantly threat to human life might happen. Moreover, early detection and mitigation are deemed necessary for infrastructure like SG since deep penetration to SG network can lead to devastating consequences.

Entropy-based techniques used by the researchers rely on experimental-based thresholds and do not adapt to

changing network conditions. Moreover, utilization of static experimental thresholds and Shannon entropy do not provide adequate security against both HR and LR-DDoS attacks for an ICT infrastructure like SG. Static thresholds need to be reconfigured on changing network conditions to adjust for high DR and low FPR and that makes it unsuitable for SG. Moreover, Shannon entropy provides low DR and FPR as compared to Tsallis entropy [23] and on detection of DDoS attack, it is important to mitigate it as well to prevent its penetration further in the network; that is missing in DDoS protection approaches. In order to improve the security and reliability of SG in reference to DDoS attacks, researchers have suggested an SDN-based approach to handle the glitches in the conventional network paradigms. However, these approaches [6, 12, 20] are still only capable of handling HR-DDoS attacks, i.e., TCP/UDP/ICMP-based flooding attacks only, not catering stealthy and low-rate DDoS attacks, and also rely heavily on static thresholds.

Hence, it makes it necessary to develop a light-weight DDoS defence mechanism for SG that is fueled by SDN environment and using Tsallis entropy for better DR and FPR. Additionally, the SDN environment should be adaptive and capable of providing early detection and mitigation of both HR and LR-DDoS attacks.

2.2. Solution and Contributions. Considering our proposed solution, DDoS detection application uses Tsallis entropy metric with traffic features, i.e., Source Address (SA), Destination Address (DA), Source Port (SP), and Destination Port (DP) for efficient detection of varying DDoS attacks. For mitigation approach, IP address and port blocking mechanism is available in SDN controller software; i.e., OpenFlow is utilized. Blocking data is provided by the local list maintained in the SDN controller. Since SDN controller provides a global view of the whole network and is centrally located, the proposed approach significantly addresses the locality problem of DDoS defense mechanism that is missing in literature. A novelty in approach is added by using dynamic thresholds for traffic features using Exponential Weighted Moving Average (EWMA) instead of static threshold values for detection purposes. Moreover, to the best of our knowledge, Tsallis entropy in SDN environment has not been used previously. The proposed approach provides a near real-time detection within 250 packets with mitigation of anomaly in real time. In the following section, the proposed system model is discussed in detail. The following contributions are made in this work:

- (i) A light-weight entropy-based detection approach is developed underlying SDN environment
- (ii) Adaptive threshold mechanism is proposed to achieve better DR and False Positive Rate (FPR) using Exponentially Weighted Moving Average (EWMA) and Tsallis entropy
- (iii) Low-rate (LR)- and HR-DDoS attacks are successfully detected
- (iv) In addition to the real-time protection mechanism, a DDoS mitigation mechanism is also explained in terms of proposed model
- (v) Resource utilization (CPU and RAM utilization is optimized without compromising LR- or HR-DDoS protection)

3. System Model

In existing SDN-based solutions, a limited level of DDoS protection, i.e., against flooding-based DDoS attacks only, is being provided. Further, deployment locality of DDoS defence mechanism is critical in efficient and in-time detection. Most of the researchers did not fully address the issue of locality. SDN controller has a global view of the network and is responsible for all routing and filtering features of a network. In other words, it is a brain and central point of network. Therefore, SDN network utilization can provide optimal deployment locality for DDoS defence mechanism.

Lastly, software-based control of SDN provides IP address/port blocking mechanisms as a built-in feature.

So, these mechanisms can be optimally utilized as a DDoS mitigation approach. The conceptual framework is divided into two parts, namely, SDN-based environment for consumer-utility provider network and intrusion detection and prevention system (IDPS) as depicted in Figure 1. In this section, a detailed description of the system model is presented, following the implementation constraints and limitations.

IDPS is divided into three modules, namely, flow collector ("FC"), anomaly detector ("AD"), and anomaly mitigation ("AM"), as depicted in Figure 1. "FC" module is located in controller and collects network flows/packets and statistics from each connected switch through Netflow standard protocol, utilized by the controller. These flows are stored in the local database of the controller and relevant features, i.e., Source Address ("SA") and Destination Address ("DA"), are extracted for further processing by "AD." "AD" calculates Tsallis entropy value per traffic feature in current window of 50 packets and compares it with corresponding feature threshold value for that window. In case of a mismatch as per conditions discussed in subsequent section, an alarm is generated and further action is taken by the "AM" module. "AM" module performs drop/deny action on the flows and pass it on to v-switch performing forwarding decisions. It also stores the blacklisted IPs in blacklist database maintained locally in the controller for scrutiny of incoming packets. Threshold calculator calculates threshold values per feature for next window by applying Exponentially Weighted Moving Average (EWMA) on current window entropy value and previous window threshold value and passes it on to "AD" module for comparison purposes. Table 3 describes the primitive flow properties being used in the analysis. After extraction of required details, data is parsed to "AD" module which follows the mechanism as discussed in upcoming sections.

3.1. Anomaly Detector Module (AD Module). After extraction of traffic features ("SA," "SP," "DA," and "DP") by "FC" from new packets destined to the controller from OF switches through Netflow protocol supported by POX controller, data is fed to the "AD" module. Data in anomaly detector is processed based upon window size that can be based upon either time stamp of packet received or number of packets. For this work, it is based upon number of packets and set to 50 packets per window for efficient detection and memory foot-print [20]. Moreover, the experimental setup constitutes not more than 50 hosts (smart meters and utility server), so 50 packets per window is an appropriate window size. Therefore, consider W as the set of data with n elements in which each data element x_{mi} signifies the event pertaining to specific traffic feature as can be seen in (1). Probability of x_{mi} happening in window W can be calculated using (2). Further, Tsallis entropy is denoted by " H_q " which can be calculated by (3) [23]. For $q > 1$, higher probabilities have more impact on the final entropy value compared to lower

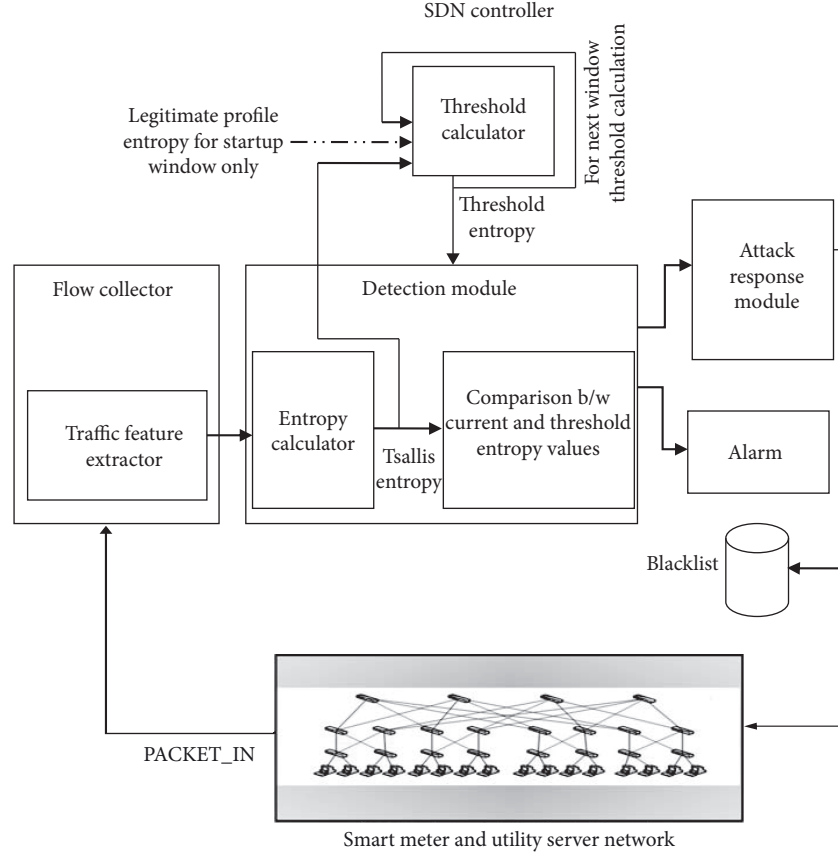


FIGURE 1: System model S-DPS.

TABLE 3: Primitive flow properties for AD.

No. (m)	Primitive flow property (xm)	Detail
1	SA	Source IP address of packet
2	DA	Destination IP address of packet

probabilities and vice versa. Here value of q is set as -1.3 or -0.8 for high DR and low FPR.

$$w = x_{mi}, \quad m = 1, 2, 3, 4, i = 1, 2, \dots, n, \quad (1)$$

$$P_{mi} = \frac{x_{mi}}{n}, \quad (2)$$

$$H_q = \frac{1}{q-1} \left(1 - \sum_{i=1}^n p_{mi}^q \right) \quad (3)$$

In each window, entropy values of four traffic features are calculated and compared with respective normal entropy value, using (4). Here $H_{q-(n)}^{x_m}$ is the entropy value of specific traffic feature taken in normal traffic conditions, i.e., without any abnormal traffic, and λ signifies the difference of entropies. In case the value of λ is positive, it means the entropy value of feature for current window has decreased; i.e., data distribution is concentrated. However, in case value of λ is negative, it means the

entropy value of feature for current window has increased; i.e., data distribution is dispersed.

$$H_{q-(n)}^{x_m} - H_q = \lambda. \quad (4)$$

Application of (4) is different for each traffic feature as depicted in Table 4. In case of DDoS attack, value of λ is positive for “DA” and negative for “SA,” whereas for different types of DDoS attacks, values of “SP” and “DP” are variable. Equations (1)–(4) are calculated for subsequent windows (50 packets per window) and in case value of λ is positive for “DA” and negative for “SA” for five consecutive windows, an alert for DDoS attack is generated. A counter for subject purpose is utilized, which is incremented on meeting the set conditions in each window. In case set conditions for “SA” and “DA” are not met in any 5 consecutive windows, counter is set to zero and process starts again with counter = 0.

3.2. Anomaly Mitigation Module. A specific action is associated by the controller with each flow in flow tables of the controller as discussed in background section related to OF protocol. In case an alarm is generated by the “AD” module, then the “SA” with maximum number of occurrences in the 5 windows is extracted and “drop/deny” action is set by the controller against the matched flows associated with that “SA” in run time.

TABLE 4: Interpretation of value of λ .

No. (m)	Flow property	Value of λ	Impact	Result
1	SA	Negative real number	Data dispersion	Attack from multiple IPs
2	DA	Positive real number	Data concentration	Attack towards specific IP

3.3. *Dynamic Threshold.* Initially, threshold values for each traffic feature are set by simulating the network environment in normal conditions, i.e., without any attack traffic. These threshold values are used to detect DDoS attack in progress as per the conditions discussed previously. Value of threshold dictates the performance of entropy-based detection approach in terms of DR and FPR. So, choosing optimal thresholds is most significant and important to achieve desired results. One approach is to conduct multiple experiments using attack traffic (tool or datasets) with normal traffic to tune these thresholds, while another approach is to utilize current network conditions in real time and system automatically updates these thresholds. The latter is more convenient and effective, considering the dynamic nature of SG network. So, in order to make the anomaly detection adaptive, consider a mean entropy value for each traffic feature as and for each subsequent window, mean entropy value for each traffic feature as a threshold, is calculated using (5). EWMA filter is used for calculating the average mean, and β value of 0.1 is used for catering current network conditions and is more reactive in nature, considering highly critical networks such as SG. Value of constant c depends upon the network characteristics.

$$H_{q(i)}^{x_m} = \left(\beta \times H_{q(i-1)}^{x_m} + (1 - \beta) \times H_{q(i)}^{x_m} \right) + c. \quad (5)$$

Threshold values, calculated as per (5), are based upon current network conditions with β value set to 0.1 (very reactive) and can result in high FPR for burst channel. Similarly, in case of stealthy attack pattern such as increasing and decreasing DDoS attacks, detection will be difficult. So, there is a need to tune the value of threshold in real time. In order to achieve optimum DR and FPR and keep the threshold in acceptable bounds, a maximum change/difference of current threshold from the normal entropy threshold (calculated during normal conditions) should not exceed by 1.5, considering 90% confidence interval for normal distribution. In case it exceeds more than 1.5 times, value of current threshold will become 1.5 times to normal entropy threshold; otherwise, it will remain the same as per the calculated mean threshold value. However, for decreasing entropy with respect to normal entropy threshold (calculated during normal conditions) for more than 1.5 times, value of current threshold will be normal entropy threshold value divided by 1.5 to normalize the threshold; otherwise, it will remain the same as per the calculated mean threshold value. The multiplication and division factor of 1.5 is used to keep the thresholds within reasonable bounds with respect to normal threshold value. The increasing entropy check is applicable for SA entropy, whereas decreasing entropy check is applicable for DA entropy. The reason is that DDoS attack tends to decrease DA entropy while

increasing SA entropy values. Flow chart for the algorithm is presented in Figure 2. In OF-based v-Switch, a packet for which no flow entry exists is passed on to controller for decision making. So, in the algorithm packet in flow step signifies entry of new packet in the controller. Traffic features of the packet as per Table 3 are checked for existence of entries already in the system. In case entries exist in the lists; then occurrence counters for each feature are incremented. Otherwise, new entries in the corresponding lists are made. If the number of packets count has reached 50 as per the set window, entropy value for each traffic feature using the corresponding traffic feature list is calculated. It is then compared with the threshold value. In case current DA entropy value is less than mean DA threshold value and current SA entropy value is greater than mean SA threshold value; then the consecutive window counter is incremented and the same cycle starts again for next window with number of packets count set to zero. Moreover, in case the current DA, entropy value is greater than mean DA threshold value and SA entropy is less than mean SA threshold value; then the cycle starts again with number of packets count set to zero.

3.4. *Implementation Constraints and Limitations.* As previously mentioned in related work, DDoS related datasets for SG are not publicly available and datasets like MIT Lincoln, FIFA, DDoSTB, and CAIDA datasets are not SG related [12]. Therefore, [1, 12] relied on simulated traffic to test the viability of their proposed approach. Similarly, in the paper, simulated normal and attack traffic is being generated using Scapy tool to test the proposed model because it is python-based and can be integrated with Mininet. Single topology is tested for different types of DDoS attacks and the traffic is simulated one. Results obtained may vary in real-time traffic. Moreover, model presented is independent of any protocol (tested for TCP/UDP/ICMP-based packets) and threshold for DDoS detection is being adjusted automatically with varying network conditions. So, solution is viable for dynamic network conditions as in modern networks. Apart from it, the solution is tested using a software-based simulator. Its capability will further be improved with powerful hardware-based SDN controller available in the markets.

Furthermore, the approach is based on single controller architecture, wherein it can present a bottleneck and security constraint when dealing with large-scale network like SG. Difference between using single-controller and multicontroller architecture is linked to load balancing, high availability, and security of controller. However, for this research it is outside the scope of work and the approach can be integrated and tested with multicontroller architecture for future research.

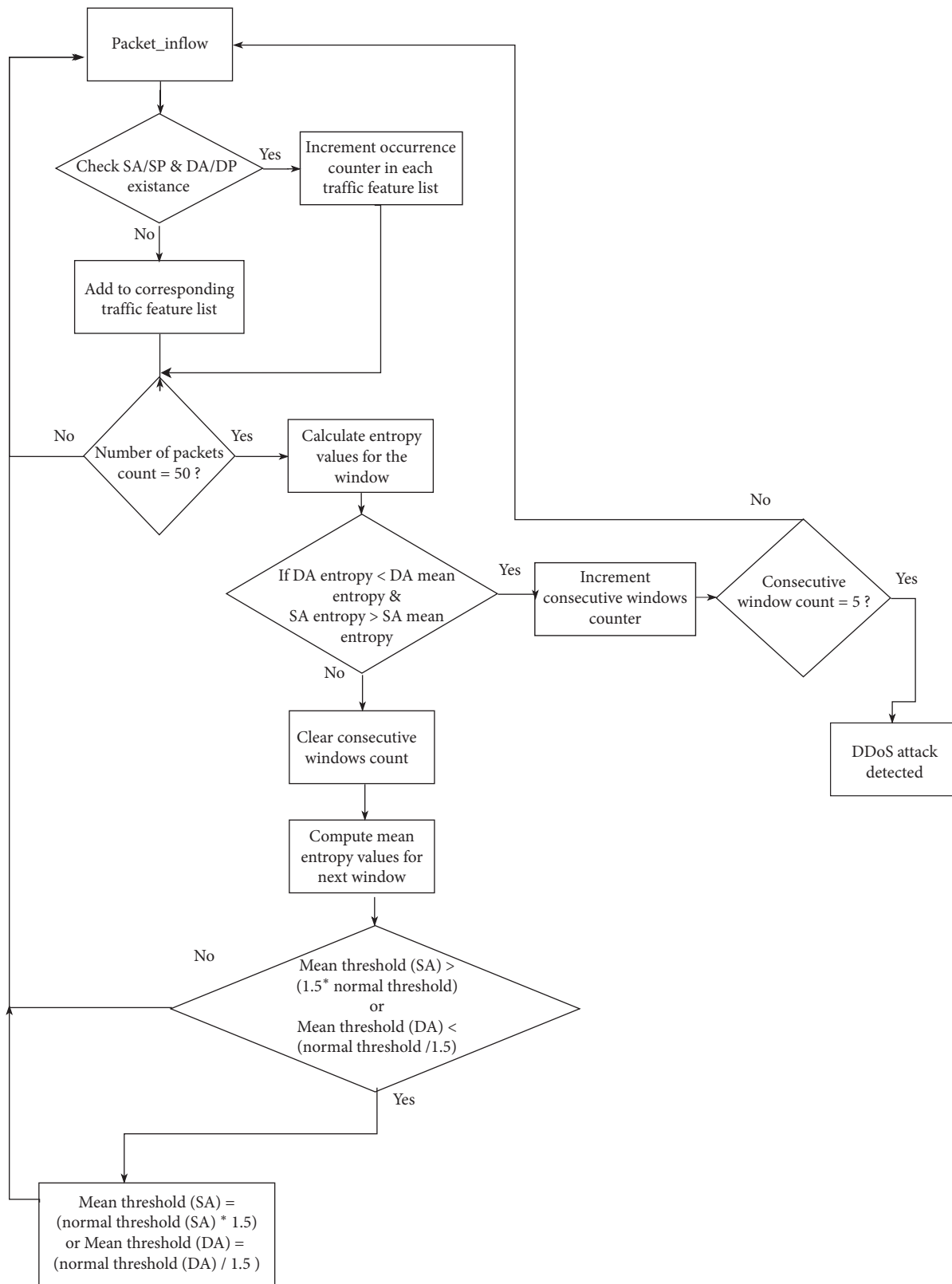


FIGURE 2: Proposed IDS: flow chart.

4. Experimental Setup

In this section, an experimental setup for validation of S-DPS against Utility-Consumer Communication Network implementation is discussed. For this purpose, a series of steps are followed in order to establish simulation for performing the experiments using test cases as discussed in the following section.

4.1. Simulation Steps

4.1.1. Controller. POX is used as SDN controller for the experiments. It is an open-source and python-based controller that is widely used in experiments. It is lightweight and developed as a platform to be customizable, meeting desired needs of a controller. It supports famous operating system like Windows, Linux, and MAC OS and has a network discovery feature installed. Apart from this, another two famous controllers like Floodlight and Beacon are also available. However, in most SDN-based papers highlighted in literature review, NOX controller, a predecessor of POX, is used. So, for the research POX controller is selected.

4.1.2. Network Emulator. Mininet 2.2.2 is used as a network emulator for the experiment. It is an open-source platform with support for SDN environment and OF protocol. It treats each network component as a kernel process and can be installed easily on a laptop or Personal Computer (PC) using kernel namespace feature. Each network namespace has its own Network Interface Card (NIC), Address Resolution Protocol (ARP) table, ping service, scripts, and routing table. Both Graphical User Interface (GUI) and command line interfaces are available to create network topologies. As a default, NOX controller is embedded in Mininet.

4.1.3. Traffic Generation. Scapy is used as a traffic generator tool, both for normal and for attack traffic. It has features of scanning, packet spoofing, packet forging, sniffing, etc. Here, TCP packets are generated using the tool. It supports python programming language and POX controller also uses python. So, both controller and traffic generation tool can be integrated. Spoofed source IP addresses and Host IP addresses are generated using python function "random."

This function returns a uniform random float in the range of 0.0 to 1.0. These random floats are joined together to form a spoofed IP address. Other options, i.e., type of packets and packets interval available in Scapy, are used to create normal and attack traffics. TCP/UDP/ICMP is set for type of packets and 0.4 seconds as interval for normal traffic between smart meter and utility server. Moreover, TCP/UDP/ICMP-based DDoS attacks with attack rates ranging between 200 and 4000 packets/sec are simulated in existing researches, i.e., [6, 20, 27, 28]. Such variations of traffic generation are catered for in existing experiments, covering both LR- and HR-DDoS attacks.

4.1.4. Network Setup. Network is set up on Laptop Dell Inspiron with Core i3 2.41 Ghz processor, 4 GB RAM, and 100/1000 Gbps NIC. Operating System is Windows 8.1 with VirtualBox 6.0.4 installed. Mininet 2.2.2 on Linux Ubuntu 14.04.4 is installed in the VirtualBox for setting up the environment. Mininet 2.2.2 supports OF version 1.3. Moreover, "mn" command is used in Mininet to set up the network. As a default, two hosts with one switch are configured. However, custom network is set up using different filters available in "mn" command, i.e., related to controller either local or remote, type of switch, number of hosts/switches, etc.

4.1.5. Network Topology. A tree-type network constituting smart-meter-utility server communication network is depicted in Figure 3. It has a depth of 2 with 10 switches and 54 hosts (smart meters and utility server). Here "smart meters" are the core of SGs because they are smart and possess the ability to sense, measure, and examine the usage of electricity, continuously transmit the data and information collected to the central location, and perform two-way communications with all other components of the SG and the consumer. Meanwhile, "utility server" has a dual-role to play; i.e., it has a two-way communication with smart meter as well as with power generation facility. It is located at control center and provides live consumption data to both users and to power generation facility. Finally, "controller" is the brain of the overall network managing all OF-enabled switches/routers by installing forwarding rules and performs centralized network and configuration management for better performance and security in the network.

Utility server is connected to Switch-1, whereas Switches 2–10 are used to connect 53 smart meters, evenly divided, i.e., 6 smart meters each. However, last switch consists of 5 smart meters. OF-enabled v-Switch available in Mininet is used to connect hosts. L3-learning module of POX controller with addition of two functions, i.e., traffic feature collection and entropy calculation, is used for the controller function of the network.

4.2. Evaluation Criteria. The S-DPS is evaluated using DR and FPR metrics where DR measures portion of the attacks that are detected correctly by the system and represented by (6) and FPR provides the percentage of events that are reported as negative events where actually they are positive events and represented by (7). In the equations, True Positive (TP) event means that the system has detected a correct anomalous event, whereas False Positive (FP) means system has detected an incorrect anomalous event; i.e., actually the event is legitimate but detected otherwise. Similarly, True Negative (TN) event means that the system has detected a correct legitimate event, whereas False Negative (FN) means system has detected an incorrect legitimate event, i.e., actually the event is anomalous but detected otherwise. Varied levels of both LR- and HR-DDoS attacks, i.e., smurf, socket stress, and SYN flood attacks, are launched against the utility server for early detection and real-time mitigation of attack.

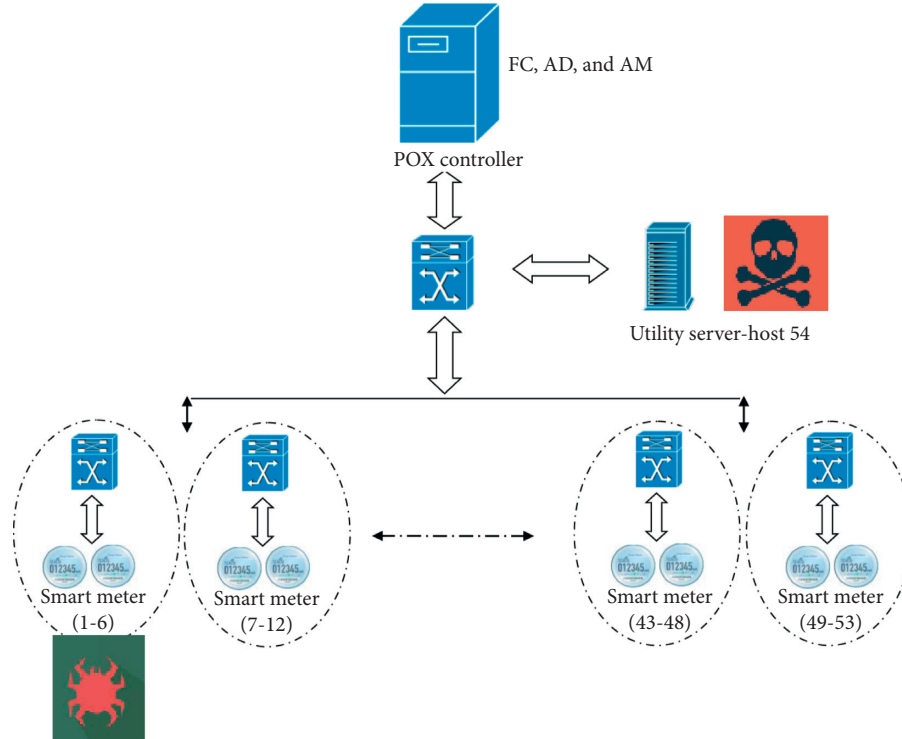


FIGURE 3: Network topology.

$$DR = \frac{TP}{TP + FN}, \quad (6)$$

$$FPR = \frac{FP}{TN + FP}, \quad (7)$$

5. Results and Discussion

The experiment covers topology highlighted in Figure 3, which contains 10 switches and 54 hosts. Each host from h1-h53 represents a smart meter, where host h54 is a utility server with which each smart meter sends its observed values. Each switch in the topology is OF-enabled v-Switch centrally connected to POX controller c0. In order to simulate the traffic between connecting entities, certain parameters like frequency of communication between smart meter and utility server, type of protocol, etc., need to be defined. AMI infrastructure does not have any standardized architecture and varying implementations exist defining the network and dynamics of communication. Frequency of communication between smart meter and utility server is also set to different intervals, i.e., 1 second, 4 seconds, 60 seconds, 5 minutes, and 15 minutes, depending upon the scheduling criteria set by utility service provider [29]. Considering the periodic traffic profile in most common architectures, smart meters are scheduled to transmit and receive at interval of 0.4 seconds [30]. Further, both UDP and TCP protocols are used in two-way communication between smart meter and utility server. Seven sets of traffic profiles are generated in the experiment, i.e., normal traffic, smurf attack, socket stress attack, and SYN flood attack. Traffic profile for the experiments is shown in Table 5. These traffic profiles are

simulated using UDP/TCP/ICMP-based packets at destination port 80/21 using random spoofed source IP addresses and source ports.

5.1. Normal Traffic Profile. In normal traffic profile, a total of 5 runs of experiment are performed, each containing 1250 packets with window size of 50 packets. Packet interval between smart meter(s) and utility server and reverse is set to 0.4 seconds. In normal circumstances, at any given point in time, a utility server is sending probe to any smart meter and any smart meter is sending its readings to utility server.

Therefore, two-way packets are generated randomly using one of the IP addresses of smart meter and of utility server with interval of 0.4 seconds to obtain average normal entropy value. The whole experiment is covering observation of 6,250 packets. The results of normal traffic separately for source IP (SrcIP) and destination IP (DestIP) are presented in Figure 4. Here, average entropy values per window for both SrcIP and DestIP are used to plot the graphs. As can be seen from Figure 4, entropy value for DestIP ranges from 1011.923 to 1372.990 and average normal entropy value is being utilized as a base entropy for DestIP in attack scenarios. Similarly, entropy value for SrcIP ranges from 1066.081 to 1722.328 and average normal entropy value is 1320.678, being utilized as a base entropy for SrcIP in attack scenarios.

5.2. Smurf Attack. A smurf attack is a type of DDoS attack in which vulnerabilities in Internal Protocol (IP) or Internal Control Message Protocols (ICMP) are exploited as such that it makes the overall computer network inoperable. For

TABLE 5: Traffic profiles.

Type of traffic	Protocol	DP	SP	Payload: number of packets	Source IP address	Destination IP address	Traffic interval	Attack type
Normal	UDP	80/21	2/3	None	10.0.0.54 or random (10.0.0.0/24)	10.0.0.54 or random (10.0.0.0/24)	0.4 sec	—
Smurf attack	ICMP	—	—	6000 bytes	10.0.0.54	10.0.0.255	—	DDoS
Socket stress	TCP	21	Random (0–65535)	None	10.0.0.4	10.0.0.54	—	DDoS
SYN flood	TCP	80	Random (1000–9000)	None	Random	10.0.0.54	—	DDoS

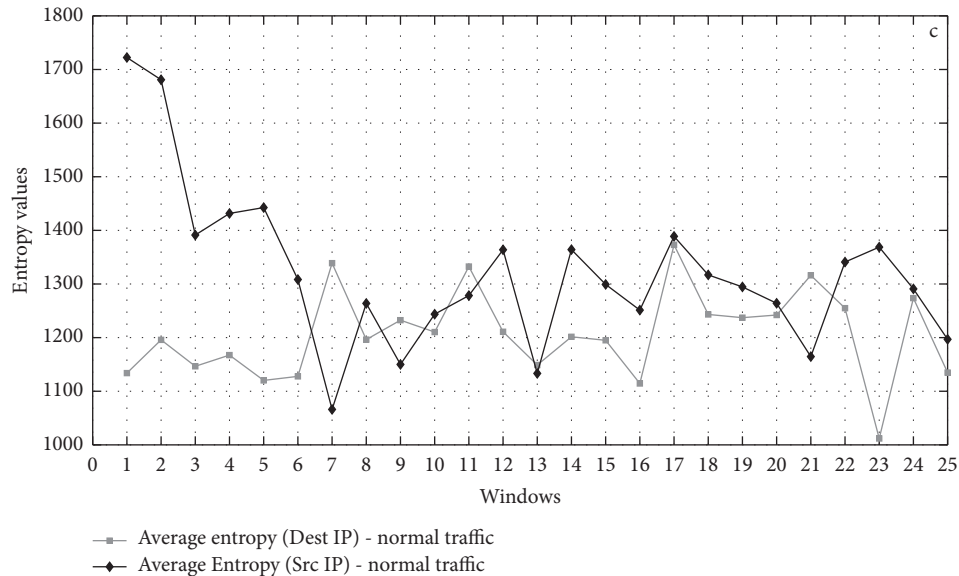


FIGURE 4: Normal traffic profile.

smurf attack to work, a false IP packet with spoofed IP is created. IP packet is basically an ICMP ping message that tells the network nodes to receive and send back echo reply. These echoes are then sent back to all network devices creating an infinite loop in the network. To further amplify the attack, IP broadcasting technique can be used.

In the experiment, an ICMP echo request is generated towards the broadcast address of all switches/routers, i.e., 10.0.0.255 using the spoofed IP address, i.e., of target address (10.0.0.54), which is a utility server. In this case, all the smart meters lying under these switches/routers will send their ICMP echo replies towards the target, i.e., utility server. In order to further amplify the attack, each smart meter is relaying 6000 bytes of junk IPv4 packets towards the target. Two separate scripts are being run manually using two random hosts, e.g., h1 and h4. At h1, normal traffic generation is carried out, whereas at h4 (attacker machine), smurf attack towards target address (utility server) is launched. Traffic profiles both for source and destination IP for the scenario are depicted in Figure 5. It can be seen from Figure 5(b) that destination IP current entropy is far below the threshold value between windows 6 and 25, meaning the number of packets with same DestIP/window, i.e., towards the target host, has increased exceptionally resulting in decrease of overall DestIP address entropy. So, the attack is detected in

these windows. Further, to verify whether it is a DoS or DDoS attack it can be seen from Figure 5(a) that source IP current entropy is above the threshold value between windows 11–22, meaning the number of packets with multiple SrcIPs/windows for the target host exists, resulting in increase of overall SrcIP address entropy from the threshold. Therefore, the attack detected is DDoS. In case it is below the threshold values, the attack is considered as DoS attack.

Moreover, comparison between the normal and attack traffic for destination IP is depicted in Figure 5(c). It can be seen that entropy values/window for attack traffic has declined significantly compared to normal traffic since most of the traffic/window is directed towards a single DestIP, resulting in decline of entropy. Moreover, it can be observed from Figures 5(a) and 5(b) that S-DPS-based threshold is changing as per the current network conditions compared to experimental static threshold that remains fixed no matter how much the network environment varies. So, S-DPS-based threshold is able to provide true picture of the network while achieving DR of 100% with 0% FPR for simulated traffic.

5.3. Socket Stress Attack. Considering socket stress attack, raw sockets are used to establish a connection with the target machine. It is an asymmetric resource consumption attack,

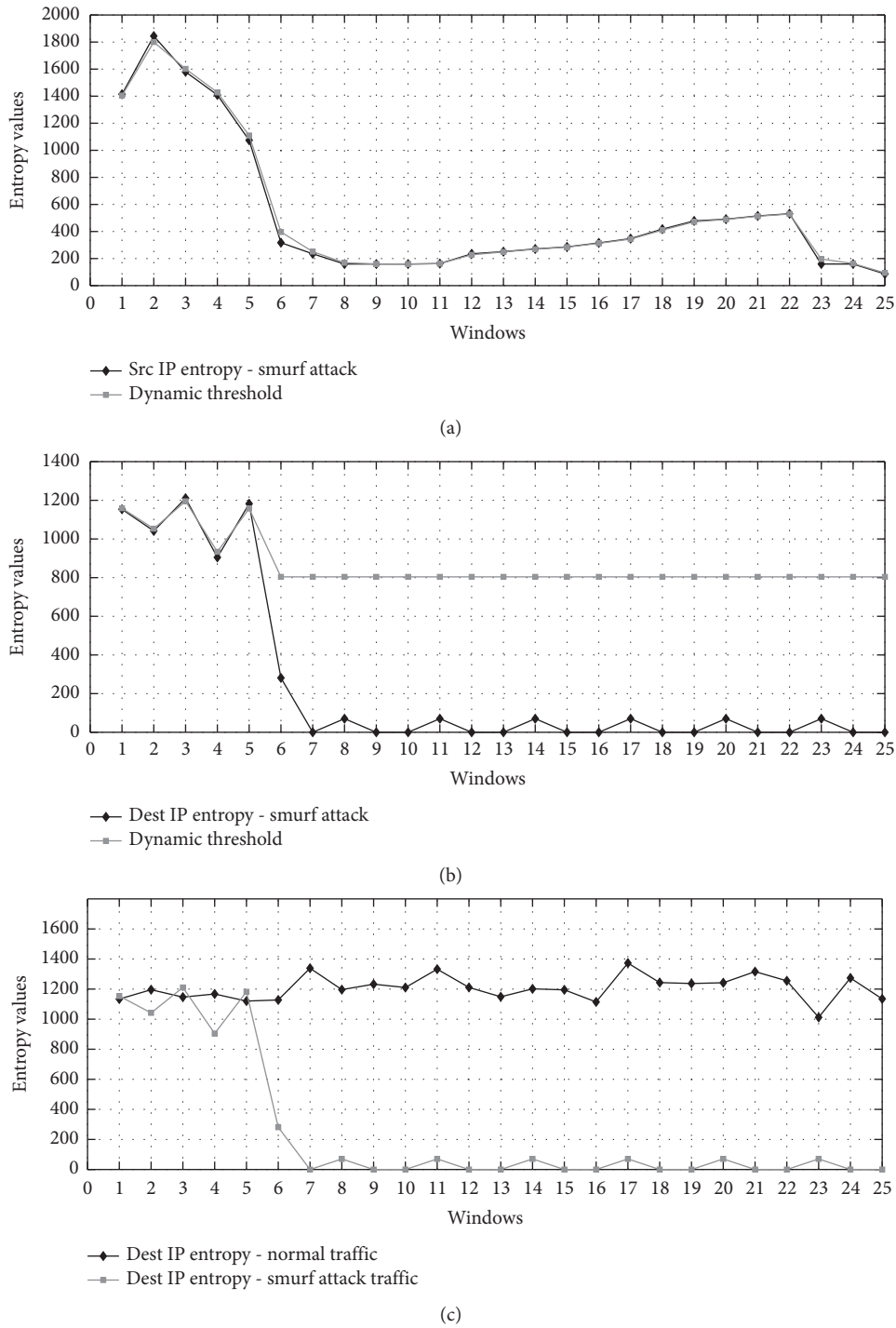


FIGURE 5: Smurf attack detection- static vs. dynamic thresholds. (a) Source IP address traffic profile; (b) destination IP address traffic profile; (c) destination IP-normal vs. smurf attack traffic.

where asymmetric refers to less requirement of resources at attacker end versus a great deal of resource consumption on target machine. For such attack to work, it should be targeted to an open port in victim’s machine. In the attack, attacker advertises a zero window at the end of three-way handshake; meaning it has not received the data so the victim will tend to open the connection and probe the client periodically to check whether data is received or not.

Similarly, multiple connections at the victim machine are opened, consuming many resources on the victims’ machine. Socket stress attack script is executed randomly on h4 (*attacker machine*) targeting utility server (*victim machine*) at IP address 10.0.0.54 and port 80. In the attack, 20 random connections using random source ports are created with a timeout value of 1 minute. Timeout value defines the time before which new connection is established to the target. So,

at any given point in time, a minimum of 20 connections remain active on the target machine. Two separate scripts are being run manually using two random hosts, say h1 and h4. At h1, normal traffic generation is carried out, whereas at h4, socket stress attack towards target address is launched. Traffic profiles both for source and destination IP for the scenario are depicted in Figure 6. It can be seen from the figure that destination IP current entropy is far below the threshold value between windows 5 and 25, meaning the number of packets with same DestIP/window, i.e., towards the target host, has increased exceptionally resulting in decrease of overall DestIP address entropy. So, the attack is detected in these windows. Further, to verify whether it is a DoS or DDoS attack, it can be seen from Figure 6(b) that source IP current entropy is not above the threshold value for consecutive windows from windows 1 to 25. It means that the number of packets with single SrcIP/window for the target host exists, resulting in decrease of overall SrcIP address entropy from the threshold. Therefore, the attack detected is DoS. Moreover, comparison between the normal and attack traffic for destination IP is depicted in Figure 6(c). It can be seen that destination IP entropy values/window for attack traffic has decreased significantly after the attack compared to normal traffic using the proposed S-DPS mechanism since most of the traffic/window is directed towards a single DestIP, resulting in decline of entropy.

5.4. SYN Flood Attack. In case of SYN flood attack, the attacker exploits part of normal TCP three-way handshake process by sending repeated SYN packets to the target machine with a frequency above its capacity to process. It can target all open ports or a specific port to block the service(s) of the target machine. The target machine responds to all received requests with SYN-ACK packets for that open port(s) and wait for ACK packets for some time. In most scenarios, source IP address and ports are malicious, i.e., spoofed, so ACK packets are never sent back or, in another case, ACK packets are not sent by the attacker deliberately to shut down the service of target machine. SYN flood attack script is executed randomly on h4 (*attacker machine*) targeting utility server (*victim machine*) at IP address 10.0.0.54 and port 80. In the attack, 10,000 packets with random source IP address and ports (ranging between 1000 and 9000) are sent to the utility server. These packets have random “seq” numbers and “window” size between 1000 and 9000. Two separate scripts are being run manually using two random hosts, say h1 and h4. At h1, normal traffic generation is carried out, whereas at h4 (*attacker machine*), SYN flood attack towards target address (*utility server*) is launched. Traffic profiles both for destination and source IPs for the scenario are depicted in Figure 7. It can be seen from Figure 7(a) that destination IP current entropy is far below the threshold value between windows 5 and 25; meaning the number of packets with same DestIP/window, i.e., towards the target host, has increased exceptionally resulting in decrease of overall DestIP address entropy. So, the attack is detected in these windows. Further, to verify whether it is a DoS or DDoS attack, it can be seen from Figure 7(b) that

source IP current entropy is above the threshold value for consecutive windows from windows 6–25, meaning the number of packets with multiple SrcIPs/windows for the target host exists, resulting in increase of overall SrcIP address entropy from the threshold. Therefore, the attack detected is DDoS. Moreover, comparison between the normal and attack traffic for destination IP is depicted in Figure 7(c). It can be seen that destination IP entropy values/window for attack traffic has decreased significantly after the attack compared to normal traffic using the proposed S-DPS mechanism since most of the traffic/window is directed towards a single DestIP, resulting in decline of DestIP entropy.

For all the attacks discussed above, although the target is *utility server (h54)*, controller being the brain of SDN network is processing all the normal and attack packets. So, in addition to utility server (*target machine*) controller is also being targeted in all attack scenarios discussed, but the detection and mitigation approach is implemented at the controller so attack is mitigated within near real time.

5.5. Mitigation of DDoS Attack. On detection of DDoS attack, it is important to mitigate it as well to prevent its penetration further in the network. OF protocol, due to its real-time reconfiguration feature, enables us to define flow rules that can block the switch ports in real time. The authors in [6, 27, 28] utilized OF port blocking or deletion of flows as a mitigation strategy, achieving time and space complexity of $O(n)$, where “ n ” may be number of packets processed for port blocking or number of flows deleted. For that matter, port blocking strategy is utilized, achieving the same complexity of $O(n)$. One timer variable of Boolean type, i.e., “timerSet,” and two functions, i.e., Preventing() and _timerfunc(), are incorporated in the default L3_learningmodule of POX controller. By default, timerSet is set to “False” so that controller continues to operate without active DDoS defense mechanism till entropy of the window does not fall under threshold value of that window. In case entropy values of the windows from the entropy dictionary are less than the threshold values, Preventing() function is invoked with global Set_Timer set to “True”; otherwise, timerSet value is set to “False” to enable/allow normal operation of the controller again, i.e., without active DDoS defense mechanism. Eventually, _timer_func is used to detect the happening of DDoS attack using the dictionary maintained by Preventing() function and block the switch ports with count greater than and equal to 5, occurring in five consecutive windows. Preventing() function is incorporated in POX controller using “_handle_openflow_packetIn” instance. Each time a new packet enters the controller, packet is accounted for in the dictionary being maintained. Dictionary constitutes a switch ID and port number with its counter. It has a form like switch ID (port number, count). Switch ID is recognized by OF parameter “event.connection.dpid” and port number by “event.port.” It is used to detect whether DDoS attack has occurred or not. After creating the dictionary for 25 windows, _timer_func() is used to detect and mitigate

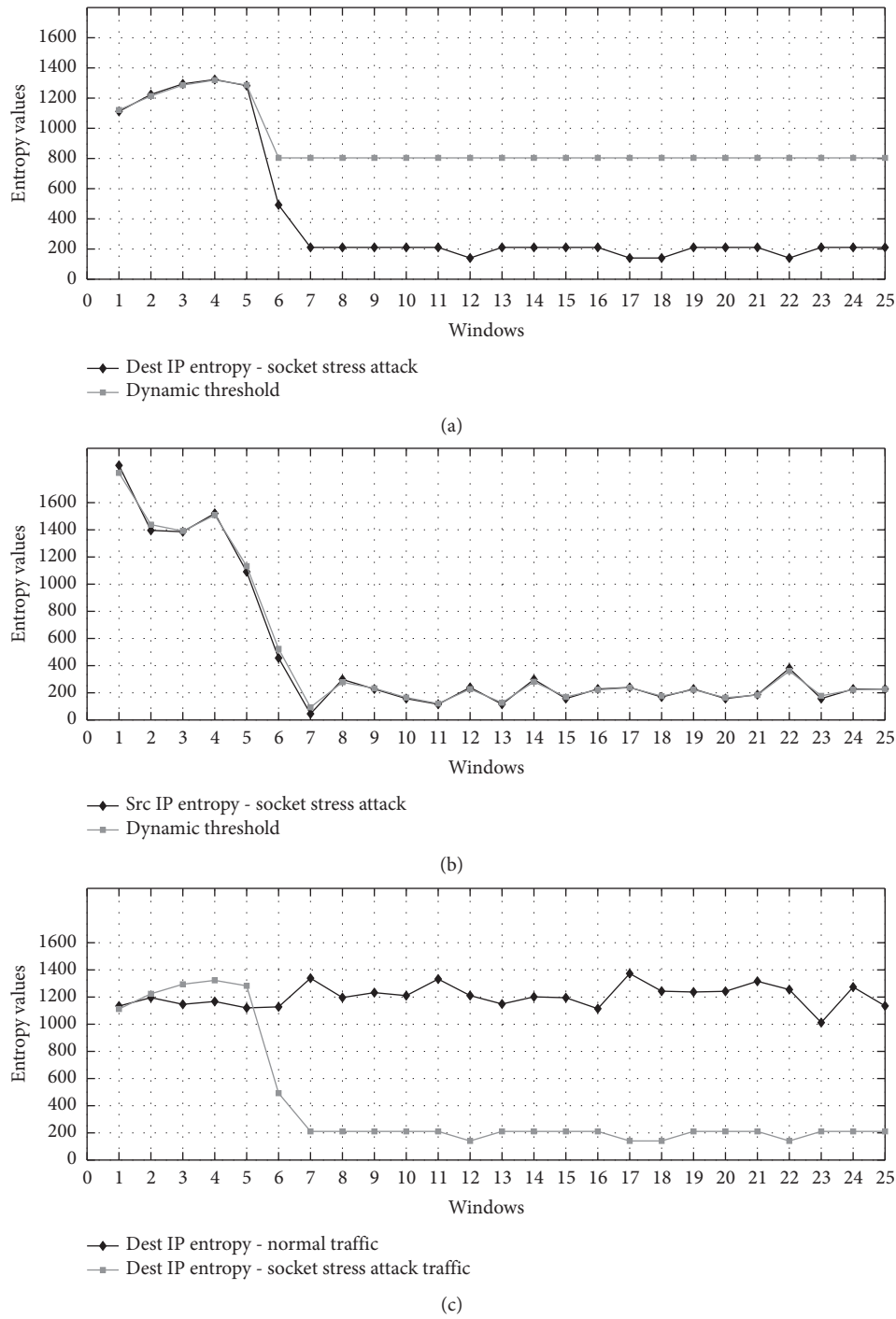


FIGURE 6: Socket stress attack detection-static vs. dynamic thresholds. (a) Destination IP address traffic profile. (b) Source IP address traffic profile. (c) Destination IP-normal vs. socket stress attack traffic.

DDoS attack. It iterates through all the items in the dictionary and if specific ports of a specific switch have its count greater than and equal to 5 and for five consecutive windows, DDoS attack is detected and these switch ports are blocked by sending message to controller using OF

procedure calls, i.e., “*of.of p_packet_out*” and “*core.open_flow.sendToDPID()*.” The mitigation strategy is performed successfully on 25% rate attack on single host. As per results, dictionary maintained by the controller contains count of 69 for Switch-1 and Port-1, 12 for Switch-2 and

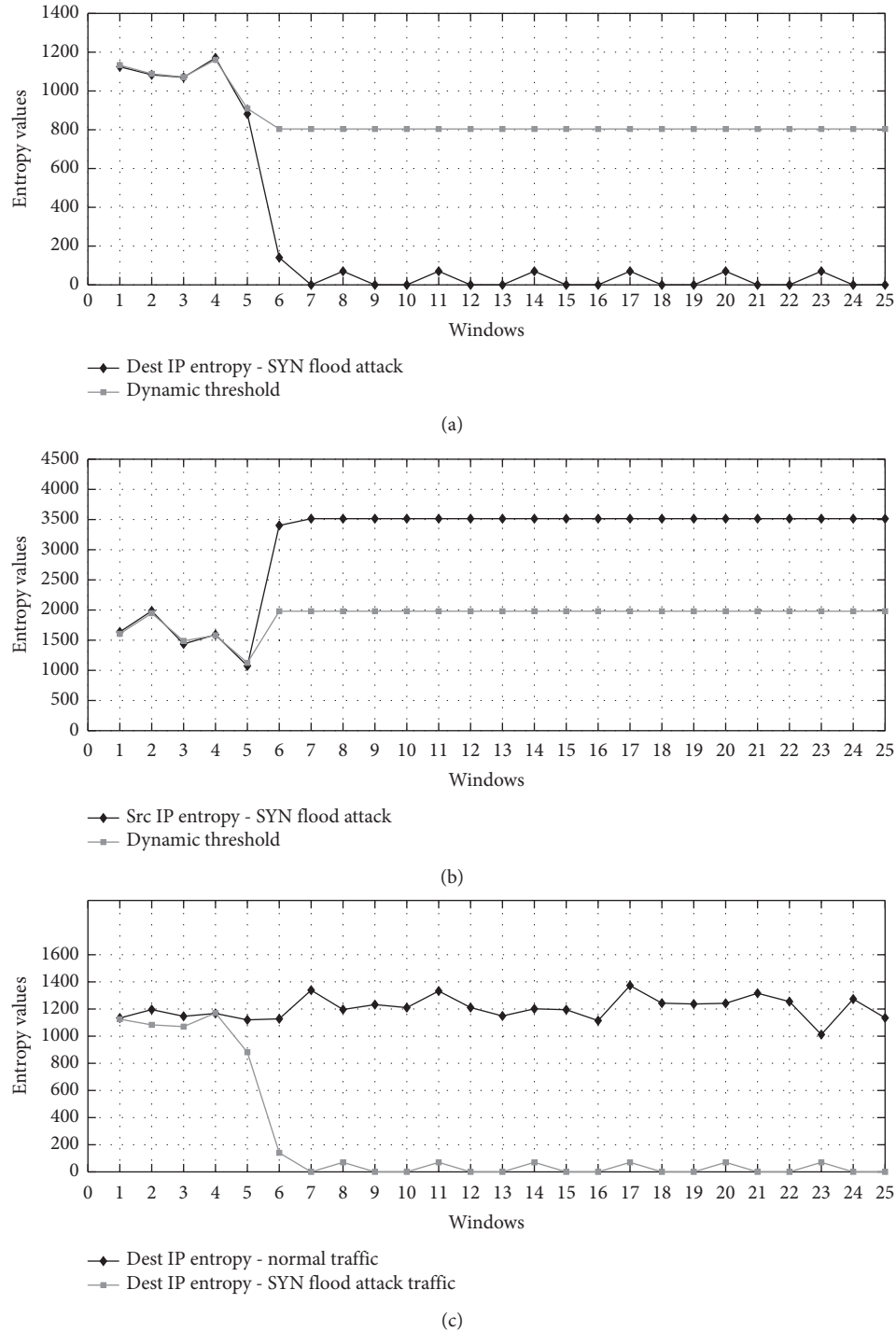


FIGURE 7: SYN flood attack detection-static vs. dynamic thresholds. (a) Destination IP address traffic profile; (b) source IP address traffic profile; (c) destination IP-normal vs. SYN flood attack traffic.

Port-1, 60 for Switch-2 and Port-4, and 61 for Switch-8 and Port-9. All such switch ports are blocked by the controller as part of prevention strategy.

6. Performance Evaluation

The performance of S-DPS is evaluated using metrics like CPU/RAM utilization. CPU/RAM utilization is measured

and compared with and without the approach using 25% attack rate on single host scenario. As mentioned in previous sections of conceptual framework, 4 additional functions are added to the L3-learning module of POX controller, i.e., traffic feature collection, entropy calculation, timer function, and preventing function, for DDoS detection and mitigation purpose. In order to see the effect of these functions on the overall CPU/RAM utilization of Mininet and on the

controller, two simulations are run again. One simulation constitutes 25% rate attack on single host without the solution and other simulation with same setting with proposed solution. The elapsed time for both simulations is 25 seconds and normal traffic ran for 200 seconds. “Top” and “Htop” commands have been used to capture the CPU/RAM utilizations. Results are depicted in Table 6. It can be seen from Table 6 that overall CPU/RAM utilization is 55.5%/171 MB in case of simulation without the solution and controller instance has consumed 12.3%/1.4% of total memory. In case of simulation with the solution, overall CPU/RAM utilization is 55.2%/205 MB and controller instance has consumed 29.6%/1.7% of total memory. There is a slight increase in controller instance CPU/RAM utilization but it is still in acceptable limits.

DDoS detection and mitigation functions are incorporated in SDN controller, considering the low computational complexity of approach used, i.e., $O(n)$ for both time and space complexity. It is verified by CPU/RAM utilization with and without the approach. At controller end, CPU utilization rises to only 29.6% from 12.3% with S-DPS. Similarly, there is a minimal increase in RAM utilization from 1.4% to 1.7%. Considering the facts, S-DPS can be both efficient and effective approach to provide DDoS protection in dynamic networks like SG communication network. The reason is its nondependency on any training requirements and due to adaptive nature of threshold calculations.

Several approaches to DDoS detection exist in literature. For example, Self-Organizing Maps (SOM), a machine learning approach, has been used by [31] to learn the behavior of network and decide whether network is attacked or not. Several hours of learning is required for better DR and FPR. In case of network or topology change, SOM is required to be trained again. With expansion of network, neurons used in SOM are also required to be increased, making the solution more expansive towards the network. The S-DPS is built in inside the controller and is easily adaptable to the changing network. No training is required upfront and computational complexity is lower than machine learning approach—SOM. Similarly, the authors in [32, 33] have utilized SNORT alongside SDN for DDoS detection. As highlighted previously, S-DPS has achieved better CPU/RAM utilization compared to SNORT. Moreover, DDoS protection mechanism is embedded in S-DPS, where in [33] separate SNORT detection system is integrated with SDN environment making it less transparent towards computational overhead, sampling requirements, and bandwidth limitations, if any. Both SOM and SNORT apply complex operations to learn the behavior of the network, e.g., processing large matrices or pattern matching schemes. In S-DPS, entropy-based mechanism is providing the same functionality without any of the complexities available in SOM and SNORT.

Benefits that are achieved through S-DPS are highlighted as follows:

- (i) High DR with no FPR
- (ii) DoS, LR-DDoS, and HR-DDoS attacks that have been successfully detected

TABLE 6: Resource utilization.

Resource utilization	Controller instance		Mininet instance	
	CPU%	RAM%	CPU%	RAM (Mbs)
Without S-DPS	12.3	1.4	55.5	171
With S-DPS	29.76	1.7	55.2	2.5

- (iii) Threshold mechanism that is adaptive rather than static and without any experimental adjustment for better DR/FPR, thus making it more suitable for modern/dynamic networks
- (iv) DDoS mitigation mechanism also provided as an addition for real-time protection

7. Conclusion

Given the nature of current dynamic networks, DDoS attacks are constantly becoming more sophisticated and are rapidly growing. These attacks can prove to be devastating for the underlying networks, with special emphasis to communication networks existing in SGs. The conventional approaches to counter these attacks are not enough to provide sufficient safety. Many researchers have claimed that the evolving SDN-based approaches are successful in dealing with the DDoS attacks in such networks. But, it is reported that these approaches employ the static threshold mechanisms to detect the attacks which is not suitable, given the dynamic and heterogeneous networks of SGs. S-DPS has claimed to efficiently address and manage these issues by employing an SDN-based environment and using a light-weight entropy-based defense mechanism. The DDoS protection strategy is made more efficient and effective through the reconfiguration of network in real time and by providing the global view of SDN networks. It is capable of detecting the threat along with the mitigation of anomaly at the same time as early as the first 250 packets by blocking the ports. Additionally, the existing SDN-based approaches are unable to detect different level of DDoS attacks but with the use of Tsallis entropy and its sensitivity factor, detection becomes possible. DR of 100% with FPR of 0% is achieved through simulation of HR-DDoS attacks. The S-DPS is able to show its capability and productiveness in both protection against DDoS and computational costs through minimum usage of CPU and RAM.

7.1. Future Works. Single controller architecture is utilized in S-DPS, making it vulnerable to computational/bandwidth bottlenecks for very large networks. In order to add resiliency in S-DPS, a multicontroller architecture is recommended. Intercontroller communication mechanism is necessary to provide synchronized operations of the protection system, with necessary recovery and failsafe mechanism.

Data Availability

The data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Natural Science Foundation of China (NSFC) under grant no. 62072217.

References

- [1] D. Jin, Z. Li, C. Hannon et al., "Toward a cyber resilient and secure microgrid using software-defined networking," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2494–2504, 2017.
- [2] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the smart grid: a comprehensive compilation of intrusion detection and prevention systems," *IEEE Access*, vol. 7, pp. 46595–46620, 2019.
- [3] H. Maziku, S. Shetty, and D. M. Nicol, "Security risk assessment for SDN-enabled smart grids," *Computer Communications*, vol. 133, pp. 1–11, 2019.
- [4] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: threats and potential solutions," *Computer Networks*, vol. 169, Article ID 107094, 2020.
- [5] M. Abujubbeh, F. Al-Turjman, and M. Fahrioglu, "Software-defined wireless sensor networks in smart grids: an overview," *Sustainable Cities and Society*, vol. 51, p. 101754, 2019.
- [6] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and 600s Flow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 601, pp. 122–136, 2014.
- [7] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of Things: evolution and technologies from a security perspective," *Sustainable Cities and Society*, vol. 54, Article ID 101728, 2019.
- [8] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "A systematic review of data protection and privacy preservation schemes for smart grid communications," *Sustainable Cities and Society*, vol. 38, pp. 806–835, 2018.
- [9] M. A. Ferrag, M. Babaghayou, and M. A. Yazici, "Cyber security for fog-based smart grid SCADA systems: solutions and challenges," *Journal of Information Security and Applications*, vol. 52, p. 102500, 2020.
- [10] S. K. Singh, Y. S. Jeong, and J. H. Park, "A deep learning-based IoT-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, Article ID 102252, 2020.
- [11] F. A. Khan, M. Asif, A. Ahmad, M. Alharbi, and H. Aljuaid, "Blockchain technology, improvement suggestions, security challenges on smart grid and its application in healthcare for sustainable development," *Sustainable Cities and Society*, vol. 55, Article ID 102018, 2020.
- [12] M. Samir, M. Azab, M. R. Rizk, and N. P. Y. G. R. I. D. Sadek, "A software development and assessment framework for grid-aware software defined networking," *International Journal of Network Management*, vol. 28, p. e2033, 2018.
- [13] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: methods, practices, and solutions," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 425–441, 2017.
- [14] O. Jung, P. Smith, J. Magin, and L. Reuter, "Anomaly detection in smart grids based on software defined networks," vol. 1, pp. 157–164, in *Proceedings of the 8th International Conference on Smart Cities and Green ICT Systems*, vol. 1, pp. 157–164, SMARTGREENS, Heraklion, Crete, May 2019.
- [15] W. Chen, S. Xiao, L. Liu, X. Jiang, and Z. Tang, "A DDoS attacks traceback scheme for SDN-based smart city," *Computers & Electrical Engineering*, vol. 81, p. 106503, 2020.
- [16] X. Min, Z. J. Jin, R. H. Yang, M. R. Heng, and C. M. Xin, "Traffic scheduling strategy of power communication network based on SDN," *Computer Engineering and Applications Journal*, vol. 9, pp. 49–60, 2020.
- [17] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generation Computer Systems*, vol. 81, pp. 557–565, 2018.
- [18] S. Ali and Y. Li, "Learning multilevel auto-encoders for DDoS attack detection in smart grid network," *IEEE Access*, vol. 7, pp. 108647–108659, 2019.
- [19] Z. Ahmed, N. Afaqui, and O. Humayan, "Detection and prevention of DDoS attacks on software defined networks controllers for smart grid," *International Journal of Computer Applications*, vol. 975, p. 8887, 2019.
- [20] T. Pandikumar, F. Atkilt, and C. A. Hassen, "Early detection of DDoS attacks in a multi-controller based SDN," *International Journal of Engineering Science*, p. 13422, 2017.
- [21] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "A multi-step outlier-based anomaly detection approach to network-wide traffic," *Information Sciences*, vol. 348, pp. 243–271, 2016.
- [22] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "E-LDAT: a lightweight system for DDoS flooding attack detection and IP traceback using extended entropy metric," *Security and Communication Networks*, vol. 9, no. 16, pp. 3251–3270, 2016.
- [23] A. A. Amaral, L. d. S. Mendes, B. B. Zarpelão, and M. L. P. Junior, "Deep IP flow inspection to detect beyond network anomalies," *Computer Communications*, vol. 98, pp. 80–96, 2017.
- [24] I. Aouini and L. B. Azzouz, "Smart grids cyber security issues and challenges," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 9, pp. 1255–1261, 2015.
- [25] R. A. Niazi and Y. Faheem, "A bayesian game-theoretic intrusion detection system for hypervisor-based software defined networks in smart grids," *IEEE Access*, vol. 7, pp. 88656–88672, 2019.
- [26] M. Ghanbari and W. Kinsner, "Detecting DDoS attacks using polyscale Analysis and deep learning," *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 14, no. 1, pp. 17–34, 2020.
- [27] P. Bull, R. Austin, E. Popov, M. Sharma, and R. Watson, "Flow based security for IoT devices using an SDN gateway," in *Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 157–163, IEEE, Vienna, Austria, August 2016.
- [28] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," vol. 1, pp. 310–317, in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, pp. 310–317, IEEE, Helsinki, Finland, 2015.
- [29] K. Balachandran, R. L. Olsen, and J. M. Pedersen, "Bandwidth analysis of smart meter network infrastructure," in *Proceedings of the 16th International Conference on Advanced Communication Technology*, pp. 928–933, IEEE, Pyeong-Chang, South Korea, February 2014.

- [30] P. Cheng, L. Wang, B. Zhen, and S. Wang, "Feasibility study of applying LTE to smart grid," in *Proceedings of the 2011 IEEE First International Workshop on Smart Grid Modeling and Simulation (SGMS)*, pp. 108–113, IEEE, Brussels, Belgium, October 2011.
- [31] E. Mota, A. Passito, and R. Braga, "Lightwight DDoS flooding attack detection using NOX/Openflow," in *Proceedings of the IEEE 35th Conference on Local Computer Networks*, IEEE, Denver, CO, USA, April 2010.
- [32] A. Mahajan, A. Gupta, and L. Sen Sharma, "Performance evaluation of different pattern matching algorithms of snort," *International Journal of Advanced Networking Applications*, vol. 10, no. 02, pp. 3776–3781, 2018.
- [33] D. Huang, L. Xu, C. Chung, and T. Xing, "SnortFlow: a openflow-based intrusion prevention system in cloud environment," in *Proceedings of the Second GENI Research and Educational Experiment Workshop*, Salt Lake, UT, USA, March 2013.

Research Article

Multiauthority Attribute-Based Encryption with Traceable and Dynamic Policy Updating

Jie Ling ¹, Junwei Chen ¹, Jiahui Chen ¹ and Wensheng Gan ²

¹School of Computer, Guangdong University of Technology, Guangzhou 510006, China

²College of Cyber Security, Jinan University, Guangzhou 510632, China

Correspondence should be addressed to Jiahui Chen; csjhchen@gmail.com

Received 9 December 2020; Revised 15 January 2021; Accepted 10 February 2021; Published 26 February 2021

Academic Editor: Prosanta Gope

Copyright © 2021 Jie Ling et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ciphertext policy attribute-based encryption (CP-ABE) is an encryption mechanism that can provide fine-grained access control and adequate cloud storage security for Internet of Things (IoTs). In this field, the original CP-ABE scheme usually has only a single trusted authority, which will become a bottleneck in IoTs. In addition, different users may illegally share their private keys to obtain improper benefits. Besides, the data owners also require the flexibility to change their access policy. In this paper, we construct a multiauthority CP-ABE scheme on prime order groups over a large attribute universe. Our scheme can support white-box traceability along with policy updates to solve the abovementioned three problems and, thus, can fix the potential requirements of IoTs. More precisely, the proposed scheme supports multiple authority, white box traceability, large attribute domains, access policy updates, and high expressiveness. We prove that our designed scheme is static secure and traceable secure based on the state-of-the-art security models. Moreover, by theoretical comparison, our scheme has better performance than other schemes. Finally, extensive experimental comparisons show that our proposed algorithm can be better than the baseline algorithms.

1. Introduction

With the help of cloud computing technology, Internet of Things (IoTs) [1] can bridge physical devices and virtual objects, which has become a promising networking scenario in the cyber world. In IoTs, more and more companies and individuals store data in the cloud, requiring the cloud servers to provide data access services. However, cloud servers are generally considered to be untrustworthy for the reason that the data of IoTs often contain sensitive information. In order to protect the privacy of these data, one of the traditional technologies is to encrypt the data, and data owners need to be online at all times to distribute their secret keys. Although these technologies achieve access control, the management of these keys will become a bottleneck when more and more users joined the system. In addition, for each type of data, it is necessary to maintain one or more copies of the ciphertext for different users with different keys, which will cause a waste of storage overhead in an IoTs system [2].

To this end, Sahai et al. [3] firstly proposed attribute-based encryption. The concept of attribute-based encryption (ABE) is a one-to-many encryption mechanism that can provide fine-grained access control and data security. Goyal et al. [4] further proposed the key policy ABE (KP-ABE) and ciphertext policy ABE (CP-ABE). Then, Bethencourt et al. [5] studied the CP-ABE scheme with a complete description, showing that CP-ABE allows data owners to define access strategies under the user's attributes. Once the user encrypts specific data, other users can decrypt them if and only if their attributes meet the access policy. Thanks to these characteristics, the CP-ABE scheme is considered a more suitable encryption mechanism for cloud storage access control than KP-ABE.

However, the original CP-ABE scheme only has a single, trusted authority dealing with the user's key distribution and attribute management, which will become a bottleneck in the cloud, especially in an IoTs system. Liu et al. [6] proposed a scheme under a different hierarchy of attributes with the

name of ciphertext-policy hierarchical attribute-based encryption. Deng et al. [7] elaborate on ABE and propose a new versatile cryptosystem referred to as ciphertext-policy hierarchical ABE. Wang et al. [8], based on the access structure layered model, proposed a novel access control scheme about file hierarchy by using ABE to solve the problem. Liu et al. [9] propose a novel T-CP-ABE system that gives high policies expressiveness in any monotone access structures and add traceability. Liang et al. [10] propose a CP-ABPRE to deal with the security problem by using the dual system encryption technology with the selective proof technique. But, the schemes mentioned above are all single attribute authorization (AA) ABE schemes. It is completely borne in the cloud environment, which not only brings a serious burden to the authorization center but also requires the authorization center to be completely trusted. Single-attribute authority cannot meet the development needs of practical applications because different attributes in different fields in many application scenarios are caused by different environments. For example, there is a situation that the data owner wants to share data with the researchers in the research institutes and the managers in the government departments. In this case, the attributes of researchers are determined by the research institutes. At the same time, the “government attributes” are managed by the government department. The abovementioned ABE schemes are not suitable for this situation where the attributes need to be managed by multiple agencies.

On the other side, in some CP-ABE schemes, it is easy to discover their attributes in the private key. There may be another situation that some malicious users illegally share their private keys to obtain economic benefits. Thus, the features of the CP-ABE scheme that can track leaked secret keys are particularly important. Therefore, we also need a traceability mechanism to track these malicious users. For example, attackers can access critical vulnerabilities in a wide variety of IoTs applications and devices to perform their malicious activities. This requires the design of effective security mechanisms in an IoTs-related application.

Except for the traceability, the policy update of the CP-ABE system also needs to be considered for supplying more functions. For instance, when addressing security, trust, and privacy in IoTs, the data owner may need to alter the access policy stored on the cloud. In that case, the traditional solution is to let the data owner find the cloud storage server’s relevant ciphertext and decrypt it, then encrypt the ciphertext using a new access strategy, and upload the newly encrypted ciphertext back to the cloud server. It, thus, brings much computational burden to the system. Therefore, the policy update is another important characteristic of the actual system.

To sum up, there are three major challenges in CP-ABE that we need to solve as follows:

- (1) How to solve the bottleneck of single authority authorization in cloud storage applications, especially in an IoTs system?
- (2) How to prevent some malicious users from illegally sharing their private keys?

- (3) How to propose an algorithm that makes the data owner’s access control more flexible in IoTs-enabled applications?

1.1. Our Contribution. This paper addresses the above-mentioned challenges by proposing a scheme named T-DPU-MCP-ABE (Traceable and Dynamic Policy Updating Multiauthority Attribute-based Encryption). More precisely, we propose a T-DPU-MCP-ABE based on the prime order bilinear group, and we prove its static security and resistance to traceable attacks under two related security models. Our security assumption utilizes the q -type hypothesis [11] and is based on the LRSW hypothesis [12]. As far as we know, we are the first one to support the properties of large attribute domain, policy update, white box traceability, multiauthorization, and high expressiveness and still have good performance. Especially, the features are described in detail as follows:

- (1) Large attribute domain: the size of public parameters is affected by the number of authorized institutions and will not increase linearly with the number of attributes. There is no need to determine the system attribute domain when the system is established.
- (2) Policy update: data owners may often need to modify the ciphertext access policy according to various requirements. Policy updates provide flexibility and allow data owners to adjust their encrypted data access policies to achieve fine-grained control.
- (3) White box Traceability: it can track malicious users who illegally share private keys. Through white box tracking that does not need to maintain a user list, the efficiency of the solution is improved, and no additional storage overhead is consumed.
- (4) Multiple authorized authorities: multiple authorized authorities undertake the key distribution work and, thus, reduce the workload and solve the problem of incomplete trustworthiness of the single authority.
- (5) High expressiveness: supports flexible access control and supports any monotonous access structure access strategy.

1.2. Organization. The rest of this paper is arranged as follows. In Section 3, we introduce the necessary background knowledge. In Section 4, we give the formal definition and security model of auditable ABE. In Section 5, we give the main constructions and security analysis. In Section 6, we provide a performance and experiment evaluation. Finally, Section 7 presents a brief conclusion and future work.

2. Related Work

Melissa [13] proposed a ciphertext strategy-based multi-agency authorization attribute-based encryption (MCP-ABE) scheme. The scheme has a central authority with the ability to decrypt each ciphertext, which reduces the security of decryption key storage. Lewko et al. [14] proposed a

multiagency authorization scheme that supports arbitrary access structures based on the groups in composite order, resulting in a low efficiency. In order to improve the efficiency of the scheme, Yannis et al. [15] proposed a CP-ABE scheme based on prime order groups and made it support large attribute domains. Then, Yannis et al. [11] proposed a multiagency authorization CP-ABE scheme based on prime order groups and also support large attribute domains. In this scheme, the authors used the linear secret-sharing scheme (LSSS) to improve expression ability. However, none of the abovementioned studies support traceability.

The traceability in ABE is divided into white-box traceable and black-box traceable [16]. In this field, Ning et al. [17] proposed a white-box traceable method that enables large attribute domains and high expressive capability. Their white-box traceable scheme is based on a single authorization center. To improve this, Li et al. [18] proposed a CP-ABE scheme with multiauthorization centers. However, this scheme only supports the access strategy of the AND gate, which limits in low expressive capability. Then, Zhou et al. [19] proposed a multiagency authorization CP-ABE scheme with white-box traceable that supports high expressive capability on medical cloud systems. However, their scheme does not support large attribute domains, and each authorization center has to maintain an identification table, which increases the storage overhead for tracking.

In the study of policy update, Ying et al. [20] proposed the first CP-ABE scheme that supports the modification of any form of fine-grained access control policy, and it is proved to be adaptive and secure under the standard model, but the system's communication overhead and storage overhead are high. After that, Liu et al. [21] proposed an ABE scheme that supports outsourcing decryption, attribute revocation, and policy update. This scheme is more flexible and practical in practice, but its privacy-protection capabilities are slightly lacking. Recently, Jing et al. [22] proposed a CP-ABE scheme that supports access policy update and rapid expansion of attributes but did not consider the application scenarios of multiauthorization agencies.

3. Background

3.1. Access Structure. We define U as a set of attributes, an access structure \mathbb{A} is a collection of nonempty subsets of U , that is, $\mathbb{A} \in 2^U / \{\emptyset\}$, and the collection contained in \mathbb{A} is called an authorization set. If the user has an authorized attribute set, the user can perform decryption, but not vice versa.

For all B and C , $B \in \mathbb{A}$, and $B \subseteq C$, if $C \in \mathbb{A}$, we say that the access structure \mathbb{A} is monotonous. We restrict to a monotone access structure in this paper.

3.2. Prime-Order Bilinear Groups. Let p be a big prime and \mathbb{G} and \mathbb{G}_T be cyclic groups with prime order p ; we say that $e: G \times G \rightarrow G_T$ is a computable bilinear map if it has the following properties:

- (1) Bilinear, i.e., $(e(P^a, Q^b) = e(P, Q)^{ab})$ for all $P, Q \in \mathbb{G}, a, b \in \mathbb{Z}_p$

- (2) Nondegeneracy, i.e., there exists $P, Q \in \mathbb{G}$ such that $e(P, Q) \neq 1$, namely, the map does not send all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity in \mathbb{G}_T
- (3) Computability, i.e., there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}$

3.3. Linear Secret-Sharing Schemes. Let U be the set of attributes, as shown in [23]; Π is a linear secret-sharing scheme (LSSS) on U if it has the following properties:

- (1) For each attribute form of a vector over \mathbb{Z}_p , there is a secret share $s \in \mathbb{Z}_p$.
- (2) The matrix for Π is called a share-generating matrix meaning a matrix M with l rows and n columns for each access structure \mathbb{A} on S . For $i = 1, \dots, l$, we define a function ρ labels row i of M with attribute $\rho(i)$. We consider the column vector $\vec{v} = (s, r_2, \dots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $r_2, \dots, r_n \in \mathbb{Z}_p$ are randomly chosen. Then, $M\vec{v} \in \mathbb{Z}_p^{l \times 1}$ is the vector of l shares of the secret s according to Π .

For the LSSS scheme, it enjoys the linear reconstruction property. More precisely, let Π be an LSSS for the access structure \mathbb{A} , $S^* \in \mathbb{A}$ be an authorized set, and let $I \subset \{1, 2, \dots, l\}$ be defined as $I = \{i \in [l] \wedge \rho(i) \in S^*\}$. Then, for constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, for any valid shares $\{\lambda_i = (M\vec{v})_{i \in I}\}_{i \in I}$ of a secret s according to Π , we have $\sum_{i \in I} \omega_i \lambda_i = s$.

3.4. Problem Assumption. Decisional q -parallel bilinear Diffie–Hellman exponent (q -PBDHE) assumption: the decisional q -parallel bilinear Diffie–Hellman exponent (decisional q -PBDHE) problem [11] is saying that, given the tuple (G, p, e, g, g^s) , it satisfies

$$\forall i \in \{1, \dots, 2q\}, j \in \{1, \dots, q\}, i \neq q + 1: \left(g^{a^i}, g^{b_j a^i} \right), \quad (1)$$

$$\forall i \in \{1, \dots, q\}: \left(g^{s/b_i} \right), \quad (2)$$

$$\forall i \in \{1, \dots, q + 1\}, j, j' \in \{1, \dots, q\}, j \neq j': \left(g^{(s a^i b_j / b_{j'})} \right), \quad (3)$$

if we can distinguish $Z = e(g, g)^{a^{q+1}s}$ from a random value in G_T .

Formally speaking, if $|\Pr[\mathcal{A}(\vec{y}, Z) = e(g, g)^{a^{q+1}s}] - \Pr[\mathcal{A}(\vec{y}, R) = 0]| \geq \epsilon$, we say that an algorithm \mathcal{A} has advantage ϵ in solving the abovementioned decisional q -PBDHE problem. Then, if all probabilistic polynomial time (PPT) algorithms have, at most, a negligible advantage in solving the decisional q -PBDHE problem, we say that the decisional q -PBDHE assumption holds.

LRSW assumption [12]: let G be the cyclic group of order p , g be a generator of G , and two random values $x, y \in \mathbb{Z}_p$ satisfy $X = g^x$ and $Y = g^y$. Let $\mathcal{O}_{X,Y}(\cdot)$ be the random oracle, which inputs $m \in \mathbb{Z}_p$ and outputs a triplet

$A = (a, a^y, a^{x+mx^y})$, where $a \in G$. If there is no probability polynomial time algorithm that can generate m, a, b, c satisfying $m \notin Q, Q \in \mathcal{O}_{X,Y}(\cdot), m \in \mathbb{Z}_p, m \neq 0, a \in G, b = a^x, c = a^{x+mx^y}$ with probability at the least ε , then the LRSW assumption in group G is said to be true.

4. Definition and Security Model

4.1. System Model. We show the framework of our system in Figure 1. There are six main entities, namely, cloud storage provider, attribute authorities (AAs), data owners, data users, system party, and trusted party. The system party will invoke the system setup algorithm and generate the public parameters (PP). The PP is then firstly distributed to the attribute authorities, data owners, data users, and the trusted party. Then, the AAs invoke the authority setup process to generate public keys (PKs) and send their public keys to the data owners, data users, and the trusted party. Also, if the data users possess valid credentials, AAs will assign the attributes to them according to their request. The data owner generates ciphertext (CT) for the message he wants to encrypt and uploads to the cloud storage provider. Once the data owner wishes to alter the access policy over the existing CT, he/she sends a policy update key to the cloud storage provider. Then, in the cloud storage, the ciphertext will be updated accordingly. Subsequently, if the users' attributes satisfy the access policy of the CT, they can use the components of secret key to generate their secret key SK and perform decryption operation. Finally, the trusted party invokes the tracing algorithm if there is dispute or suspicion and reports the suspected user's ID (gid) to the AAs.

4.2. Definition. Our proposed cryptosystem according to the abovedescription consists of the following eight algorithms:

Setup(λ) \rightarrow (PP): on input of a security parameter λ , the algorithm (run by the system) outputs the global PPs.

AuthoritySetup(aid, PP) \rightarrow (SK_{aid}, PK_{aid}): we assume each authority is recognized by an identifier aid. On input of the global PPs and aid, the algorithm outputs the public key PK_{aid} and the cloud secret key SK_{aid} .

KeyGen(gid, S, $\{SK_{aid}\}$, PP) $\rightarrow SK_{S,gid}$: on input of the user identity (gid), a set of user's attributes S, and the corresponding authority's secret keys SK_{aid} and PP, the algorithm outputs the private key $SK_{S,gid}$ for user matching his/her attribute set S.

Encrypt(msg, (M, ρ) , PK_{aid} , PP) \rightarrow (CT): this algorithm is run by a data owner who wants to share the data in the cloud. The algorithm inputs the message (msg) concerning an access policy (M, ρ) , a set of respective public keys PK_{aid} and PP, and outputs the ciphertext CT.

Decrypt(CT, $SK_{S,gid}$, PP) \rightarrow msg: this algorithm is run by a data user. On input of the global PPs, a ciphertext CT and a private key $SK_{S,gid}$ matching an

attribute set S and the algorithm outputs the message msg if decryption is possible.

PolicyUpdateKeyGen(PP, PK_{aid} , SharesInfo(msg), (M, ρ) , (M', ρ')) $\rightarrow UK_{msg}$: this algorithm is run by a data owner. On input of the global PPs, a set of public keys PK_{aid} , the encryption information SharesInfo(msg), the old access policy (M, ρ) , and new access policy (M', ρ') , the algorithm outputs the policy update key UK_{msg} .

CTUpte(CT, UK_{msg}) $\rightarrow CT'$: this algorithm is run by the cloud storage provider. On input of the ciphertext CT and updated key UK_{msg} , the algorithm outputs an updated ciphertext CT' .

Trace($SK_{S,gid}$, $\{PK_{aid}\}$, PP) \rightarrow gid or \perp : this algorithm is run by the trusted party. On input of the decryption key $SK_{S,gid}$ and the public keys $\{PK_{aid}\}$ for corresponding authorities and PPs, the algorithm outputs an authority gid.

4.3. Security Model. We focus on two types of adversaries as follows:

- (1) We consider the malicious data users as the *static adversary*. For static adversaries [11], we request that no unauthorized user can decrypt encrypted data stored in the cloud. In addition, we request that the collusion of a group of unauthorized malicious users is still unable to obtain unauthorized decryption privileges, which means our scheme needs to have collusion resistance.
- (2) We consider the "honest but curious" cloud provider as the *traceable adversary*. We assume that the traceable adversary [24] will follow the protocol's specification but will collect as much information as possible, i.e., secret/private keys. The traceable adversary is not allowed to obtain more secret information than it already has. In addition, it cannot identify "who has accessed the encrypted data" and "who has requested the decryption service." Also, it cannot link a valid decryption request to a previous decryption request.

Then, we have the following two security models.

4.3.1. Model 1: Security for Static Adversary. The security model for static adversary is based on the static security model [11]. To define the security of our scheme (satisfying the abovementioned requirements), we design the following security games:

Init. The adversary \mathcal{A} selects a set of corrupted authorization agencies, records it as $C_{aid} \subseteq U_{aid}$, and keeps it unchanged throughout the game. The normal authorized agencies are recorded as $N_{aid} \subseteq U_{aid}$ with $N_{aid} \subseteq U_{aid} = \emptyset$; \mathcal{A} knows the secret key of each corrupted organization $\{SK_{aid}\}_{aid \in C_{aid}}$.

Setup. The challenger \mathcal{C} runs the system Setup of the solution in this article and sends the global PP to the opponent.

Query. \mathcal{A} requests $\{(\text{gid}_j, S_j)\}_{j \in [m]}$ as the relevant private key, where $S_j \subseteq U$ is the attribute set of the user with identity gid_j . All users' identities are unique, and for arbitrary $i \in S$, there holds $T(i) \notin C_{\text{aid}}$. Then, the adversary sends two messages msg_0 and msg_1 with the same length and a set of challenges $\{(M_i, \rho_i), \dots, (M_p, \rho_p)\}$. For each challenge, the access policy must satisfy the nonauthorization set. Finally, the ciphertext policy is requested to update any two access policies of the query challenge message and among them.

Challenge. The challenger \mathcal{C} randomly selects and responds to the adversary according to the RW scheme [11], including a set of public keys of the normal authority, a satisfied user private key, and a set of verification ciphertexts used to challenge the adversary. We use the simulator to convert the adversary's query into a form that the challenger can recognize as a RW scheme and also convert the challenger's response to the adversary.

Guess. \mathcal{A} outputs a guess $b' = \{0, 1\}$ for b .

As can be seen in this game, the advantage of \mathcal{A} is defined as $\text{Adv} = |\Pr[b' = b] - 1/2|$.

According to [11], we have the following definition.

Definition 1. The T-DPU-MCP-ABE scheme is static secure if all PPT adversaries have at most a negligible advantage in the abovementioned game.

4.3.2. Model 2: Security for Traceable Adversary. The security game for traceable adversary is similar to the game of the static one except the *Setup*, *Query*, and *Forgery* (identical to *Guess*) as follows:

Setup. \mathcal{C} runs $\text{Setup}(\lambda)$ and $\text{AuthoritySetup}(\text{aid}, \text{PP})$ and sends the PP and the authority public key PK_{aid} to \mathcal{A} .

Query. \mathcal{A} requests $\{(\text{gid}_j, S_j)\}_{j \in [m]}$ as the relevant private key, where $S_j \subseteq U$ is the attribute set of the user with identity gid_j . Then, \mathcal{C} runs $\text{KeyGen}(\text{gid}_j, S_j, \text{SK}_{\text{aid}}, \text{PP})$ and sends $\{\text{SK}_{S_j, \text{gid}_j}\}_{j \in [m]}$ to \mathcal{A} .

Forgery. \mathcal{A} outputs a forgery secret key SK^* , if $\text{Trace}(\text{SK}_{S, \text{gid}}, \{\text{PK}_{\text{aid}}\}, \text{PP}) \notin \Delta$, and $\text{gid} \notin \{\text{gid}_1, \dots, \text{gid}_m\}$.

According to [24], we have the following definition.

Definition 2. The T-DPU-MCP-ABE scheme is traceable secure if all PPT adversaries have at most a negligible advantage $|\Pr[\text{Trace}(\text{SK}^*, \{\text{PK}_{\text{aid}}\}, \text{PP}) \notin \Delta, \text{gid}_1, \dots, \text{gid}_m]|$ in the abovementioned game.

5. Traceable and Dynamic Policy Updating Multiauthority Attribute-Based Encryption

Here, we present our attribute-based key encryption scheme. Our scheme is constructed on the bilinear group G with a large prime order p and utilizes the LSSS access strategy together with two random oracle hash functions H_1 and H_2 . We realize the traceability by adopting the CL (Camenisch–Lysyanskaya) signature scheme [25]. Our scheme has two domains, namely, the attribute domain U and the authority domain U_{aid} . There is a corresponding authorized authority aid releasing an effective attribute set to the users for each attribute.

Then, our scheme is specifically constructed as follows.

5.1. Our Construction

Setup(λ) \rightarrow (PP): this algorithm takes as input the security parameter λ and gets $D = (G, G_T, p, e)$, where p is the prime order and G_T, e is the bilinear mapping $e: G \times G \rightarrow G_T$. It sets the attribute universe be $\mathcal{U} = \mathbb{Z}_p$. It then chooses random $g \in G$ and three cryptographic hash functions H_1, H_2 , and T , where $H_1, H_2: \{0, 1\}^* \rightarrow \mathbb{G}$ are used to hash the identity and the attribute of a user into an element of G , respectively. Also, $T: \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ is used to hash the attribute i into the corresponding aid. Finally, this algorithm sets the global public parameters $\text{PP} = (G, G_T, p, e, g, H_1, H_2, T)$ as output.

AuthoritySetup(aid, PP) \rightarrow ($\text{SK}_{\text{aid}}, \text{PK}_{\text{aid}}$): the algorithm chooses three random $\alpha_{\text{aid}}, \beta_{\text{aid}}, \gamma_{\text{aid}} \in \mathbb{Z}_p$. Together with the inputs aid and PP, it then publishes the public key $\text{PK}_{\text{aid}} = \{e(g, g)^{\alpha_{\text{aid}}}, g^{\beta_{\text{aid}}}, g^{\gamma_{\text{aid}}}\}$ of the AU and sets the secret key as $\text{SK}_{\text{aid}} = \{\alpha_{\text{aid}}, \beta_{\text{aid}}, \gamma_{\text{aid}}\}$.

KeyGen(gid, S, $\{\text{SK}_{\text{aid}}\}$, PP) \rightarrow $\text{SK}_{S, \text{gid}}$: the algorithm chooses random $t \in \mathbb{Z}_p, u \in G, u \notin H_1(\text{gid})$ and computes

$$\begin{aligned} K_{1,i, \text{gid}} &= g^{\alpha_{\text{aid}}} \cdot H_1(\text{gid})^{\beta_{\text{aid}}} \cdot H_2(i)^t \cdot u^{\beta_{\text{aid}}(\text{gid} + \gamma_{\text{aid}})} K_{2,i, \text{gid}} \\ &= u^{\gamma_{\text{aid}}} K_{3,i, \text{gid}} = u K_{4,i, \text{gid}} = g^t K_{5, \text{gid}} = \text{gid}. \end{aligned} \quad (4)$$

It outputs the secret key $\text{SK}_{S, \text{gid}} = \{\{K_{1,i, \text{gid}}, K_{2,i, \text{gid}}, K_{3,i, \text{gid}}, K_{4,i, \text{gid}}\}_{i \in S}, K_{5, \text{gid}}\}$.

Encrypt(msg, (M, ρ) , PK_{aid} , PP) \rightarrow (CT): on input of the message (msg), the PPs and an access policy (M, ρ) (where M is an $l \times n$ matrix), the public key of the agency PK_{aid} , and the public parameters PP, the algorithm firstly chooses a random $s \in \mathbb{Z}_p$. Then, it chooses random $x_2, \dots, x_n \in \mathbb{Z}_p$, sets two vectors $\mathbf{v} = (s, x_1, x_2, \dots, x_n)$ and $\mathbf{v} = (0, v_2, \dots, v_n)$, and computes the vectors of shares of s and 0 as $\lambda_x = \mathbf{M}_x \mathbf{v}^T$ and $\omega_x = \mathbf{M}_x \mathbf{v}^T$, respectively (where T denotes the transpose of the matrix).

Finally, it chooses random $r_x \in \mathbb{Z}_p$ and computes

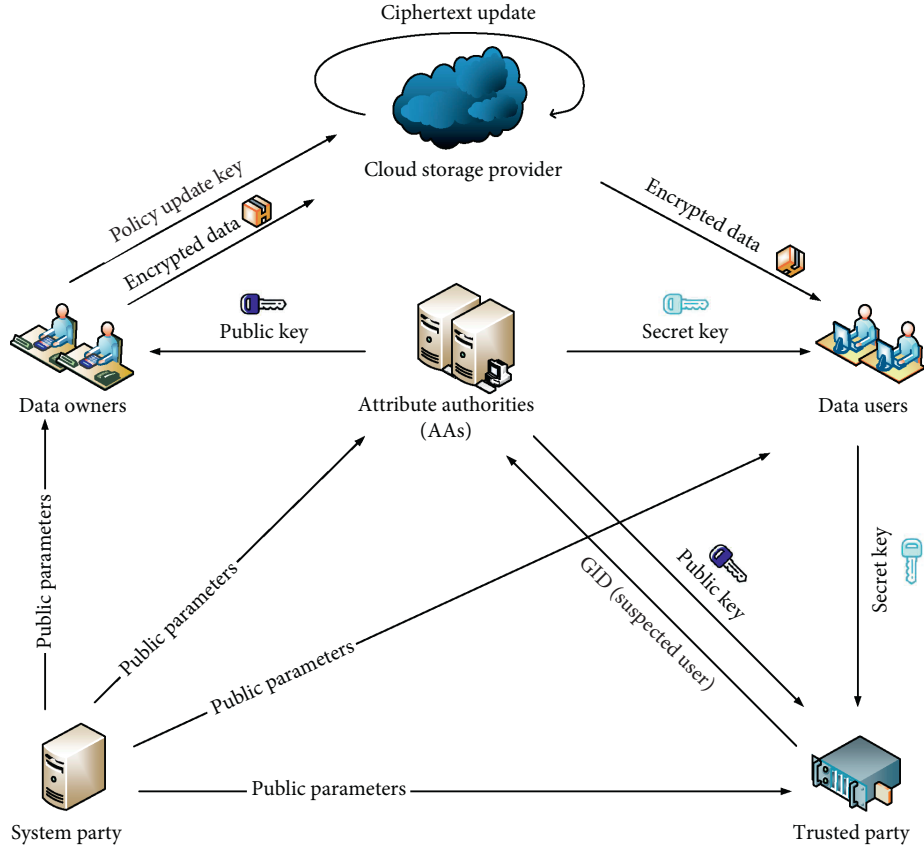


FIGURE 1: Framework of the system model.

$$\begin{aligned}
 C_0 &= \text{msg} \cdot e(g, g)^s, \\
 C_{1,x} &= e(g, g)^{\lambda_x + \alpha \delta(x) r_x}, \\
 C_{2,x} &= g^{\omega_x}, \\
 C_{3,x} &= g^{\beta \delta(x) r_x}, \\
 C_{4,x} &= H_2(\rho(x))^{r_x}, \\
 C_{5,x} &= g^{-r_x}.
 \end{aligned} \tag{5}$$

The ciphertext CT is set as $CT = \{C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}, C_{5,x}\}_{x \in \{1, \dots, l\}}\}$.

Decrypt(CT, $SK_{S, \text{gid}}$, PP) \rightarrow msg: on input of $CT = \{C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}, C_{5,x}\}_{x \in \{1, \dots, l\}}\}$, S, $SK_{S, \text{gid}}$, and PP, the algorithm sets the identification set as $I \subseteq \{1, \dots, l\}$. For all $x \in I$ and $\{x: \rho(x) \in S\}$, the algorithm computes

$$\begin{aligned}
 D_x &= C_{1,x} \cdot e(H_1(K_{5, \text{gid}}), C_{2,x} \cdot C_{3,x}) \\
 &\quad \cdot e(K_{2, \rho(x), \text{gid}} \cdot K_{3, \rho(x), \text{gid}}^{K_{5, \text{gid}}}, C_{3,x}) \cdot e(C_{4,x}, K_{4, \rho(x), \text{gid}}) \\
 &\quad \cdot e(K_{1, \rho(x), \text{gid}}, C_{5,x}),
 \end{aligned} \tag{6}$$

where $\{c_x\}_{x \in I}$ and $\sum_{x \in I} c_x \mathbf{M}_x = (1, 0, \dots, 0)$.

Finally, the message is recovered by computing

$$\text{msg} = \frac{C_0}{\prod_{x \in I} D_x}. \tag{7}$$

PolicyUpteKeyGen(PP, PK_{aid} , Shares, (M, ρ) , (M', ρ')) \rightarrow UK_{msg} : M is a generator matrix of $1, \dots, n$, and Shares represents the information of the two random vectors \mathbf{v} and v contained in the encryption algorithm. We define the function $\delta(i) = T(\rho(i))_{i \in [l]}$ and $\delta'(j) = T'(\rho(j))_{j \in [l]}$.

First, the new access strategy and the old access strategy are used as input through the strategy comparison method in the literature [26] to generate three subset record rows indexes $I_{1,M}, I_{2,M}, I_{3,M}$. Then, it picks two random vectors $\mathbf{v}' = (s, v'_2, \dots, v'_n)$ and $v' = (0, v'_2, \dots, v'_n)$ and then calculates $\lambda'_j = \mathbf{M}'_j \mathbf{v}'^T$ and $\omega'_j = \mathbf{M}'_j v'^T$ with $j \in \{1, \dots, l'\}$.

When the row index satisfies $(j, i) \in I_{1,M'}$ (marked as module 1), the algorithm generates the update key as

$$\{UK_{j,i,\text{msg}}\}_1 = \{UK_{1,j,i,\text{msg}} = g^{\lambda'_j - \lambda_i}, UK_{2,j,i,\text{msg}} = g^{\omega'_j - \omega_i}\}. \tag{8}$$

When the row index satisfies $(j, i) \in I_{2,M'}$ (marked as module 2), the algorithm randomly picks $a_j \in \mathbb{Z}_p$ and calculates the update key as

$$\{\text{UK}_{j,i,\text{msg}}\}_2 = \left\{ a_j, \text{UK}_{1,j,i,\text{msg}} = g_i^{\lambda_j - a_j \lambda}, \text{UK}_{2,j,i,\text{msg}} = g^{\omega_j - a_j \omega_i} \right\}. \quad (9)$$

When the row index satisfies $(j, i) \in I_{3,M'}$ (marked as Module 3), the algorithm randomly picks $r'_j \in \mathbb{Z}_p$ and generates the update key as

$$\{\text{UK}_{j,i,\text{msg}}\}_3 = \left\{ \begin{array}{l} \text{UK}_{1,j,i,\text{msg}} = g^{\lambda_j + \alpha_{\delta'(j)} r'_j}, \\ \text{UK}_{2,j,i,\text{msg}} = g^{\omega_j}, \\ \text{UK}_{3,j,i,\text{msg}} = g^{\beta_{\delta'(j)} r'_j}, \\ \text{UK}_{4,j,i,\text{msg}} = H_2(\rho'(j))^{r'_j}, \\ \text{UK}_{5,j,i,\text{msg}} = g^{r'_j} \end{array} \right\}. \quad (10)$$

Finally, the data owner sends the updated key UK_{msg} to the cloud storage service provider with $\text{UK}_{\text{msg}} = \{\{\text{UK}_{j,i,\text{msg}}\}_1, \{\text{UK}_{j,i,\text{msg}}\}_2, \{\text{UK}_{j,i,\text{msg}}\}_3\}$.

CTUpdate(CT, UK_{msg}) \rightarrow CT': after the cloud storage service provider receives the update key, it updates the ciphertext CT to CT'. By doing so, the cloud storage service provider cannot obtain relevant information during the re-encryption process of the ciphertext. The specific updates are as follows:

When the row index belongs to module 1, the update parameter is

$$\begin{aligned} C'_{1,j} &= C_{1,i} \cdot e(g, \text{UK}_{1,j,i,\text{msg}}) = e(g, g)^{\lambda_j + \alpha_{\delta'(j)} r'_j}, \\ C'_{2,j} &= C_{2,i} \cdot \text{UK}_{2,j,i,\text{msg}} = g^{\omega_j}, \\ C'_{3,j} &= C_{3,i} = g^{\beta_{\delta'(j)} r'_j}, \\ C'_{4,j} &= C_{4,i} = H_2(\rho'(j))^{r'_j}, \\ C'_{5,j} &= C_{5,i} = g^{-r'_j}, \\ r'_j &= r_i, \delta'(j) = \delta(i) = H_2(\rho'(j)) = H_2(\rho(i)). \end{aligned} \quad (11)$$

When the row index belongs to module 2, the update parameter is

$$\begin{aligned} C'_{1,j} &= (C_{1,i})^{a_j} \cdot e(g, \text{UK}_{1,j,i,\text{msg}}) = e(g, g)^{\lambda_j + \alpha_{\delta'(j)} r'_j}, \\ C'_{2,j} &= (C_{2,i})^{a_j} \cdot \text{UK}_{2,j,i,\text{msg}} = g^{\omega_j}, \\ C'_{3,j} &= (C_{3,i})^{a_j} = g^{\beta_{\delta'(j)} r'_j}, \\ C'_{4,j} &= (C_{4,i})^{a_j} = H_2(\rho(i))^{r_i a_j} = H_2(\rho'(j))^{r'_j}, \\ C'_{5,j} &= (C_{5,i})^{a_j} = g^{-r_i a_j} = g^{-r'_j}, \\ r'_j &= r_i a_j, \delta'(j) = \delta(i). \end{aligned} \quad (12)$$

When the row index belongs to module 3, the update parameter is

$$\begin{aligned} C'_{1,j} &= e(g, \text{UK}_{1,j,i,\text{msg}}) = e(g, g)^{\lambda_j + \alpha_{\delta'(j)} r'_j}, \\ C'_{2,j} &= \text{UK}_{2,j,i,\text{msg}} = g^{\omega_j}, \\ C'_{3,j} &= \text{UK}_{3,j,i,\text{msg}} = g^{\beta_{\delta'(j)} r'_j}, \\ C'_{4,j} &= \text{UK}_{4,j,i,\text{msg}} = H_2(\rho'(j))^{r'_j}, \\ C'_{5,j} &= \text{UK}_{5,j,i,\text{msg}} = g^{-r'_j}, \\ r'_j &= r_i a_j, \delta'(j) = \delta(i). \end{aligned} \quad (13)$$

Finally, the updated ciphertext CT' is $\text{CT}' = \{C_0, \{C'_{1,j}, C'_{2,j}, C'_{3,j}, C'_{4,j}, C'_{5,j}\}_{j \in \{1, \dots, l'\}}\}$.

Trace($\text{SK}_{S,\text{gid}}, \{\text{PK}_{\text{aid}}\}, \text{PP}$) \rightarrow gid or \perp : the algorithm inputs the decryption key $\text{SK}_{S,\text{gid}}$ and the public key $\{\text{PK}_{\text{aid}}\}$ associated with the global public parameter PP. If the decryption key $\text{SK}_{S,\text{gid}}$ is not in the form $\text{SK}_{S,\text{gid}} = \{\{K_{1,i,\text{gid}}, K_{2,i,\text{gid}}, K_{3,i,\text{gid}}, K_{4,i,\text{gid}}\}_{i \in S}, K_{5,\text{gid}}\}$ or cannot pass the key integrity check, the algorithm will output a special symbol to indicate that there is no need to trace $\text{SK}_{S,\text{gid}}$. The key integrity check of this scheme is as follows:

$$\begin{aligned} K_{1,i,\text{gid}}, K_{2,i,\text{gid}}, K_{3,i,\text{gid}}, K_{4,i,\text{gid}} &\in G, K_{5,\text{gid}} \in \mathbb{Z}_p^*, \\ e(K_{2,i,\text{gid}}, g) &= e(K_{3,i,\text{gid}}, g^{\gamma_{\text{aid}}}), \\ e(K_{1,i,\text{gid}}, g) &= e(g, g)^{\alpha_{\text{aid}}} \cdot e(H(K_{5,\text{gid}}), g^{\beta_{\text{aid}}}) \cdot e(F(i), K_{4,i,\text{gid}}) \cdot e(K_{2,\rho(x),\text{gid}} \cdot K_{3,\rho(x),\text{gid}}^{K_{5,\text{gid}}}, g^{\beta_{\text{aid}}}). \end{aligned} \quad (14)$$

If there is an attribute $i \in S$ that satisfies equations (14), it is considered that the key $\text{SK}_{S,\text{gid}}$ passes the integrity check, and the identity gid is output as the trace identity.

5.2. Correctness. The correctness of our scheme can be obtained from the following equations. It is known that

$$\begin{aligned} D_x &= C_{1,x} \cdot e(H_1(K_{5,\text{gid}}), C_{2,x} \cdot C_{3,x}) \\ &\cdot e(K_{2,\rho(x),\text{gid}} \cdot K_{3,\rho(x),\text{gid}}^{K_{5,\text{gid}}}, C_{3,x}) \cdot e(C_{4,x}, K_{4,\rho(x),\text{gid}}) \\ &\cdot e(K_{1,\rho(x),\text{gid}}, C_{5,x}). \end{aligned} \quad (15)$$

According to the corresponding values of CT and $\text{SK}_{S,\text{gid}}$, we can obtain

$$\begin{aligned}
D_x &= e(g, g)^{\lambda_x + \alpha_{\delta(x)} r_x} \cdot e(H_1(\text{gid}), g^{\omega_x} \cdot g^{\beta_{\delta(x)} r_x}) \\
&\quad \cdot e(u^{\gamma_{\delta(x)}} \cdot u^{\text{gid}}, g^{\beta_{\delta(x)} r_x}) \\
&\quad \cdot e(H_2(\rho(x))^{r_x}, g^t) \cdot e(g^{\alpha_{\delta(x)}} \cdot H_1(\text{gid})^{\beta_{\delta(x)}} \\
&\quad \cdot H_2(i)^t \cdot u^{\beta_{\delta(x)} (\text{gid} + \gamma_{\delta(x)})}, g^{-r_x}), \\
&= e(g, g)^{\lambda_x + \alpha_{\delta(x)} r_x} \cdot e(H_1(\text{gid}), g)^{\omega_x + \beta_{\delta(x)} r_x} \\
&\quad \cdot e(u, g)^{(\gamma_{\delta(x)} + \text{gid}) \beta_{\delta(x)} r_x} \cdot e(H_2(\rho(x)), g)^{r_x t} \\
&\quad \cdot e(g, g)^{-\alpha_{\delta(x)} r_x} \cdot e(H_1(\text{gid}), g)^{-\beta_{\delta(x)} r_x} \\
&\quad \cdot e(H_2(\rho(x)), g)^{-r_x t} \cdot e(u, g)^{-(\gamma_{\delta(x)} + \text{gid}) \beta_{\delta(x)} r_x}, \\
&= e(g, g)^{\lambda_x} \cdot e(H_1(\text{gid}), g)^{\omega_x}.
\end{aligned} \tag{16}$$

Then, for $\{c_x\}_{x \in I}$ and $\sum_{x \in I} c_x \mathbf{M}_x = (1, 0, \dots, 0)$, we have

$$\begin{aligned}
\sum_{x \in I} \lambda_x c_x &= \sum_{x \in I} \mathbf{M}_x \mathbf{v}^T c_x = (1, 0, \dots, 0) \cdot \mathbf{v}^T = s, \\
\sum_{x \in I} \omega_x c_x &= \sum_{x \in I} \mathbf{M}_x \mathbf{v}^T c_x = (1, 0, \dots, 0) \cdot \mathbf{v}^T = 0.
\end{aligned} \tag{17}$$

Hence, we have

$$\begin{aligned}
\prod_{x \in I} D_x^{c_x} &= \prod_{x \in I} (e(g, g)^{\lambda_x} \cdot e(H_1(\text{gid}), g)^{\omega_x})^{c_x}, \\
&= e(g, g)^{\sum_{x \in I} \lambda_x c_x} \cdot e(g, g)^{\sum_{x \in I} \omega_x c_x}, \\
&= e(g, g)^s.
\end{aligned} \tag{18}$$

This proves that the message can be correctly restored to

$$\text{msg} = \frac{C_0}{\prod_{x \in I} D_x^{c_x}}. \tag{19}$$

5.3. Security Analysis

Theorem 1. *Assume the CP-ABE system in [11] is statically secure; then, the T-DPU-MCP-ABE system is static secure with respect to Definition 1.*

Proof. For simplicity, we use Σ_{RW} , Σ_{tdpum} to denote the CP-ABE system in [11] and our T-DPU-MCP-ABE system, respectively. We suppose there exists a static polynomial time attacker \mathcal{A} that breaks Σ_{RW} with a nonnegligible advantage in selectively with a challenge LSSS access policy (M^*, ρ^*) , where M^* is an $l \times n$ matrix. We will build a PPT algorithm \mathcal{B} that breaks Σ_{tdpum} with a nonnegligible advantage.

Init: \mathcal{B} gets a challenge LSSS access policy (M^*, ρ^*) from \mathcal{A} and transmits the received (M^*, ρ^*) to the Σ_e challenger \mathcal{C} .

Setup: \mathcal{C} generates the common parameter $\text{PP} = (G, G_T, p, e, g, H_1, H_2, T)$ and sends it to \mathcal{A} .

Query: \mathcal{B} initializes an integer counter $j = 0$ and an empty table T . Then, \mathcal{A} makes the following queries:

Receiving \mathcal{A} 's decryption key query with an attribute does not satisfy (M^*, ρ^*) , \mathcal{B} sets the attribute as S_j and $j = j + 1$, then sends them to the Σ_{tdpum} challenger, and obtains a secret key $\text{SK}'_{S, \text{gid}} = (\{K'_{1, \tau, \text{gid}}, K'_{2, \tau, \text{gid}}, K'_{3, \tau, \text{gid}}, K'_{4, \tau, \text{gid}}\}_{\tau \in [S]}, K'_{5, \text{gid}})$. \mathcal{A} chooses a corrupted AA $C_{\text{aid}} \in U_{\text{aid}}$ and generates the corresponding public key $\text{PK}'_{\text{aid}} = (e(g, g)^{\text{aid}}, g^{\beta_{\text{aid}}})$ in S_{RW} . Also, for each $\text{aid} \in C_{\text{aid}}$, \mathcal{A} randomly chooses $\gamma_{\text{aid}} \in \mathbb{Z}_p^*$ and generates the system public key $\text{PK}_{\text{aid}} = (e(g, g)^{\text{aid}}, g^{\beta_{\text{aid}}}, g^{\gamma_{\text{aid}}})$. Then, \mathcal{A} responses for the normal AA N_{aid} , the corrupted AA C_{aid} by interacting with \mathcal{B} as follows. \mathcal{A} requires $\{(\text{gid}_j, S_j)\}_{j \in [m]}$, where $S_j \subset U$ is the corresponding attribute set of user gid_j . All users' gid_j are unique and for arbitrary $i \in S$, we have $T(i) \notin C_{\text{aid}}$. Then, \mathcal{A} fixes a coin $b \in \{0, 1\}$, which is used to generate message msg_0 or msg_1 with the same length. \mathcal{A} chooses a set of challenge $\{(M_1, \rho), \dots, (M_q, \rho_q)\}$. Finally, \mathcal{A} sends all the chosen parameters to \mathcal{B} .

Challenge: \mathcal{A} chooses two same length messages (m_0, m_1) and sends to \mathcal{B} . Then, \mathcal{B} submits (m_0, m_1) to the Σ_{tdpum} challenger, obtains a challenge common public key $\text{PK}'_{\text{aid}} = (e(g, g)^{\text{aid}}, g^{\beta_{\text{aid}}})$, and generates a ciphertext $\text{ct}^* = (C_0^* \{C_{1,x}^*, C_{2,x}^*, C_{3,x}^*, C_{4,x}^*, C_{5,x}^*\}_{x \in \{1, \dots, l\}})$. \mathcal{B} chooses a random bit $b_{\mathcal{B}} \in \{0, 1\}$, computes $\text{key}_{b_{\mathcal{B}}} = C^* / m_{b_{\mathcal{B}}}$, and sends the new ciphertext $\text{ct}^* = (C_0^* \{C_{1,x}^*, C_{2,x}^*, C_{3,x}^*, C_{4,x}^*, C_{5,x}^*\}_{x \in \{1, \dots, l\}})$ to \mathcal{A} .

Guess: finally, after receiving the abovementioned responses, \mathcal{A} outputs a guess $b_{\mathcal{A}} \in \{0, 1\}$. If $b_{\mathcal{A}} = 1$, it means that \mathcal{A} guesses that $\text{key}_{b_{\mathcal{B}}}$ is a random key, and \mathcal{B} outputs $1 - b_{\mathcal{B}}$. If $b_{\mathcal{A}} = 0$, meaning that \mathcal{A} guesses that $\text{key}_{b_{\mathcal{B}}}$ is the key from ct_{new}^* , \mathcal{B} outputs $b_{\mathcal{B}}$.

Since the real system is the same as the distributions of the challenge ciphertext, if \mathcal{A} breaks the security of S_{RW} with a nonnegligible advantage, then the simulator \mathcal{B} can selectively break S_{tdpum} with the same advantage. \square

Theorem 2. *Assume the CL signature scheme in [25] is against existing forgery, and the T-DPU-MCP-ABE system in Section 5.1 is traceable secure with respect to Definition 2.*

Proof. The security proof of the T-DPU-MCP-ABE system with respect to Definition 2 (i.e., for traceable adversary) is identical to the abovementioned proof except that the adversary runs the **Forgery** phase instead of the **Guess** phase. Here, we suppose there exists a PPT attacker \mathcal{A} that selectively breaks the CL scheme with a nonnegligible advantage. We can build a PPT simulator algorithm \mathcal{B} that selectively breaks Σ_{tdpum} with a nonnegligible advantage. It is proved that the CL scheme is secure against existential forgery under adaptive chosen message attack with LRSW assumption.

Setup: the CL scheme challenger \mathcal{C} delivers each authority's public keys $\{G, G_T, p, g, g^{\beta_{\text{aid}}}, g^{\gamma_{\text{aid}}}\}$ to the simulator algorithm \mathcal{B} . \mathcal{B} chooses random values

$\alpha_{\text{aid}} \in \mathbb{Z}_p^*$ for each authority, runs Setup(λ) and AuthoritySetup(aid, PP) to generate the public key $\text{PK}_{\text{aid}} = \{e(g, g)^{\alpha_{\text{aid}}}, g^{\beta_{\text{aid}}}, g^{\gamma_{\text{aid}}}\}$, and sends the public parameter PP and the authority public key PK_{aid} to \mathcal{A} . The two hash functions H_1 and H_2 of our scheme are managed by simulator \mathcal{B} .

Query. \mathcal{A} requests $\{(\text{gid}_j, S_j)\}_{j \in [m]}$ as the relevant private key, where $S_j \subseteq U$ means the attribute set of the user gid_j . Before \mathcal{A} forges the key, to maintain hash functions H_1 and H_2 , \mathcal{B} will set two empty tables, T_1 and T_2 , respectively, and update them according to the query of \mathcal{A} . When the gid queried by \mathcal{A} does not exist in the table of T_1 and T_2 , \mathcal{B} will select a random element $t_{\text{gid}} \in \mathbb{Z}_p^*$ and a random element $t_i \in \mathbb{Z}_p^*$ and then record $(t_{\text{gid}}, g^{t_{\text{gid}}})$ and (t_i, g^{t_i}) with T_1 and T_2 , respectively. At the same time, simulator \mathcal{B} will return the hash value of H_1 or H_2 according to opponent the query of \mathcal{A} . For each $i \in S_j$, if the attribute authority $\text{aid} = T(i)$, then \mathcal{B} will submit $(\text{gid}_j, \text{aid})$ to Challenger C according to the query of \mathcal{A} so as to obtain the signature $(u, u^{\gamma_{\text{aid}}}, u^{\beta_{\text{aid}} \cdot ((\gamma_{\text{aid}}/\text{gid}_j)^{+1})})$ in the CL scheme. Then, \mathcal{B} takes the random value $t \in \mathbb{Z}_p^*$ and runs KeyGen($\text{gid}_j, S_j, \text{SK}_{\text{aid}}, \text{PP}$) as well as sends $\left\{ \text{SK}_{S_j, \text{gid}_j} \right\}_{j \in [m]}$ to \mathcal{A} . In this step, \mathcal{B} should compute the following:

$$\begin{aligned} K_{1,i,\text{gid}} &= g^{\alpha_{\text{aid}}} \cdot H_1(\text{gid}_j)^{\beta_{\text{aid}}} \cdot H_2(i)^t \cdot u^{\beta_{\text{aid}}(\text{gid}+\gamma_{\text{aid}})}, \\ K_{2,i,\text{gid}} &= u^{\gamma_{\text{aid}}}, \\ K_{3,i,\text{gid}} &= u, \\ K_{4,i,\text{gid}} &= g^t, \\ K_{5,\text{gid}} &= \text{gid}. \end{aligned} \quad (20)$$

Then, the final calculation is $\text{SK}_{S_j, \text{gid}_j}$ as $\left\{ \left\{ K_{1,i,\text{gid}_j}, K_{2,i,\text{gid}_j}, K_{3,i,\text{gid}_j}, K_{4,i,\text{gid}_j} \right\}_{i \in S_j}, K_{5,\text{gid}_j} \right\}$.

Forgery. in this step, \mathcal{A} already queries from simulator \mathcal{B} the value of $H_1(\text{gid})$ and $H_2(i)$ and obtains $H_1(\text{gid})$ as $g^{t_{K_5,\text{gid}}}$ and $H_2(i)$ as g^{t_i} . \mathcal{A} assumes the unknown $K_{3,i,\text{gid}} = g^{t_3}$ and $K_{4,i,\text{gid}} = g^{t_4}$. Through formula (14) in Section 5.1, we could get that $K_{2,i,\text{gid}} = (K_{3,i,\text{gid}}) = g^{t_3 \gamma_{\text{aid}}}$. Also through formula (14) in Section 5.1, we could get that $K_{1,i,\text{gid}} = g^{\alpha_{\text{aid}} + t_{K_5,\text{gid}} \beta_{\text{aid}}} \cdot (K_{4,i,\text{gid}})^{t_i} \cdot (K_{3,i,\text{gid}})^{\beta_{\text{aid}}(K_5,\text{gid} + \gamma_{\text{aid}})}$.

Then, \mathcal{B} calculates a legal signature σ according to the CL scheme, and the calculation process is as follows:

$$\begin{aligned} \sigma_1 &= \frac{K_{1,i,\text{gid}}}{g^{\alpha_{\text{aid}} + t_{K_5,\text{gid}} \beta_{\text{aid}}} \cdot (K_{4,i,\text{gid}})^{t_i}}, \\ &= (K_{3,i,\text{gid}})^{\beta_{\text{aid}}(K_5,\text{gid} + \gamma_{\text{aid}})}. \end{aligned} \quad (21)$$

Then, \mathcal{A} picks a gid as a message and gives $\sigma = (K_{3,i,\text{gid}}, K_{2,i,\text{gid}}, (\sigma_1 / K_{3,i,\text{gid}}^{\text{gid}}))$ as the signature of the message gid according to the CL scheme.

Finally, \mathcal{A} outputs a forgery secret key SK^* , if $\text{Trace}(\text{SK}_{S_j, \text{gid}}, \{\text{PK}_{\text{aid}}\}, \text{PP}) \notin \Delta$ and $\text{gid} \notin \{\text{gid}_1, \dots, \text{gid}_m\}$. As $\text{gid} \notin \{\text{gid}_1, \dots, \text{gid}_m\}$, we know that the signature of message gid is not invoked by \mathcal{B} yet. Thus, the simulator \mathcal{B} breaks the CL scheme with the same advantage.

Since in the abovementioned game the whole system has the decryption keys, the distributions of the public parameters, and challenge ciphertext, if \mathcal{A} breaks the security of the CL scheme, then the simulator \mathcal{B} can selectively break S_{tdpvm} with the same advantage. Hence, if the LRSW assumption holds true, the proposed cryptosystem is against forgery, meaning that our scheme is traceable secure for the adversary. \square

5.4. Proof of Collusion Prevention. In our scheme, we use the unique gid and construct the hash function value corresponding to gid to resist collusion attack, which has been proved to be feasible by Allison and Waters [14]. In the process of decryption, the data user needs to calculate $D_x = e(g, g)^{\lambda_x} \cdot e(H_1(\text{gid}), g)^{\omega_x}$. For a single user with the access policy satisfaction attribute set, since ω_x are the shares of secret value 0, $e(H_1(\text{gid}), g)^{\omega_x}$ can be eliminated, where $e(H_1(\text{gid}), g)^{\omega_x} = 1$. In case of collusion attack, two or more users will have different gid ; thus, the value of $H_1(\text{gid})$ will also be different; $e(H_1(\text{gid}), g)^{\omega_x}$ with a secret value of 0 cannot be constructed, and thus, it cannot be eliminated. Therefore, two or more users cannot share their attribute key values to generate collusion attacks, which means this scheme is resistant to collusion attack.

6. Performance Evaluations

6.1. Theoretical Analysis. We first theoretically make a comparison of our scheme with others. The comparison of feature and performance of our work and related works is given in Tables 1 and 2.

It can be seen from Table 1 that the YB scheme [11] does not realize the traceability, nor does it have the function of dynamic access policy update; although the JZXL scheme [27] has both traceability and large attribute domains, it is constructed based on composite orders and is a single authorization which will become a bottleneck. Since the QLZH scheme [28] and the YLLT scheme [29] are based on tree access structure, they do not have the functions of large attribute domain, dynamic access strategy update, and traceability. The YLMH scheme [30] can realize the dynamic access strategy update but does not support traceability; while the ZLML scheme [31] does not have the function of dynamic access policy update. Compared with the abovementioned related schemes, our scheme not only supports traceability, large attribute domain, and dynamic access policy update at the same time under multiple authorization agencies but also is based on the prime order bilinear group structure, which is more efficient.

Let G and G_T be the size of elements in G and an exponentiation in G_T , respectively. Let e be a pairing and \exp be the maximum amounts of time to compute an exponentiation in G . Let A be the number of ciphertext attributes, $|S|$ be the size of the attribute set of a private key, and l be the

TABLE 1: Characteristics comparison of ABE schemes.

	YB [11]	JZXL [27]	QLZH [28]	YLLT [29]	YLMH [30]	ZLML [31]	Ours
Order groups	Prime	Composite	Prime	Prime	Prime	Prime	Prime
Large universe	√	√	×	×	×	√	√
Policy updating	×	×	×	×	√	×	√
Traceable	×	√	×	×	×	√	√
Access structure	LSSS	LSSS	TREE	TREE	LSSS	LSSS	LSSS
Multiauthority	√	×	√	√	√	√	√

TABLE 2: Performance comparison of multiauthority ABE schemes (<https://github.com/monzxcv/ABE>).

	YB [11]	SPB [32]	YLMH [30]	ZLML [31]	Ours
AA's public key	$G + G_T$	$3G + G_T$	$(n_i + 1)G + G_T$	$3G + G_T$	$2G + G_T$
User's private key	$2 S G$	$4 S G + G_T$	$2 S G + G_T$	$4 S G$	$4 S G + G_T$
Ciphertext	$3lG + (l + 1)G_T$	$4lG + (l + 1)G_T$	$(2l + 1)G + G_T$	$5lG + (l + 1)G_T$	$4lG + (l + 1)G_T$
Encryption cost	$3l \exp + (l + 1)e$	$4l \exp + (l + 1)e$	$(2l + 1)\exp + e$	$5l \exp + (l + 1)e$	$4l \exp + (l + 1)e$
Decryption cost	$3 I $	$4 I $	$2 S + 2 I $	$3 I $	$4 I $
Security assumption	q-type	q-type	q-PBDHE	q-type	q-type

output size of a function. Let I be the number of rows of the matrix when decrypting.

In Table 2, we show the communication cost and the computing cost comparison. Compared with other solutions, our scheme is relatively better in the process of adding multiple functions. On the one hand, for the communication cost, we can draw the following conclusions: Firstly, our scheme has the advantages in the length of the private key that our scheme supports big attribute universe. More precisely, the public key of our scheme does not increase linearly with the size of the attribute domain in an attribute authority, while that of the YLMH scheme will, and the storage occupied by our public key is smaller than that of the SPB scheme [32] and the ZLML scheme. Secondly, although the user's private keys in the YB scheme and the YLMH scheme are relatively small, none of these schemes support traceability. In order to enhance the security of the system, the scheme in this paper supports the traceability function, and the user's private key does not increase too much. Furthermore, compared to the YLMH scheme and the ZLML scheme, the length of the ciphertext in our scheme is optimized, which is only linearly related to the number of rows from the generator matrix. On the other hand, for the calculation cost, our scheme supports an access strategy update algorithm, while the YB scheme and the YLMH scheme do not support this function. Finally, for the decryption cost, our scheme is much smaller than that of the YLMH scheme. The decryption cost in our scheme is only related to the number of attribute organizations where the attributes belong. Although the decryption cost in our scheme is slightly higher than that of the YB scheme and the ZLML scheme, the YB scheme does not support traceability and the ZLML scheme does not support access policy update.

6.2. Experimental Analysis. In this section, we conduct a simulation experiment to evaluate the comparison of our scheme and the baseline algorithms (the simulation code is available in (<https://github.com/monzxcv/ABE>)). We select the scheme in [11] (YB scheme) and the scheme in [30] (YLMH scheme) as our baseline algorithms and run the

experiments in five aspects: system initialization, key generation, data encryption, user decryption, and access strategy re-encryption. All the experiments are run on a 64-bit operating system of the Ubuntu 14.04 platform with a core 1.8 GHz processor and 4 GB RAM. We used Charm version 0.50 and Python version 3.7 as our program languages. We first convert the YB scheme, YLMH scheme, and our scheme into asymmetric bilinear mapping and use the famous supersingular symmetric elliptic curve group ("SS512"). Then, in the process of encryption and decryption, the YB scheme, YLMH scheme, and our scheme are only related to the number of access policy attributes. Therefore, in this experiment, we change the number of user attributes and calculate the time of system initialization and user key generation under the same condition to get our first comparison. In addition, we change the access policy and calculate the time of the user encryption and decryption to get another comparison. Finally, the time consumed for updating ciphertext under the same condition is calculated. The experimental attributes are constructed with $A_N, N \in [1, \dots, 50]$. The strategy set is selected $(A_1 \wedge A_2 \wedge \dots \wedge A_N)$. We increase the number of attributes from 5 to 50, and there are ten different access strategies. In order to ensure the accuracy of the conclusion, every experiment is run 15 times.

The system initialization cost and the average time cost of user private key generation are shown in Figures 2 and 3 when the number of attributes varies from 5 to 50. We fix the number of AAs in 8, and we also fix the number of attributes in the access policy in 8. Since both our scheme and YB scheme support large attribute domains, the system initialization process has nothing to do with the number of attributes, as is verified in Figure 2. It can be seen that as the number of attributes increases, the cost of the YLMH scheme increases, and the cost of our scheme still keeps a constant value, so the larger the number of attributes, the more the advantage in our scheme. It can be seen from Figure 3 that the cost of the user private key generation time in all the three schemes increases linearly with the increase of

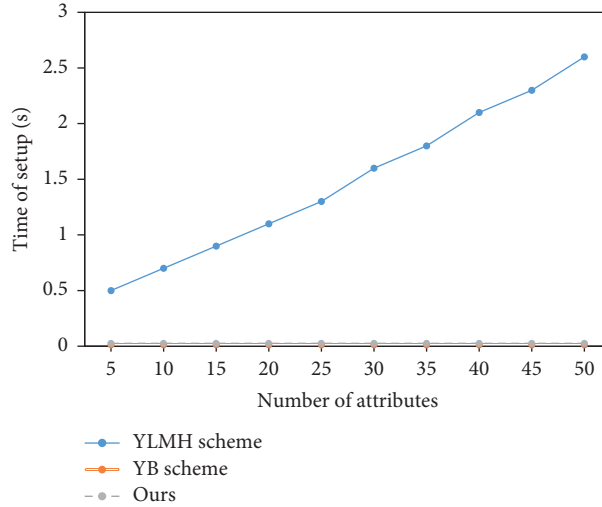


FIGURE 2: Comparison of the system setup process.

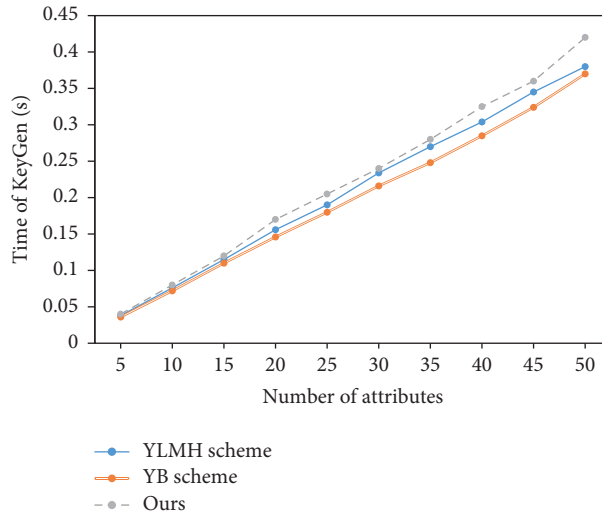


FIGURE 3: Comparison of the KeyGen process.

attributes. This is because each attribute in the user's private key must be calculated accordingly. Finally, the generation time cost is not much different from that of the YB scheme and the YLMH scheme.

Figure 4 shows the average time cost of the encryption and decryption process when the number of attributes used in the access policy varies from 5 to 50. We fix the number of AAs in 8, and the number of attributes for each user is also fixed in 8. It can be seen from Figure 4 that the average execution time of the key generation and encryption/decryption process of the proposed scheme is equivalent to that of the YB scheme, while our scheme is more practical than the YB scheme, such as supporting traceability and dynamic access policy update. Although the YLMH scheme's encryption cost is the smallest, its decryption cost is the largest among the three schemes and is related to the number of attributes the user has. If

the user's attributes increase, the decryption time cost of the YLMH scheme will be higher.

Figure 5 shows the algorithms' average computing time in the YB scheme, YLMH scheme, and our scheme in policy update. Since the YB scheme does not support dynamic strategy updates, we use the traditional update method. There are three modes for updating of dynamic strategy in the YLMH scheme and our scheme. We use mode 3 (which has the highest cost) for comparison. In addition, the number of AAs is fixed in 8, and the number of attributes for each user is also fixed in 8. We vary the number of attributes by 5, 10, and 15. As it can be seen from Figure 5, our scheme and YLMH scheme can dynamically update the strategy. Thus, the time cost is less than that of the YB scheme. Although our scheme costs slightly more than the YLMH scheme, our scheme supports traceability, which is considered to be more practical.

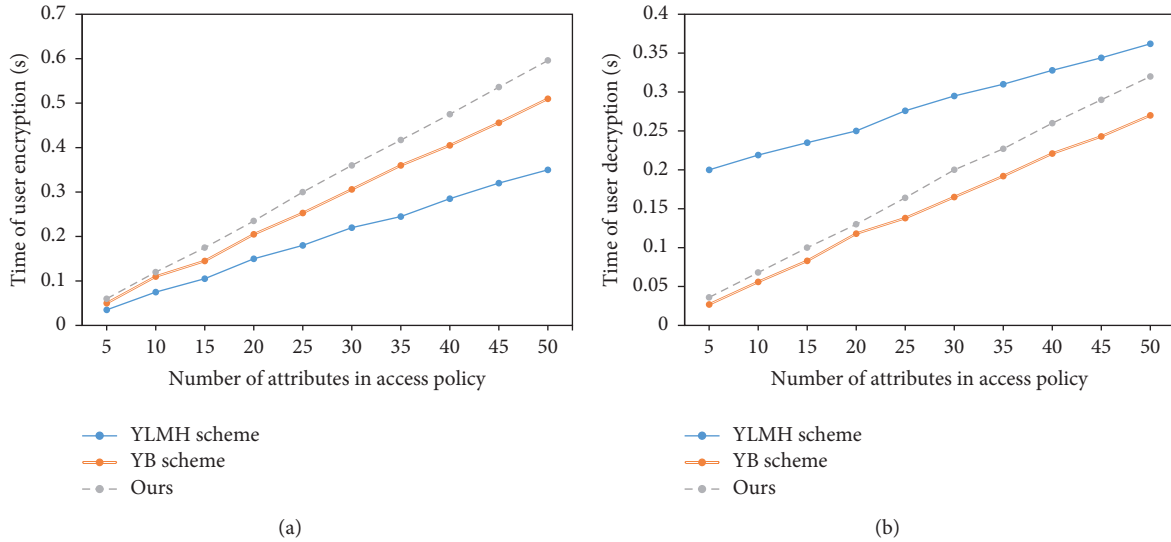


FIGURE 4: Comparison of encryption algorithms and decryption algorithms.

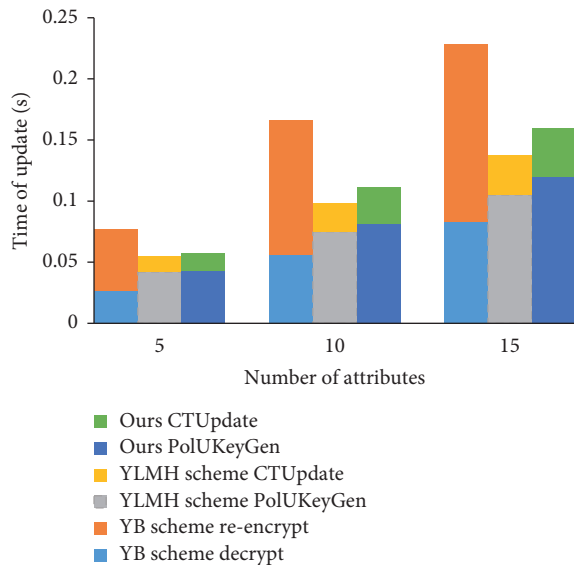


FIGURE 5: Comparison of policy updating.

7. Conclusions and Future Work

Regarding the three problems in the CP-ABE scheme of multiauthority, traceability, and the flexibility in changing the access policy, we propose a scheme to achieve good solutions. Our scheme supports multiple authorities, white box traceability, large attribute domains, access policy updates, and high expressiveness. Then, we prove that our scheme is static secure and traceable secure based on the state-of-the-art security models. By supporting the traceability, there is no need to maintain the authorized institution's identity table; thus, our solution is more practical. The experimental results indicate that our scheme has efficient performance while enjoying the abovementioned features. In future work, we plan to conduct a study on computational outsourcing and hidden access strategies for CP-ABE.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Key Areas Research and Development Program of Guangdong Province (Grant no. 2019B010139002), the Project of Guangzhou Science and Technology (Grant no. 202007010004), and National Natural Science Foundation of China (Grant no. 61902079 and 62002136).

References

- [1] E. K. Wang, C.-M. Chen, M. M. Hassan, and A. Almogren, "A deep learning based medical image segmentation technique in internet-of-medical-things domain," *Future Generation Computer Systems*, vol. 108, pp. 135–144, 2020.
- [2] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, vol. 14, pp. 1–16, 2020.
- [3] S. Amit and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Aarhus, Denmark, May 2005.
- [4] V. Goyal, O. Pandey, S. Amit, and B. Waters, "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, Alexandria, VA, USA, November 2006.
- [5] J. Bethencourt, S. Amit, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 321–334, Oakland, CA, USA, September 2007.

- [6] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *IJ Network Security*, vol. 16, no. 6, pp. 437–443, 2014.
- [7] H. Deng, Q. Wu, B. Qin et al., "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Information Sciences*, vol. 275, pp. 370–384, 2014.
- [8] S. L. Wang, J. P. Yu, P. Zhang, and P. Wang, "A novel file hierarchy access control scheme using attribute-based encryption," in *Applied Mechanics and Materials* Trans Tech Publ, Stafa-Zurich, Switzerland, 2015.
- [9] Z. Liu, Z. Cao, and S. Duncan, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 76–88, 2012.
- [10] K. Liang, W. Susilo, D. S. Wong et al., "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation Computer Systems*, vol. 52, pp. 95–108, 2015.
- [11] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 315–332, Juan, Puerto Rico, November 2015.
- [12] L. Anna, R. L. Rivest, S. Amit, and S. Wolf, "Pseudonym systems," in *International Workshop on Selected Areas in Cryptography*, Springer, Berlin, Germany, 1999.
- [13] M. Chase, "Multi-authority attribute based encryption," in *Proceedings of the Theory of Cryptography Conference*, pp. 515–534, Amsterdam, Netherland, February 2007.
- [14] L. Allison and B. Waters, "Decentralizing attribute-based encryption," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 568–588, Tallinn, Estonia, May 2011.
- [15] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proceedings of the ACM SIGSAC Conference on Computer & Communications Security*, pp. 463–474, London, UK, November 2013.
- [16] Z. Liu and D. S. Wong, "Practical attribute-based encryption: traitor tracing, revocation and large universe," *The Computer Journal*, vol. 59, no. 7, pp. 983–1004, 2016.
- [17] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1274–1288, 2015.
- [18] L. Jin, Q. Huang, X. Chen, S. M. Sherman, S. Duncan, and D. Xie, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pp. 386–390, Hong Kong, China, March 2011.
- [19] J. Zhou, Z. Cao, X. Dong, and X. Lin, "White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems," in *Proceedings of the IEEE Conference on Computer Communications*, Hong Kong, China, March 2015.
- [20] Z. Ying, H. Li, J. Ma, J. Zhang, and J. Cui, "Adaptively secure ciphertext-policy attribute-based encryption with dynamic policy updating," *Science China Information Sciences*, vol. 59, no. 4, Article ID 042701, 2016.
- [21] Z. Liu, Z. L. Jiang, X. Wang, and S. M. Yiu, "Practical attribute-based encryption: outsourcing decryption, attribute revocation and policy updating," *Journal of Network and Computer Applications*, vol. 108, pp. 112–123, 2018.
- [22] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes," *International Journal of Information Security*, vol. 17, no. 5, pp. 533–548, 2018.
- [23] M. V. Dijk, "A linear construction of secret sharing schemes," *Designs, Codes and Cryptography*, vol. 12, no. 2, pp. 161–201, 1997.
- [24] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable σ -time outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2017.
- [25] C. Jan and L. Anna, "Signature schemes and anonymous credentials from bilinear maps," in *Proceedings of the Annual International Cryptology Conference*, pp. 56–72, Santa Barbara, CA, USA, August 2004.
- [26] K. Yang, X. Jia, and K. Ren, "Secure and verifiable policy update outsourcing for big data access control in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3461–3470, 2014.
- [27] J. Ning, Z. Cao, X. Dong, and L. Wei, "White-box traceable cp-abe for cloud storage service: how to catch people leaking their access credentials effectively," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 883–897, 2016.
- [28] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, pp. 487–497, 2015.
- [29] X. Yan, Y. Liu, Z. Li, and Y. Tang, "Multi-authority attribute-based encryption scheme with privacy protection," *Journal of Computer Research and Development*, vol. 55, no. 4, p. 846, 2018.
- [30] X. Yan, H. Ni, Y. Liu, and D. Han, "Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in phr," *Computer Science and Information Systems*, vol. 16, no. 3, pp. 831–847, 2019.
- [31] K. Zhang, H. Li, J. Ma, and X. Liu, "Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability," *Science China Information Sciences*, vol. 61, no. 3, Article ID 032102, 2018.
- [32] K. Sethi, A. Pradhan, and P. Bera, "Practical traceable multi-authority cp-abe with outsourcing decryption and access policy updation," *Journal of Information Security and Applications*, vol. 51, Article ID 102435, 2020.

Review Article

Assessing Security of Software Components for Internet of Things: A Systematic Review and Future Directions

Zitian Liao ¹, Shah Nazir ², Habib Ullah Khan ³ and Muhammad Shafiq⁴

¹University of Sydney, School of Architecture Design & Planning, New South Wales 2006, Sydney, Australia

²Department of Computer Science, University of Swabi, Swabi, Khyber Pakhtunkhwa, Pakistan

³Department of Accounting & Information Systems, College of Business & Economics, Qatar University, Doha, Qatar

⁴Cyberspace Institute of Advance Technology, Guangzhou University, Guangzhou, China

Correspondence should be addressed to Zitian Liao; zitianliao@sina.com and Shah Nazir; snshahnzr@gmail.com

Received 7 December 2020; Revised 14 January 2021; Accepted 4 February 2021; Published 15 February 2021

Academic Editor: Shehzad Chaudhry

Copyright © 2021 Zitian Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Software component plays a significant role in the functionality of software systems. Component of software is the existing and reusable parts of a software system that is formerly debugged, confirmed, and practiced. The use of such components in a newly developed software system can save effort, time, and many resources. Due to the practice of using components for new developments, security is one of the major concerns for researchers to tackle. Security of software components can save the software from the harm of illegal access and damages of its contents. Several existing approaches are available to solve the issues of security of components from different perspectives in general while security evaluation is specific. A detailed report of the existing approaches and techniques used for security purposes is needed for the researchers to know about the approaches. In order to tackle this issue, the current research presents a systematic literature review (SLR) of the present approaches used for assessing the security of software components in the literature by practitioners to protect software systems for the Internet of Things (IoT). The study searches the literature in the popular and well-known libraries, filters the relevant literature, organizes the filter papers, and extracts derivations from the selected studies based on different perspectives. The proposed study will benefit practitioners and researchers in support of the report and devise novel algorithms, techniques, and solutions for effective evaluation of the security of software components.

1. Introduction

The role of component-based software engineering (CBSE) is obvious in software development. Software is designed according to previous experiences and component reusability which can save a lot of time, effort, and resources [1, 2]. Its effort is to bring commercial, cost-effective, and quality system by integrating the existing components. A system is designed using available components which is cheap, already tested, and error-free [1, 3–6]. An individual component is a single part of a software system and is a unit to facilitate reputable functionality in the system. The functionality of such components is combined which forms a complete software system. Two types of interfaces are used in a component such as provided and required interfaces.

Both of these interfaces are a source of communication inside the software system. A component can be replaced, modified, and changed according to the requirements of the system. The developments with the use of existing components can save about half of the complete developed software [7]. Compositional approaches have many benefits in the development of software systems from the appearance of development of components which has accordingly produced substantial attention in research and developments in business standards for architectures of domain-specific, component interaction, toolkits, and numerous other applicable fields.

A number of approaches exist for the security of systems [8–12]. The elementary prerequisites of security are demarcated in Availability, Integrity, and Confidentiality

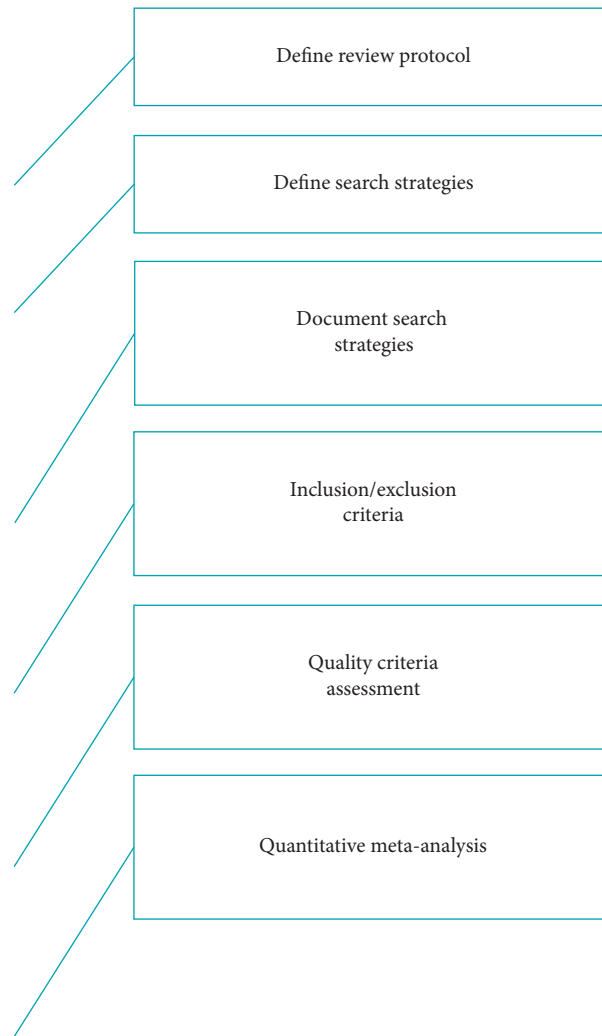


FIGURE 1: Process of conducting a systematic literature review.

[10, 13–17]. Diverse reviews, frameworks, surveys, models, and analysis affecting the IoT security for security investigation are in use. Tekeoglu and Tosun [18] offered a framework of layer-based packet capturing for inspecting IoT devices' privacy and security. Mazhelis and Tyrväinen [19] assessed platforms of IoT from application provider perceptions. Machine learning (ML) algorithms have exposed a substantial enactment in diverse applications and fields such as text recognition, facial recognition, and detection of spam. These applications of ML have understandable performance in different areas and domains [9, 12, 20–25]. The devices of the Internet of Medical Things (IoMT) are susceptible to quite a lot of security threats, attacks, and liabilities. IoMT devices are suffering commencing massive threats of security due to little costs and power, unlike typical mobile and desktop devices. The malware reproduces itself by negotiating the joining that links the devices of IoT [26]. Mao et al. [27] planned an approach for structuring dependencies of security to measure the implication of system security from an extensive perception. The consequence of small-world and power-law

distribution for in- and out-degree in security dependence networks was observed. The authors in [28] planned a method to measure the performance and services' evaluation of security for the cloud on the ground of a set of evaluation measures using Goal-Question-Metric. The authors in [29] conceived a framework for testing the security of interfaces of automotive Bluetooth with the help of a proof-of-concept tool for carrying out a test on a vehicle with the support of a planned framework. Nazir et al. [1] presented an approach for assessing software security of components via the analytic network process (ANP). The approach of ANP can work in a complex situation where the dependence arises among diverse network nodes.

The proposed research presents an SLR of the existing approaches used by practitioners to protect software systems. The protocol followed for conducting the proposed study is based on [3]. The study searches the literature in the popular and well-known libraries, filters the relevant literature, organizes the filter papers, and extracts derivations from the selected studies based on different perspectives. The following key contributions are achieved by the proposed study:

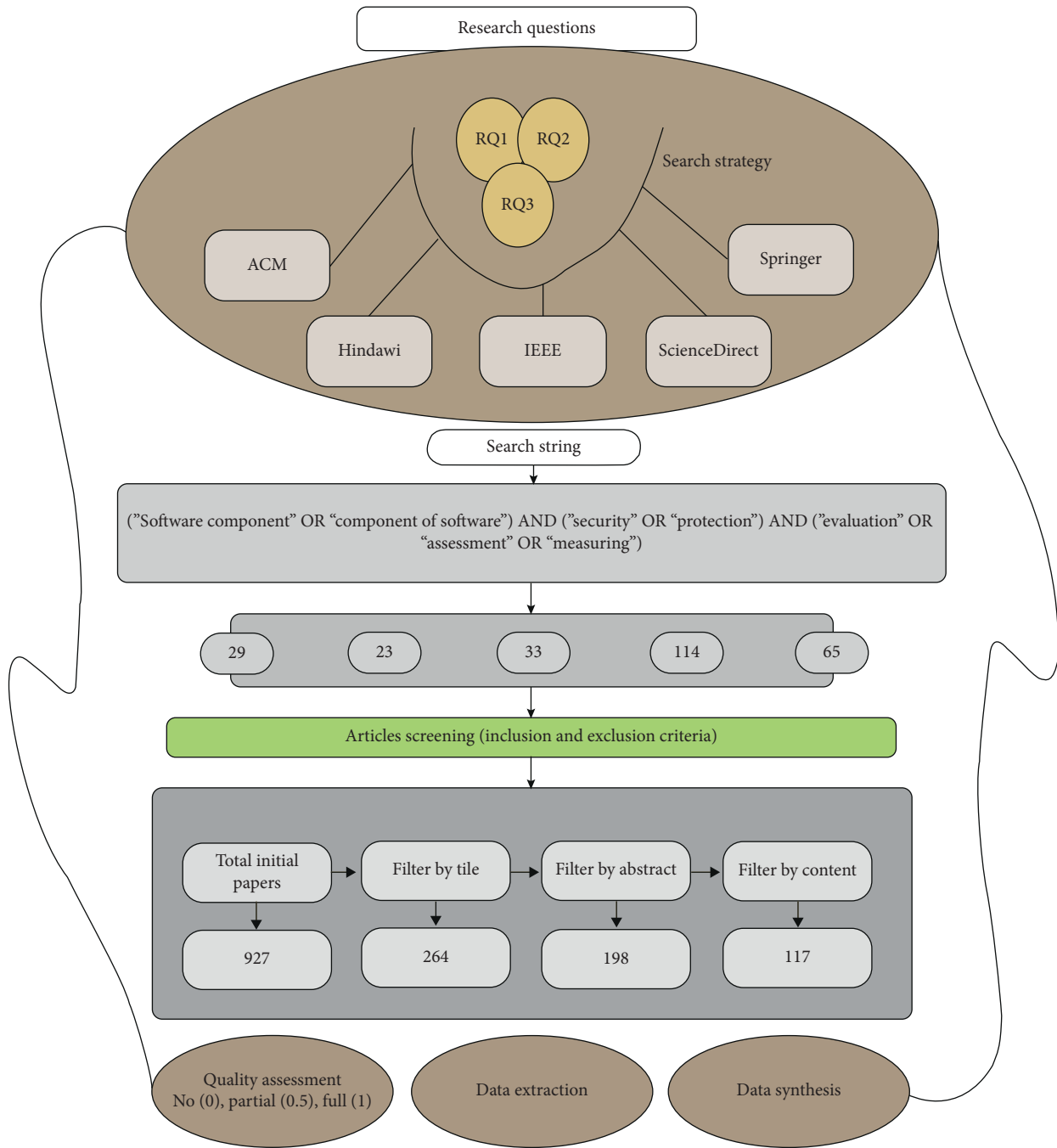


FIGURE 2: Protocol process and the libraries.

- (i) To study the security measures for assessing software security of components
- (ii) To identify the techniques and methods available for assessing software security of components
- (iii) To show how these techniques efficiently work for evaluating the security of components

The paper is structured as follows. Section 2 shows the research method focusing on SLR for showing the analysis of the current study. Section 3 shows the results and discussions of the paper with answers to the research questions. The conclusion is presented in Section 4.

2. Methodology

2.1. Research Plan and Process. The SLR is a formal way of searching the keywords, identifying the relevant materials associated with the research, organizing in an efficient way, and deriving meaningful information and derivations from the studies selected. Figure 1 represents the steps followed for the proposed research where firstly the review protocol is defined, then the search strategies are defined for the research, then the search strategies are documented, the relevant materials are included while the rest of the materials are excluded, the quality assessment is done for the selected

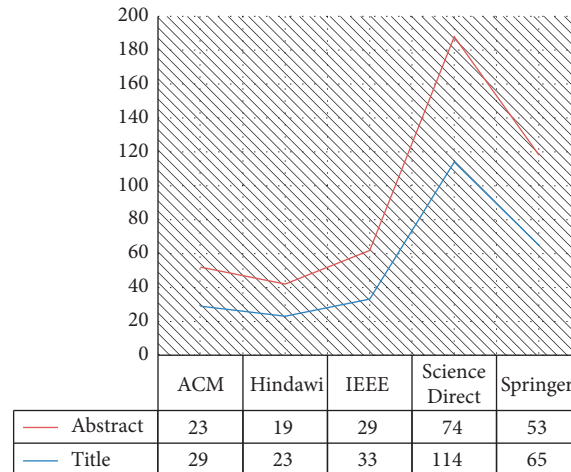


FIGURE 3: Overall search results.

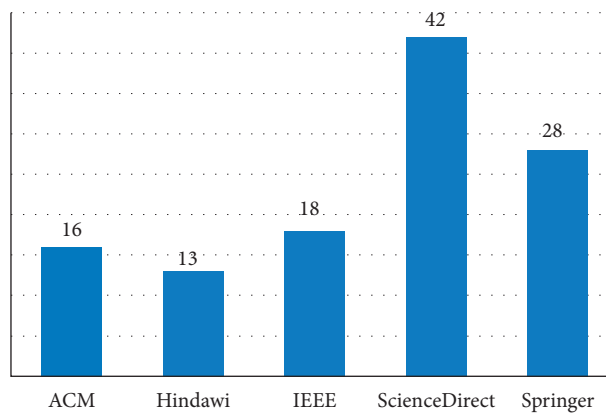


FIGURE 4: Final filtered papers by contents.

papers, and lastly the data analysis is extracted from the included papers.

2.2. Research Questions. Below are the questions which were defined for the current study:

- (1) What can be the security measures for assessing software security of components?
- (2) What are the techniques and methods available for assessing the security of software components?
- (3) How efficiently the techniques work for evaluating component security?

2.3. Keywords and Libraries. The keywords (“Software components” OR “components of software”) AND (“security” OR “protection”) AND (“evaluation” OR “assessment” OR “measuring”) were defined to search the libraries. The following libraries were adopted for the process of search. Other libraries were skipped due to the reason that these

libraries are publishing materials which are peer-reviewed, while Google Scholar has all of the materials.

- (i) ACM
- (ii) Hindawi
- (iii) IEEE
- (iv) ScienceDirect
- (v) Springer

The following are the details of the process of the search for each of the selected library.

- (i) ACM: [[[All: “software components”] OR [All: “components of software”]] AND [[All: “security”] OR [All: “protection”]] AND [All: (] OR [All: (] OR [All: “evaluation”] OR [All: “assessment”] OR [All: “measuring”]]
- (ii) Hindawi: (“Software components” OR “components of software”) AND (“security” OR “protection”) AND (“evaluation” OR “assessment” OR “measuring”)

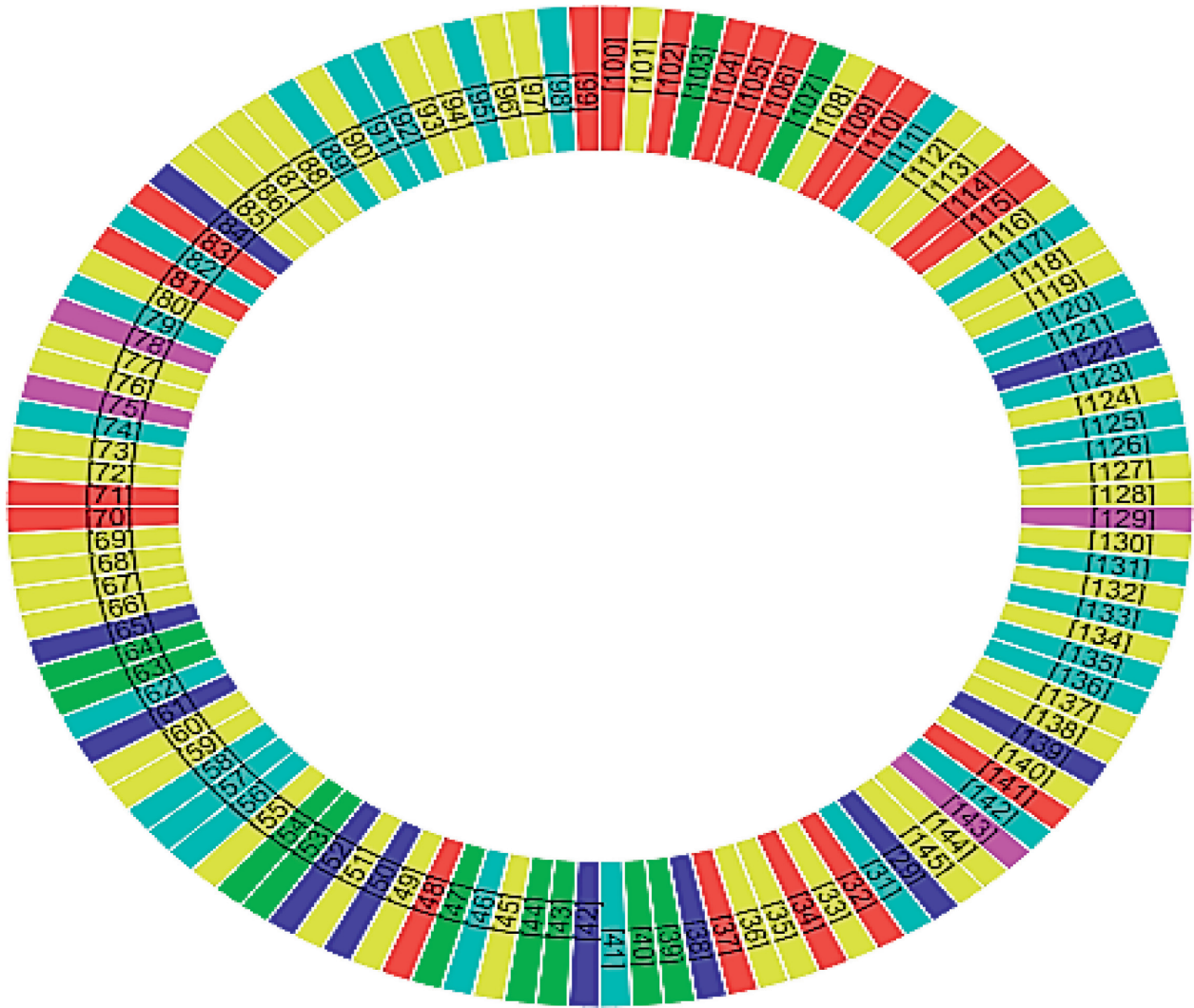


FIGURE 5: Selected articles.

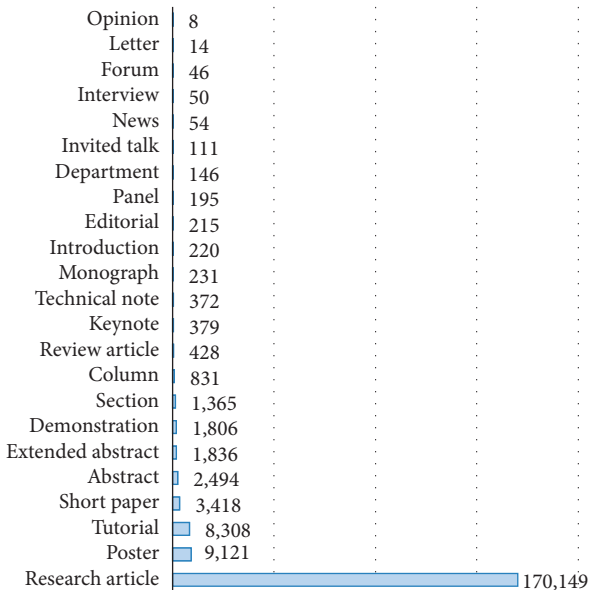


FIGURE 6: Article types and the number of papers.

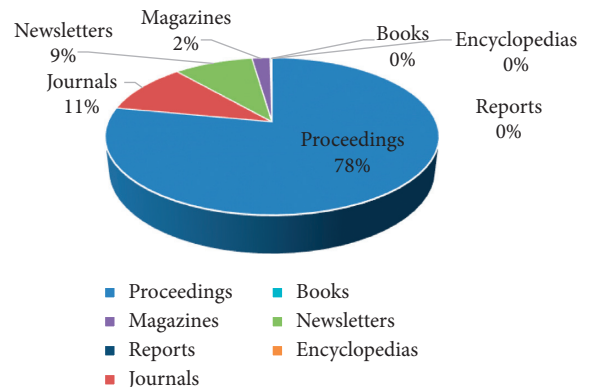


FIGURE 7: Article types and publications.

(iii) IEEE: (“All Metadata”:Software components) OR “All Metadata”:components of software) AND “All Metadata”:security) OR “All Metadata”:protection) AND “All Metadata”:evaluation) OR “All Metadata”: assessment) AND “All Metadata”:measuring)

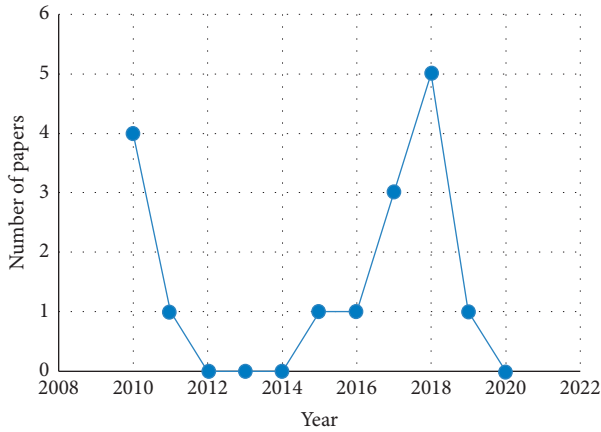


FIGURE 8: Year and number of papers published.

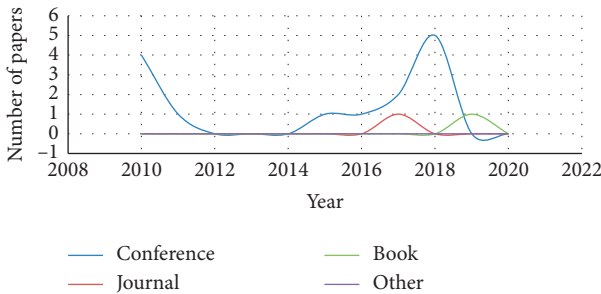


FIGURE 9: Year and number along with the type of paper published.

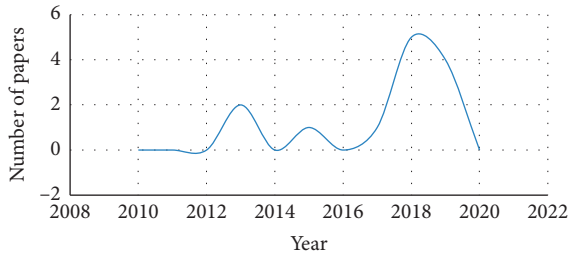


FIGURE 10: Year and total of articles.

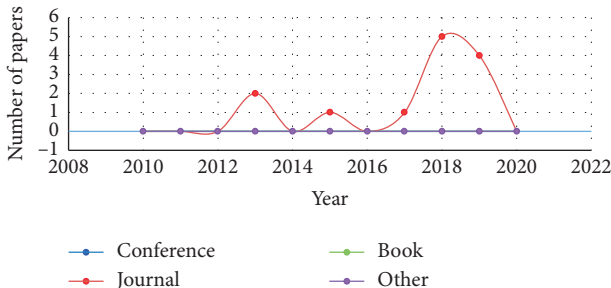


FIGURE 11: Year and number along with the type of paper published.

(iv) ScienceDirect: “(“Software components” OR “components of software”) AND (“security” OR “protection”) AND (“evaluation” OR “assessment” OR “measuring”)”

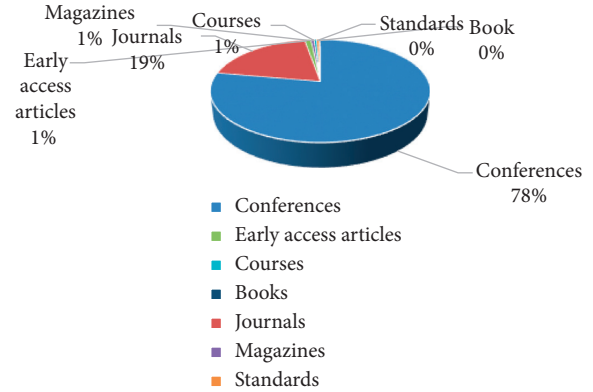


FIGURE 12: Number of publications along with the type of publication.

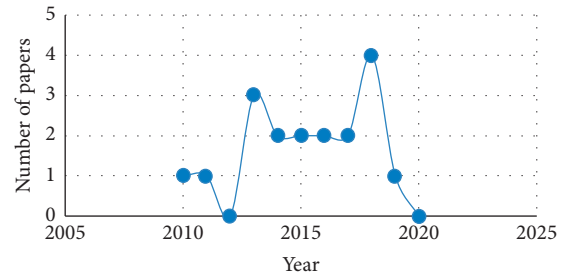


FIGURE 13: Year and the total number of articles.

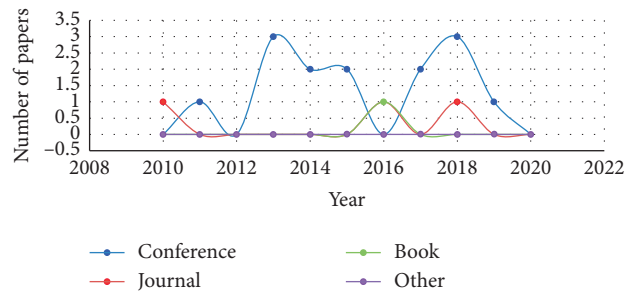


FIGURE 14: Year and number along with the type of paper published.

(v) Springer: “(“Software components” OR “components of software”) AND (“security” OR “protection”) AND (“evaluation” OR “assessments” OR “measuring”)”

Figure 2 shows the process of searching the keywords in the given libraries with the results of the search obtained. The filtering process of papers by title, abstract, and finally contents is also shown in the figure. The figure is initially based on the research questions defined and then the search process in the given libraries with the use of Boolean operators “AND” and “OR.”

Figure 3 shows the number of papers filtered by title and then by an abstract in the given libraries. Initially, huge

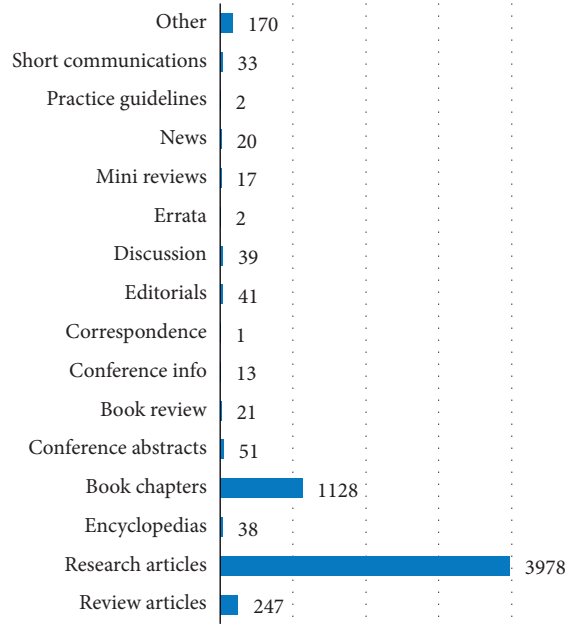


FIGURE 15: Publication type and the number of papers published.

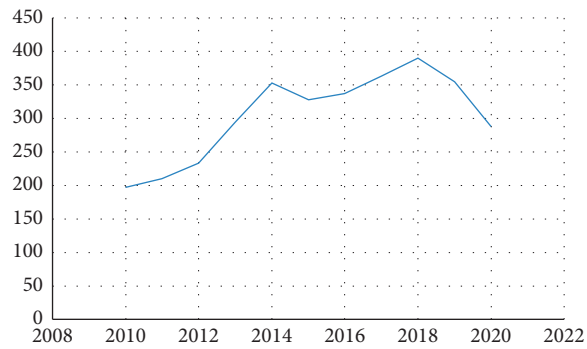


FIGURE 16: Number of publications in the given year.

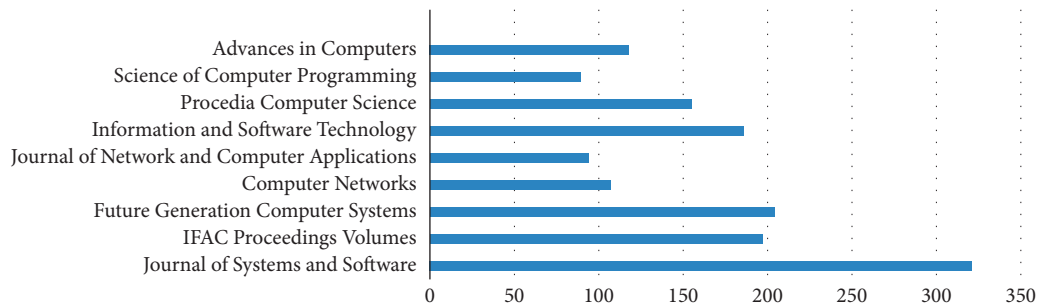


FIGURE 17: Publication title and number of papers.

numbers of papers were obtained during the search process. It was considered that the analysis of all the searched papers was difficult, so due to this reason, the papers were filtered by title for obtaining the relevant papers. After this, a total of 264 papers were obtained which was also difficult to analyze in one process, so these articles were then filtered by abstract, and a total of 198 articles were achieved.

The articles were filtered based on content, and a total of 117 articles were achieved for the given libraries which are shown in Figure 4.

The articles selected are shown in Figure 5.

After this, the details of each library were analyzed which are given hereinafter. The library of ACM was analyzed in the first step for the research article type and content type. This search was for the initial results of the search which is shown in Figure 6.

The article type for the ACM library is shown in Figure 7.

After the initial search process, the materials were filtered to extract only relevant studies. Figure 8 shows the articles published in the mentioned years.

The article types were viewed in the given year. Figure 9 depicts article types and the total number of articles in given years.

After searching the ACM library, the library of the Hindawi publisher was checked for relevant materials related to the proposed study. Figure 10 presents year-wise publication numbers in the library of Hindawi.

Figure 11 represents the total number of articles published in given years based on the types of publications.

The library of IEEE was searched for identifying relevant studies to the proposed research. Figure 12 shows initial search results for publications with publication types in the IEEE library.

The obtained papers from the searched process in the IEEE were then filtered to extract only relevant papers. Figure 13 shows the total number of articles in given years in the IEEE library.

Figure 14 presents publication types with publication numbers in given years in the same library.

The library of ScienceDirect was considered to find the relevant materials to the proposed research. During the initial search process, the publication types were checked which is shown in Figure 15.

The total number of articles was checked in given years. The total number of articles with the year of articles is presented in Figure 16.

The publication titles were also checked that where the papers are published. Figure 17 presents the titles of the articles with a total number of articles.

After filtering the process of papers in the ScienceDirect library, the number of articles in given years was reviewed. The details are given in Figure 18.

Figure 19 presents the total number of articles with the types of publications in given years.

Finally, the library of Springer was searched to obtain the associated material to the proposed research. The initial search results for the number of publications with article types are shown in Figure 20.

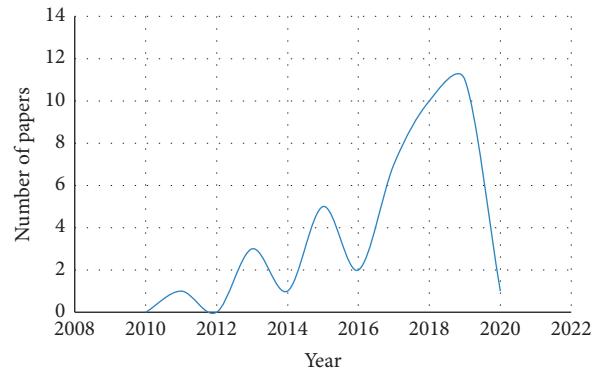


FIGURE 18: Number of publications in the given years.

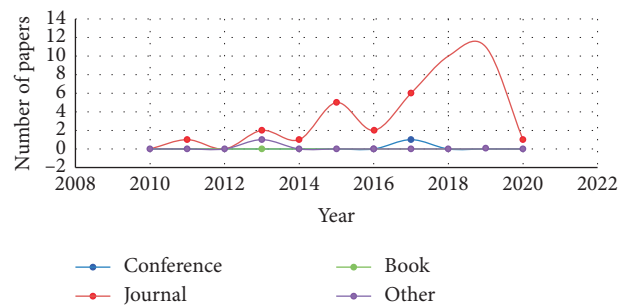


FIGURE 19: Article type with the total number of articles.

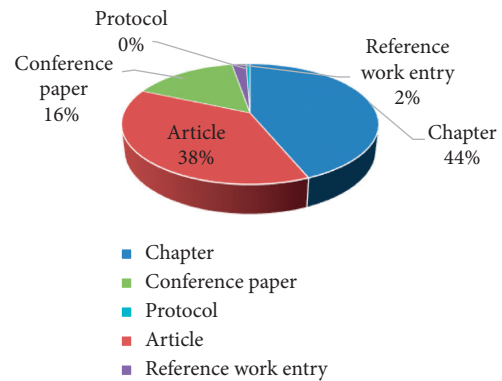


FIGURE 20: Articles with the type of papers.

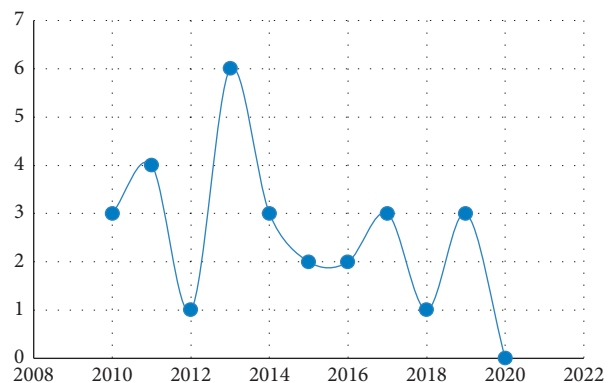


FIGURE 21: Number of publications in the given year.

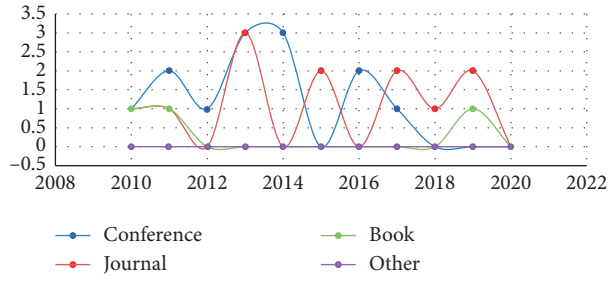


FIGURE 22: Article type with the total number of papers in the given year.

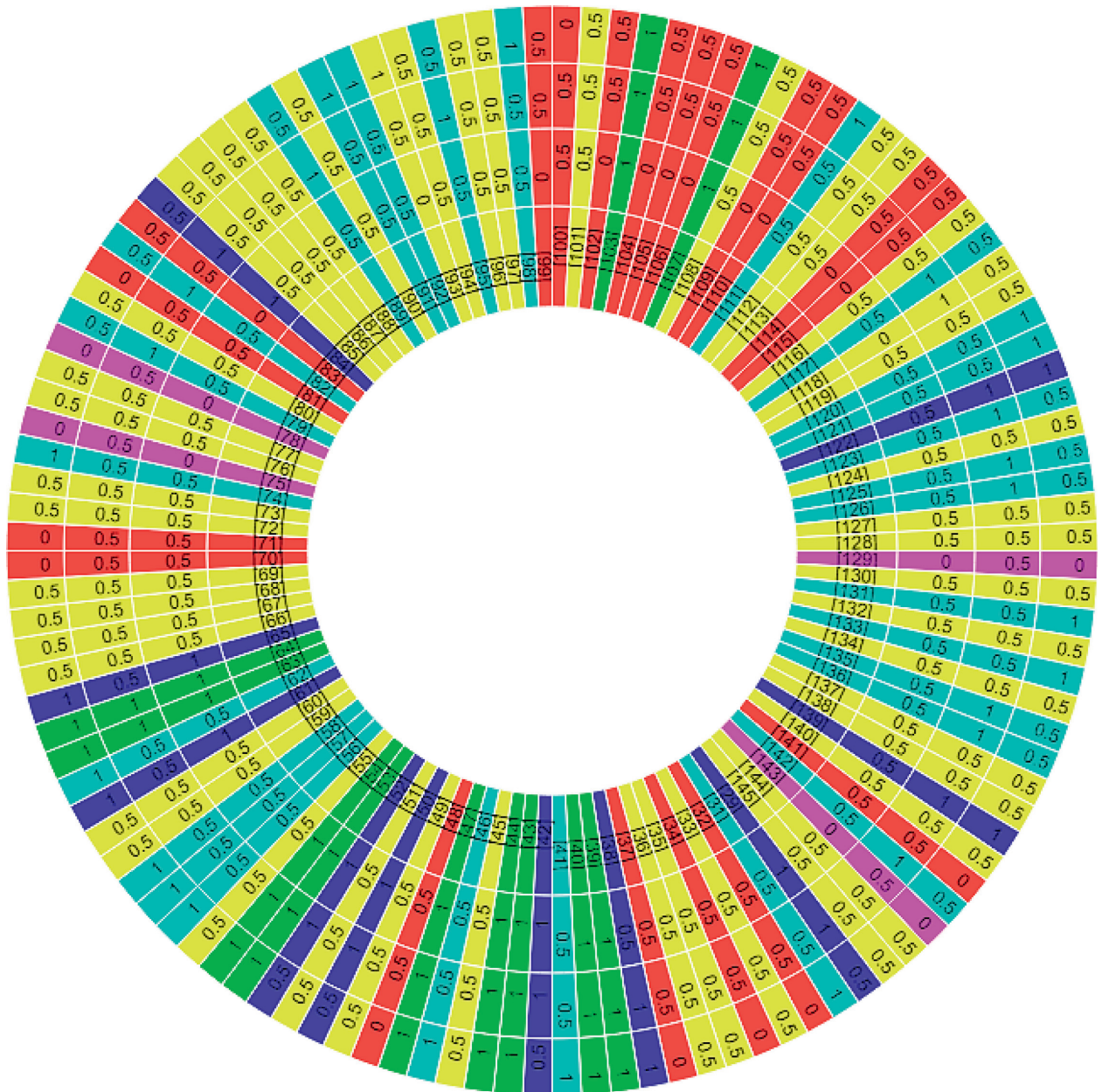


FIGURE 23: Score of research questions for each paper.

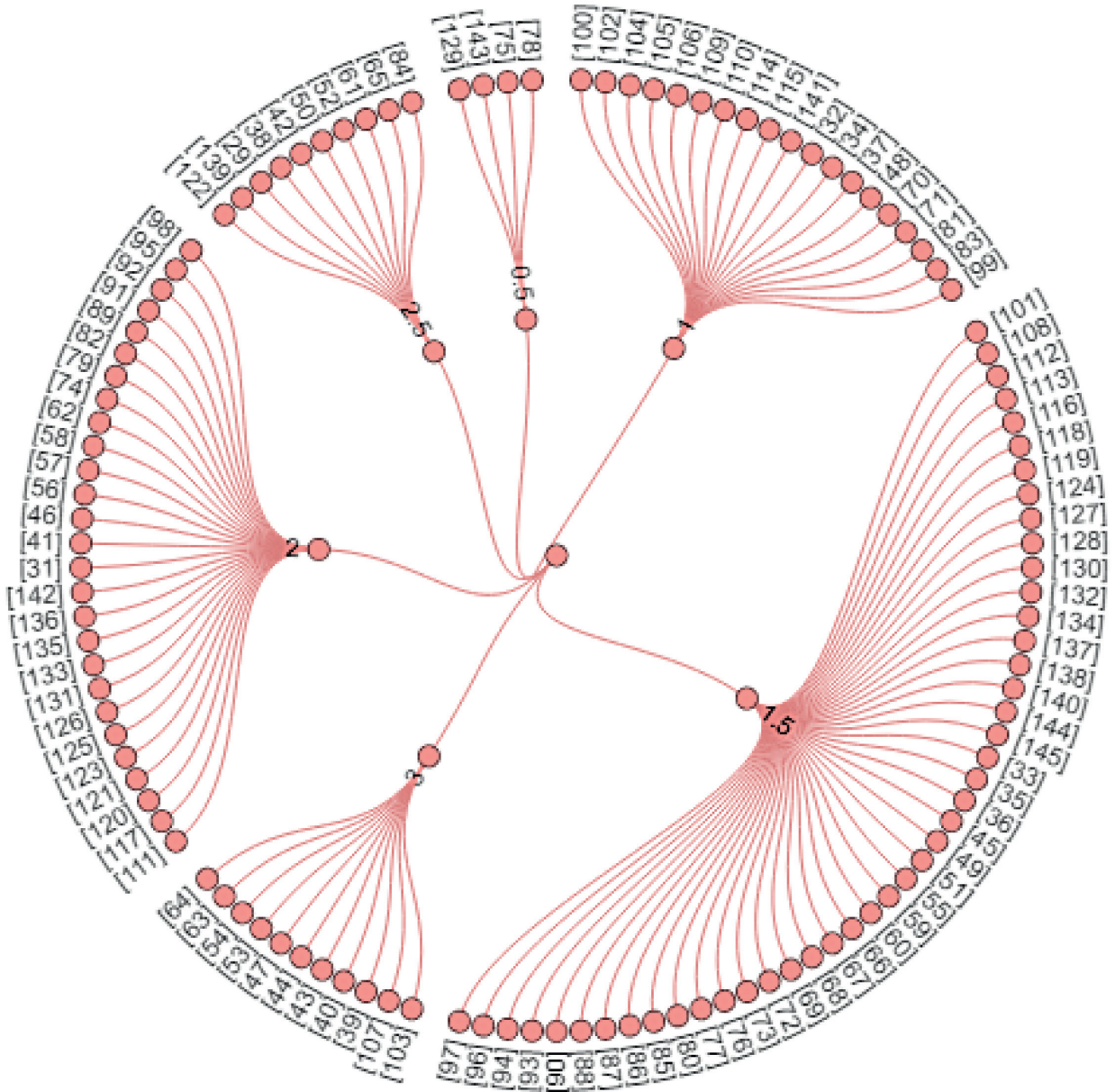


FIGURE 24: Sum of scores for each paper.

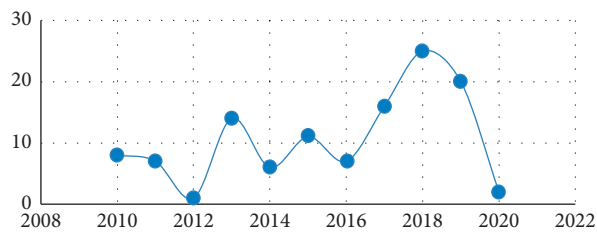


FIGURE 25: Overall number of papers in all the libraries in the given years.

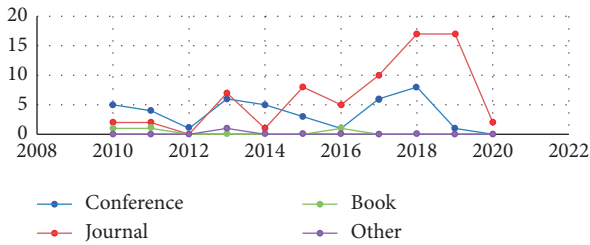


FIGURE 26: Overall number of papers and type of papers in all libraries in the given years.

TABLE 1: Identified list of security features presented by researchers.

References	Features
[30]	Maximum signal range Variety of network topologies Safety and security of data transfer Reliability and dependability of WCT Throughput and data rate Applicability of WCT Wireless power transfer Minimum latency
[17]	Confidentiality Availability Integrity Authentication Access control Authorization Auditing Trust Privacy Reputation metering Accountability Replay protection Anonymization Resilience to attacks Fault tolerance Nonrepudiation
[31]	Confidentiality Integrity Availability Fault tolerance Accountability System trust
[32]	Privacy and security Technology Communication Culture Job Legal regulation
[33]	Privacy protection Node information certificate Secure cloud computing Encryption mechanism Anti-DDoS Platform security Secure multiparty computation Information application security Antiattack security Heterogeneous network recognition

After filtering the process of papers, the results were analyzed to obtain meaningful results related to the proposed research. Figure 21 represents article numbers in the given year in the library of Springer.

Figure 22 represents the total number of publications with the type of publications in the given year in the Springer library.

2.4. *Quality Assessment of the Selected Papers.* The quality assessment process of the carefully chosen articles was done in order to know how much the paper is related to the proposed study. A score of “1” was given to the research paper which completely fulfills the research question, “0.5” was given to the paper somewhat satisfying the research question, and “0” was given to the paper not satisfying the research question. Figure 23 shows the quality score for each paper based on the defined research questions.

Figure 24 shows the sum of the overall score for each paper. The assigned values of the selected papers for all the research questions were summed and the total score is shown in the figure.

3. Results and Discussion

After individual analysis of the libraries, all the references were merged into a single Endnote file to analyze them. It was found that there is an increase in the year-wise number of publications related to the proposed research. Figure 25 shows the number of publications in the given years for the overall libraries collectively.

Figure 26 shows the number of publications along with the type of publications in the given years for all the libraries collectively.

3.1. *What Can Be the Security Measures for Assessing the Security of Software Components?* Security features can play a significant role in the smooth running of a particular system. A number of features were identified from the literature based on which the security is evaluated. Table 1 shows the identified list of features from the literature presented by different researchers.

3.2. *What Are the Techniques and Methods Available for Assessing the Security of Software Components?* Diverse approaches are presented by the researchers to tackle the issue of security evaluation of software and its components. These approaches work from different perspectives. Table 2 shows the summary of the existing techniques available for security evaluation.

3.3. *How Efficiently the Techniques Work for Assessing the Security of Components?* There is high need of effective security evaluation techniques which can efficiently evaluate the security of software system. Such techniques

TABLE 2: Existing approaches for evaluating security.

Citation	Technique	Description
[34]	Quantitative assessment approach	This approach evaluates the component security level quantitatively and identifies efficiently the component security vulnerabilities.
[35]	Secure multiparty computation (SMC)	This paper revisits the history of developments to SMC that completed the years and studies the opportunity of coupling reliable hardware with SMC.
[36]	Software-defined networking (SDN)	The analysis demonstrated that SDN appears to be the most attractive developmental structure for upcoming networks.
[37]	Conventional security mechanisms	They focus on emerging security threats aiming at vulnerabilities, human errors, and defects of a mobile device structure in existing schemes.
[38]	Abstract network model	The analysis shows that the abstract network model is a valuable method for attack graph-based assessments.
[39]	Logic programming	In this article, model-based testing and logic programming was introduced for detecting accessible SQL injection (SQLI) and cross-site scripting (XSS) of web applications.
[40]	Cognitive dimensions questionnaire	Results revealed that the usability issues of security application programming interfaces (APIs) may be determined using this methodology with significantly good reliability and validity.
[28]	Goal-question-metric (GQM) method	The proposed assessment methodology might help cloud service providers (CSPs) to practice a security self-evaluation and is suitable for the level of their security services within the cloud market.
[29]	Threat model	This model is helpful for the evaluation of the Bluetooth interface on a range of built-in automotive infotainment systems.
[41]	Security assessment	This study presents the cybersecurity associated principles for the smart grid which address the issue in different ways and to various extents.
[42]	Semantic model	In this paper, a semantic model for structuring and risk visualization implemented into the metric visualization system (MVS) was presented.
[43]	NIST national vulnerability database (NVD) combined with EBIOS risk analysis and evaluation methodology	The finding of this research has demonstrated that virtual networks, SDN controllers, and hypervisors continue to present new attack capabilities that are continually being exposed, further escalating the security risk of modern data centers.
[44]	Security behavior	The research findings show that psychological ownership, descriptive norm, response cost, self-efficacy, and perceived vulnerability all were significant in determining personal computing security intentions and behavior for both the mobile device and home computer users.
[45]	Countermeasure-centered approach	In this article, a prototype implementing such a security management system is described.
[46]	Threat model	This work presents a quantitative study on the security solutions for communication quality used in robotics, while security capabilities are enabled.
[47]	Supervisory control and data acquisition (SCADA) systems security	This provides an insight into developing a framework that can be used to assist critical infrastructure sectors.
[48]	Innovative ontology and graph-based approach	For network security evaluation, an innovative approach that uses ontology was proposed. The ontology is intended to illustrate security knowledge such as that of attacks, vulnerabilities, assets, and the relationships between them.
[49]	Information-theoretic model	For the computer systems security analysis, the entropy concept was utilized and a quantitative model was derived. The assessment process consists of dynamic and static phases.
[50]	International symposium on formal methods (FM 2012)	This short paper is intended to accompany a talk at the 18th international symposium (FM 2012). It discusses software security with a highlight on formal aspects, defenses, and low-level attacks.
[51]	Security metrics and risk analysis	In this work, formal analysis of associations between risk and security metrics and formal definition of risk were provided.
[52]	Security information and event management (SIEM) systems	The article proposed a general framework for the visualization of SIEM which permits integration of different visualization approaches and expands simply the application functionality.

TABLE 2: Continued.

Citation	Technique	Description
[53]	Big data framework	A framework for big data in this work was proposed to build up the security capability of small enterprises.
[54]	Usability of security software	This article addresses the usability of security alerts across a wider range of security products.
[55]	Security evaluation using Bayesian belief networks	This article demonstrates parts of the gap, in particular the challenges associated with variable quality of information, lack of empirical information, limited budget, short time-to-market, and lack of resources.
[56]	Multimetrics approach for security	This article presents a multimetric approach jointly with a methodology to estimate the system security, privacy, and dependability (SPD) level throughout both the running and design process.
[57]	Ontology-based model for security assessment	In this article, the ontology-based framework was classified in five dimensions for assessing attack effect; they are defense, vulnerability, attack target, attack vector, and attack impact.
[58]	Vulnerability-centric requirements engineering framework	This paper gives an engineering framework to maintain the elicitation of security requirements and analysis based on vulnerabilities.
[59]	Evaluation and assessment of the security of wearable devices	This paper examined the usefulness and design of SecuWear platform for recognizing vulnerabilities in these areas and assists wearable security research to mitigate them.
[60]	Assessment of platforms	This paper explains how the PRIME platform trust can enhance trust and manager operates.
[61]	Software-defined security framework	For protecting the distributed cloud, a software-defined security framework was proposed in this paper.
[62]	Software-defined mobile network security	This article gives a survey of software-defined mobile network (SDMN) and its related security issues.
[63]	Reputation model	In this article, the most critical as well as essential security threats for a utility-based reputation model in grids were assessed.
[64]	IoT monitoring solution	A monitoring tool based on the extension of the Montimage network monitoring tools for IoT systems was presented in this paper.
[65]	A comprehensive pattern-driven security methodology	ASE—a comprehensive pattern-driven security methodology intended particularly for (common) distributed systems—focuses on the early life cycle phases and particularly the design phase.
[66]	Contract-based security assertion monitoring	This article demonstrates how in a live environment on Linux a contract-based security assertion monitoring can be attained.
[67]	Network security visualization	For the security visualization systems evaluation such as ranking and rating, a framework was proposed in this paper.
[68]	Empirical study	This article empirically examines how refactoring can progress the security of an application by removing code bad smells.
[69]	Computational approach	For the standardization of the software development process, a computational approach was proposed in this work.
[70]	Multitarget approach	In this paper, for the estimation of scores and vulnerability characteristics from the technical description, a model of the combination of multitarget classification and text analysis approaches was created.
[71]	A new threat identification approach	In this paper, for the assessment of security threats quantitatively, a new approach was adopted, which is modular, extendable, and systematic.
[72]	Regression model	For the identification of security requirements, a linear based approach was proposed in this work.
[73]	Problem-oriented security patterns	Based on the problem frames technique, a systematic approach was proposed in this work for the iterative development of software architectures and requirements analysis.
[74]	A framework for semiautomated coevolution	For the security maintenance and support, a model-based framework was addressed in this paper for a software system during the long-term evolution.
[75]	A manual approach	The legal and security risks were discussed in this paper which arise from reuse.
[76]	A coarse approach to quantitative modeling and analysis	For the integrated vulnerability assessment, a methodology using a coarse approach to quantitative analysis and modeling was discussed in this paper.

TABLE 2: Continued.

Citation	Technique	Description
[77]	Cyberdefense and cloud vulnerability assessment	In order to decrease, evaluate, and assess the vulnerability level of distributed computing systems (DCIs), an IT security audit framework was created in this paper.
[1]	Analytic network process (ANP)	For the component security evaluation, an ANP was proposed in this paper.
[78]	Distributed security systems	Distributed security systems were examined in this paper with devoted server modules that perform client modules' monitoring and managing.
[79]	Threatened-based software security evaluation method	In software security literature, for the software security assessment, a new concept was introduced in this paper: the threatened-based method.
[80]	Measurement frameworks	This paper reports a measurement framework for software development.
[81]	A cloud data monitoring system	Based on autonomic computing, a data security monitoring approach was proposed in this paper for the feasibility verification through simulation.
[82]	Hybrid reputation model	Based on both explicit definition of reputation and implicit reputation calculation, a hybrid reputation model is presented in this article.
[83]	Security architecture	In this paper, the implementation and design of a security framework to FPGA-based heterogeneous systems developed on top of MAC-based OS/Hypervisors was presented.
[84]	Website security analysis	A model-based website security testing method was proposed in this paper.
[85]	Methodology for enhancing software security	For enhancing software security in the development life cycle, a methodology was proposed in this paper.
[86]	Dynamic disassembly of machine instructions	This paper talks about a novel concept RECSRf, consisting of the runtime execution complexity (REC) and its evaluation method security risk factor (SRF).
[87]	Protection of IoT devices using Berkeley packet filters	This paper reports a practical approach which is an easy-to-use framework to protect IoT devices against attacks.
[88]	Software security knowledge	For the secure software development that incorporates an artifact and a knowledge-based management system, a case-based management system (CBMS) was proposed in this work.
[89]	Security analysis of android applications	This paper addresses a mobile app security investigation tool StaDART that merges dynamic and static examination to present the existence of dynamic code update.
[90]	Surveys and overviews	This paper summarizes the field of software vulnerability examination and discovery that uses machine learning and data mining approaches.
[91]	Security and privacy	This paper talks about safe patch fingerprinting.
[92]	Text mining	This paper focuses on text mining approaches and their different classification techniques (support vector machines, neural networks, and decision trees).
[93]	Software security engineering	This paper described an attempt to benchmark and baseline the state of company software and also incorporates state of software reliability data across the company's products.
[94]	Quantitative measurement	In this paper, for software engineering service bus (EngSB) platform assessment, a set of quantitative metrics was proposed.
[95]	Common vulnerability scoring system	This article reports which information cues decrease or increase vulnerability evaluation by humans.
[96]	Automatic approach	In this article, an automatic approach was proposed for detecting the software vulnerabilities on multiple systems using/sharing API libraries or similar code.
[97]	Software and application security	This paper talks about the software vulnerabilities by means of descriptions only via deep learning and word embedding approaches.
[98]	Threat analysis	This paper talks about the threat agent approach.
[99]	Machine learning techniques	This paper reports a lightweight dynamic and static features approach for the software vulnerability testing detection by means of machine learning methods.
[100]	Models of computation	In this paper, a cryptographically secure attestation scheme was proposed, which detects direct memory access (DMA) attacks.
[101]	Understanding security requirements and challenges	This work describes the state-of-the-art efforts in ensuring security in the IoT network.

TABLE 3: Summary of the existing techniques for evaluating security.

Citation	Technique	Description
[102]	A framework for the comparison of security adaptation approaches	Five security adaptations were compared in this framework. The framework includes three perspectives that are life cycle, security, and adaptation. The evaluation illustrated that in each adaptation approach the monitor and analysis phase is described.
[103]	Information security risk assessment	The analysis showed that this method gets more scientific evaluation and reliable and stable results on the evaluation of the risk of the control systems of industry.
[104]	State fusion finite state machine model	In this paper, an SF-FSM model was proposed to recognize a legitimate application to evaluate its vulnerabilities and illegal behavior of unauthorized parties for an industrial control system.
[105]	Core unified risk framework (CURF)	This approach is suitable for the qualitative comparison of activities and processes in each method of information security risk assessment (ISRA) and presented a measure of completeness.
[106]	Complexity metrics for software security improvement	For the security level of computer-based systems, improving software security is essential.
[107]	Security vulnerability assessment, prevention, and prediction (SVAPP)	The proposed SVAPP methodology exploits an active security barrier approach and adapts it to suit the security facet.
[108]	Security quality requirements engineering (SQUARE) method	In this paper, SQUARE effectiveness was evaluated in terms of its artifacts (attack tree, security templates, system architecture diagram and use-case diagram, and scenarios), a set of security goals, vulnerabilities, threats, and prioritized and categorized security requirements.
[109]	SODA	In this paper, SODA was introduced, which leverages integrate virtual network functions (VNFs) and software-defined networking (SDN) to realize service management and security policy for IoT environments.
[110]	Evaluating of security risks framework	In this article, the security risks for IEC 61850 network, intelligent electronic devices (IEDs), and distributed denial of service (DDoS) attack assessment within an SDN-enabled smart grid communication network.
[111]	Security analysis and security rules	This analysis investigates four in-app payments' implementation and also summarizes a series of security rules.
[112]	Formal framework	In this paper, a formal framework for the strength of software obfuscation evaluation was proposed. It is used for the protection of secret data or control-flow graphs (CFGs) of a program.
[113]	Machine learning methods	The contribution of this paper is a methodology for analyzing features from C source code to classify functions as vulnerable or nonvulnerable.
[114]	UML or SysML language	In this article, the state of the art associated with quantification, verification, and security specification for systems and software that are modeled by means of UML or SysML language is reviewed.
[115]	Security diagnosis as a service (SDaaS)	The scalability, performance, and accuracy of the framework were evaluated. The results of the evaluation reveal that SDaaS demonstrates information flow vulnerabilities with not merely scalability, performance, and accuracy, but furthermore lightweight footprint on resource utilization.
[116]	Calculus IoT-LySa	This article presents a methodology, based on the process calculus IoT-LySa, to infer quantitative measures on the evolution of systems.
[117]	Framework for modeling and assessing the security of the Internet of Things (IoT)	The IoT is facilitating innovative applications in a variety of domains. The key contributions of this article were to assess the framework using three scenarios, including environment monitoring, wearable healthcare monitoring, and smart home.
[118]	Broadcasting service	This article describes and records all probable threats to broadcasting services
[119]	Security in software evolution	In this chapter, four challenges including relevant knowledge, the impact of available knowledge, reestablishing, and reactions of security were addressed.
[120]	Framework for security testing	In this article, the proposed framework is used for security testing subsequent to the system implementation.
[121]	Multiperspective security management	The projected modeling approach for managing and designing IT security in institution account used for diverse perceptions is based on multiperspective enterprise modeling.
[122]	Embedded device design and verification	This paper focused on the approaches for verification and design of information systems with embedded devices.

TABLE 3: Continued.

Citation	Technique	Description
[123]	Automotive security assurance	In this article, a systematic security assessment to specify undesirable behaviors, enabling the assignment of severity ratings in a (semi-) automated manner was explored.
[124]	Pattern-based method	In this paper, for establishing a cloud-specific information security management system (PACTS), a pattern-based method was presented.
[125]	Temporal hierarchical attack representation model	In this article, network changes were systematically formalized and categorized on the basis of their causes of the change.
[126]	Stochastic modeling	For the security metrics quantitative assessment, a state-based stochastic model was proposed in this paper.
[127]	Experimental assessment	In the presence of denial of service (DoS) attacks for the assessment of the security of web service frameworks, an experimental approach was proposed in this article.
[128]	Hash power distribution analysis model	In this article, a hash power distribution analysis model for the profitability of miner measurement was proposed based on various incentives toward an evaluation of Bitcoin security.
[129]	mHealth apps security framework (MASF)	To secure the execution of mHealth apps and their users' data, the mHealth apps security framework (MASF) was proposed in this article.
[130]	Abstract model	In this article, for the support of single sign-on (SSO) development, an abstract model was provided.
[131]	A proactive approach	To quantitatively assess the security of network systems, a proactive approach was addressed in this paper for validating, formulating, and identifying a number of essential features that mostly affect its security.
[132]	Trust modeling and evaluation	For a component-based software system, an autonomic trust management solution was introduced in this paper.
[133]	Static analysis	For the security static analysis tools, an evaluation framework was introduced in this paper.
[134]	SecuWear platform	This paper presents a multicomponent research platform, called SecuWear, for mitigating, analyzing, and testing vulnerabilities in software and hardware.
[135]	One-to-many bilateral e-trade negotiation framework	A mobile agent-based secure one-to-many bilateral e-trade negotiation framework was presented in this paper.
[136]	Model integrated computing	For rapidly deploying cyberphysical system (CPS) attack experiments, a model-based software development framework integrated with a hardware-in-the-loop (HIL) testbed was presented in this work.
[137]	Concise binary object representation (CBOR)	This paper reports instantiated architecture for verification and secure measurement of dynamic runtime information for Linux-based OS.
[138]	Multidomain networks	In this article, a framework was proposed for leveraging service function chaining (SFC) and software-defined networking (SDN) to improve collaboration among security service functions (SSFs).
[139]	Security-informed safety	This paper talks about security-informed safety.
[140]	Trust model	In this article, for cloud-edge-based data-sharing infrastructure, a 5 level trust model was proposed.
[141]	Security and risk assessment	This paper gives suggestions about unmasking the uncertainty of risk assessment and facilitating oversight of its practice by public actors, judicial and legislative.
[142]	Software security vulnerabilities	In this work, for recurring software vulnerabilities, an empirical study was reported.
[143]	Self-destructive tamper response	In this paper, a method for tamper-resistant software was created, so as to be resistant to dynamic analysis as well as static analysis.
[144]	Model of virtual machine (VM)	Based on memory introspection, a model of VM security monitoring was proposed in this article.
[145]	Software-defined networking (SDN)	This paper reports the NOSArmor, which contains various security mechanisms, such as a security building block (SBB), into a consolidated SDN controller.
[146]	Binary-level patch analysis framework	SPAIN which is a patch analysis framework was proposed in this paper for summarizing patch patterns, security patches identification, and their corresponding vulnerability patterns.

can be useful for the success of software from a business perspective. Table 3 shows the summary of the efficiently used techniques for evaluating the security of software systems.

4. Conclusion

Components of software play an important role in the functionality of the activities of software systems. Components are considered to be reused due to the properties that are already tested, debugged, and experienced in practice. The security of components is important for its nature due to avoidance of happening of illegal or malicious activities that can harm the success of the software system. The security of component can be high if it has a higher level of security. Security of software components can save the software from the harm of illegal access and damages of its contents. Diverse approaches are available to tackle the issues of security of components from diverse perceptions. A detailed report of the existing approaches and techniques used for security purposes is needed through which the researchers should know the in-depth knowledge of approaches, tools, and techniques. The proposed research presents an SLR of the approaches used by practitioners to protect software systems for IoT. The study has searched the literature in the popular and well-known libraries, filters the relevant literature, organizes the filter papers, and extracts derivations from the selected studies based on different perspectives. The proposed research will help practitioners and researchers in presenting new algorithms, techniques, and solutions for efficient assessment of the software components from security perspectives.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. Nazir, S. Shahzad, M. Nazir, and H. U. Rehman, "Evaluating security of software components using analytic network process," in *Proceedings of the 11th International Conference on Frontiers of Information Technology (FIT)*, pp. 183–188, IEEE, Islamabad, Pakistan, December 2013.
- [2] P. S. Sandhu and H. Singh, "A neuro-fuzzy based software reusability evaluation system with optimized rule selection," in *Proceedings of the 2006 International Conference on Emerging Technologies*, pp. 664–669, Peshawar, Pakistan, November 2006.
- [3] B. Liao, Y. Ali, S. Nazir, L. He, and H. U. Khan, "Security analysis of IoT devices by using mobile computing: a systematic literature review," *IEEE Access*, vol. 8, pp. 120331–120350, 2020.
- [4] M. Li, S. Nazir, H. U. Khan, S. Shahzad, and R. Amin, "Modelling features-based birthmarks for security of end-to-end communication system," *Security and Communication Networks*, vol. 2020, Article ID 8852124, 9 pages, 2020.
- [5] H. U. Rahman, A. U. Rehman, S. Nazir, I. U. Rehman, and N. Uddin, "Privacy and security—limits of personal information to minimize loss of privacy," in *Proceedings of the Future of Information and Communication Conference*, pp. 964–974, 2019.
- [6] S. Nazir, S. Shahzad, S. Mahfooz, and M. N. Jan, "Fuzzy logic based decision support system for component security evaluation," *International Arab Journal of Information and Technology*, vol. 15, pp. 1–9, 2015.
- [7] A. Rawashdeh and B. Matalkah, "A new software quality model for evaluating COTS components," *Journal of Computer Science*, vol. 2, no. 4, pp. 373–381, 2006.
- [8] H. H. Song, "Testing and evaluation system for cloud computing information security products," in *Proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019)*, pp. 84–87, Kunming, China, May 2019.
- [9] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, "Survey on IoT security: challenges and solution using machine learning, artificial intelligence and blockchain technology," *Internet of Things*, vol. 11, 2020.
- [10] R. Diesch, M. Pfaff, and H. Krcmar, "A comprehensive model of information security factors for decision-makers," *Computers & Security*, vol. 92, p. 101747, 2020.
- [11] N. A. B. Mohd and Z. F. Zaaba, "A review of usability and security evaluation model of ecommerce website," in *Proceedings of the Fifth Information Systems International Conference 2019*, pp. 1199–1205, Surabaya, Indonesia, July 2019.
- [12] Z. Katzir and Y. Elovici, "Quantifying the resilience of machine learning classifiers used for cyber security," *Expert Systems with Applications*, vol. 92, pp. 419–429, 2018.
- [13] S. Alam, M. M. R. Chowdhury, and J. Noll, "Interoperability of security-enabled internet of things," *Wireless Personal Communications*, vol. 61, no. 3, pp. 567–586, 2011.
- [14] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: the case study of a smart home," in *Proceedings of the 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pp. 618–623, Kona, HI, USA, March 2017.
- [15] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: current status, challenges and prospective measures," in *Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 336–341, London, UK, December 2015.
- [16] A. W. Atamli and A. Martin, "Threat-based security analysis for internet of things," in *Proceedings of the 2014 International Workshop on Secure Internet of Things*, pp. 35–43, Wroclaw, Poland, September 2014.
- [17] K. C. Park and D.-H. Shin, "Security assessment framework for IoT service," *Telecommunication Systems*, vol. 64, no. 1, pp. 193–209, 2017.
- [18] A. Tekeoglu and A. Ş. Tosun, "An experimental framework for investigating security and privacy of IoT devices," in *Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pp. 63–83, Vancouver, Canada, November 2017.
- [19] O. Mazhelis and P. Tyrväinen, "A framework for evaluating Internet-of-Things platforms: application provider viewpoint," in *Proceedings of the 2014 IEEE World Forum on*

- Internet of Things (WF-IoT)*, pp. 147–152, Seoul, South Korea, March 2014.
- [20] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, “Performance analysis of machine learning algorithms in intrusion detection system: a review,” *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.
- [21] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, “Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city,” *Future Generation Computer Systems*, vol. 107, pp. 433–442, 2020.
- [22] S. Manjiatahsien, Hadiskarimipour, and Petrosspachos, “Machine learning based solutions for security of Internet of Things (IoT): a survey,” *Journal of Network and Computer Applications*, vol. 161, 2020.
- [23] X. Wang, J. Li, X. Kuang, Y.-A. Tan, and J. Li, “The security of machine learning in an adversarial setting: a survey,” *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.
- [24] M. Marwan, A. Kartit, and H. Ouahmane, “Security enhancement in healthcare cloud using machine learning,” *Procedia Computer Science*, vol. 127, pp. 388–397, 2018.
- [25] M. Belouch, S. El Hadaj, and M. Idhammad, “Performance evaluation of intrusion detection based on machine learning using Apache Spark,” *Procedia Computer Science*, vol. 127, pp. 1–6, 2018.
- [26] C. Hosmer, “IoT vulnerabilities,” in *Defending IoT Infrastructures with the Raspberry Pi: Monitoring and Detecting Nefarious Behavior in Real Time*, pp. 1–15, Apress, Berkeley, CA, USA, 2018.
- [27] W. Mao, Z. Cai, D. Towsley, Q. Feng, and X. Guan, “Security importance assessment for system objects and malware detection,” *Computers & Security*, vol. 68, pp. 47–68, 2017.
- [28] T. Halabi and M. Bellaiche, “Towards quantification and evaluation of security of cloud service providers,” *Journal of Information Security and Applications*, vol. 33, pp. 55–65, 2017.
- [29] M. Cheah, S. A. Shaikh, O. Haas, and A. Ruddle, “Towards a systematic security evaluation of the automotive bluetooth interface,” *Vehicular Communications*, vol. 9, pp. 8–18, 2017.
- [30] I. Sidenko, “Multi-Criteria selection of the wireless communication technology for specialized IoT network,” in *Proceedings of the CEUR Workshop*, Rome, Italy, November 2014.
- [31] B. Uslu, T. Eren, Ş. Gür, and E. Özcan, “Evaluation of the difficulties in the internet of things (IoT) with multi-criteria decision-making,” *Processes*, vol. 7, no. 3, p. 164, 2019.
- [32] A. Hinduja and M. Pandey, “An ANP-GRA-based evaluation model for security features of IoT systems,” in *Advances in Intelligent Systems and Computing, Intelligent Communication, Control and Devices*, pp. 243–253, Springer, Berlin, Germany, 2020.
- [33] I. Cvitić and M. Vujić, “Classification of security risks in the IoT environment,” *Annals of DAAAM & Proceedings*, vol. 26, no. 1, 2015.
- [34] J. Chen, Y. Lu, H. Wang, and C. Mao, “A quantitative assessment approach to COTS component security,” *Mathematical Problems in Engineering*, vol. 2013, Article ID 165029, 11 pages, 2013.
- [35] J. I. Choi and K. R. B. Butler, “Secure multiparty computation and trusted hardware: examining adoption challenges and opportunities,” *Security and Communication Networks*, vol. 2019, Article ID 1368905, 28 pages, 2019.
- [36] H. Zhang, Z. Cai, Q. Liu, Q. Xiao, Y. Li, and C. F. Cheang, “A survey on security-aware measurement in SDN,” *Security and Communication Networks*, vol. 2018, Article ID 2459154, 14 pages, 2018.
- [37] X. Su, Z. Wang, X. Liu, C. Choi, and D. Choi, “Study to improve security for IoT smart device controller: drawbacks and countermeasures,” *Security and Communication Networks*, vol. 2018, Article ID 4296934, 14 pages, 2018.
- [38] S. Zhang, X. Ou, and J. Homer, “Effective network vulnerability assessment through model abstraction,” in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 17–34, Amsterdam, Netherlands, July 2011.
- [39] P. Zech, M. Felderer, and R. Breu, “Knowledge-based security testing of web applications by logic programming,” *International Journal on Software Tools for Technology Transfer*, vol. 21, no. 2, pp. 221–246, 2019.
- [40] C. Wijayarathna and N. A. G. Arachchilage, “Using cognitive dimensions to evaluate the usability of security APIs: an empirical investigation,” *Information and Software Technology*, vol. 115, pp. 5–19, 2019.
- [41] R. Leszczyna, “Standards on cyber security assessment of smart grid,” *International Journal of Critical Infrastructure Protection*, vol. 22, pp. 70–89, 2018.
- [42] O.-M. Latvala, J. Toivonen, A. Evesti, M. Sihvonen, and V. Jordan, “Security risk visualization with semantic risk model,” *Procedia Computer Science*, vol. 83, pp. 1194–1199, 2016.
- [43] F. Munodawafa and A. I. Awad, “Security risk assessment within hybrid data centers: a case study of delay sensitive applications,” *Journal of Information Security and Applications*, vol. 43, pp. 61–72, 2018.
- [44] N. Thompson, T. J. McGill, and X. Wang, ““Security begins at home”: determinants of home computer and mobile device security behavior,” *Computers & Security*, vol. 70, pp. 376–391, 2017.
- [45] B. Robisson, M. Agoyan, P. Soquet et al., “Smart security management in secure devices,” *Journal of Cryptographic Engineering*, vol. 7, no. 1, pp. 47–61, 2017.
- [46] F. Martín, E. Soriano, and J. M. Cañas, “Quantitative analysis of security in distributed robotic frameworks,” *Robotics and Autonomous Systems*, vol. 100, pp. 95–107, 2018.
- [47] S. Ismail, E. Sitnikova, and J. Slay, “Towards developing scada systems security measures for critical infrastructures against cyber-terrorist attacks,” in *Proceedings of the IFIP International Information Security Conference*, pp. 242–249, Marrakech, Morocco, June 2014.
- [48] S. Wu, Y. Zhang, and W. Cao, “Network security assessment using a semantic reasoning and graph based approach,” *Computers & Electrical Engineering*, vol. 64, pp. 96–109, 2017.
- [49] J. Almasizadeh and M. Abdollahi Azgomi, “Mean privacy: a metric for security of computer systems,” *Computer Communications*, vol. 52, pp. 47–59, 2014.
- [50] M. Abadi, “Software security: a formal perspective,” in *Proceedings of the International Symposium on Formal Methods*, pp. 1–5, Paris, France, August 2012.
- [51] L. Krautsevich, F. Martinelli, and A. Yautsiukhin, “Formal analysis of security metrics and risk,” in *Proceedings of the IFIP International Workshop on Information Security Theory and Practices*, pp. 304–319, Crete, Greece, June 2011.
- [52] I. Kotenko and E. Novikova, “Vissecanalyzer: a visual analytics tool for network security assessment,” in *Proceedings of the International Conference on Availability, Reliability, and*

- Security*, pp. 345–360, Regensburg, Germany, September 2013.
- [53] H.-K. Kim, W.-H. So, and S.-M. Je, “A big data framework for network security of small and medium enterprises for future computing,” *The Journal of Supercomputing*, vol. 75, no. 6, pp. 3334–3367, 2019.
- [54] T. Ibrahim, S. M. Furnell, M. Papadaki, and N. L. Clarke, “Assessing the usability of end-user security software,” in *Proceedings of the International Conference on Trust, Privacy and Security in Digital Business*, pp. 177–189, Bilbao, Spain, August 2010.
- [55] S. H. Houmb, I. Ray, I. Ray, and S. Chakraborty, “Trust-based security level evaluation using Bayesian belief networks,” in *Transactions on Computational Science X*, pp. 154–186, Springer, Berlin, Germany, 2010.
- [56] I. Garitano, S. Fayyad, and J. Noll, “Multi-metrics approach for security, privacy and dependability in embedded systems,” *Wireless Personal Communications*, vol. 81, no. 4, pp. 1359–1376, 2015.
- [57] J.-B. Gao, B.-W. Zhang, X.-H. Chen, and Z. Luo, “Ontology-based model of network and computer attacks for security assessment,” *Journal of Shanghai Jiaotong University (Science)*, vol. 18, no. 5, pp. 554–562, 2013.
- [58] G. Elahi, E. Yu, and N. Zannone, “A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities,” *Requirements Engineering*, vol. 15, no. 1, pp. 41–62, 2010.
- [59] M. L. Hale, K. Lotfy, R. F. Gamble, C. Walter, and J. Lin, “Developing a platform to evaluate and assess the security of wearable devices,” *Digital Communications and Networks*, vol. 5, no. 3, pp. 147–159, 2019.
- [60] S. Crane and S. Pearson, “Security/trustworthiness assessment of platforms,” in *Digital Privacy*, pp. 457–483, Springer, Berlin, Germany, 2011.
- [61] M. Compastié, R. M. Badonnel, O. Festor, R. He, and M. Kassi-Lahlou, “Towards a software-defined security framework for supporting distributed cloud,” in *Proceedings of the IFIP International Conference on Autonomous Infrastructure, Management and Security*, pp. 47–61, Zurich, Switzerland, July 2017.
- [62] M. Chen, Y. Qian, S. Mao, W. Tang, and X. Yang, “Software-defined mobile networks security,” *Mobile Networks and Applications*, vol. 21, no. 5, pp. 729–743, 2016.
- [63] O. Kussul, N. Kussul, and S. Skakun, “Assessing security threat scenarios for utility-based reputation model in grids,” *Computers & Security*, vol. 34, pp. 1–15, 2013.
- [64] V. Casola, A. De Benedictis, A. Riccio, D. Rivera, W. Mallouli, and E. M. De Oca, “A security monitoring system for internet of things,” *Internet of Things*, vol. 7, p. 100080, 2019.
- [65] A. V. Uzunov, E. B. Fernandez, and K. Falkner, “ASE: a comprehensive pattern-driven security methodology for distributed systems,” *Computer Standards & Interfaces*, vol. 41, pp. 112–137, 2015.
- [66] A. M. Hoole, I. Traore, and I. Simplot-Ryl, “Application of contract-based security assertion monitoring framework for telecommunications software engineering,” *Mathematical and Computer Modelling*, vol. 53, no. 3–4, pp. 522–537, 2011.
- [67] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “An evaluation framework for network security visualizations,” *Computers & Security*, vol. 84, pp. 70–92, 2019.
- [68] H. Mumtaz, M. Alshayeb, S. Mahmood, and M. Niazi, “An empirical study to improve software security through the application of code refactoring,” *Information and Software Technology*, vol. 96, pp. 112–125, 2018.
- [69] A. K. Srivastava and S. Kumar, “An effective computational technique for taxonomic position of security vulnerability in software development,” *Journal of Computational Science*, vol. 25, pp. 388–396, 2018.
- [70] G. Spanos and L. Angelis, “A multi-target approach to estimate software vulnerability characteristics and severity scores,” *Journal of Systems and Software*, vol. 146, pp. 152–166, 2018.
- [71] M. Jouini, L. B. A. Rabai, and R. Khedri, “A multidimensional approach towards a quantitative assessment of security threats,” *Procedia Computer Science*, vol. 52, pp. 507–514, 2015.
- [72] W. Wang, K. R. Mahakala, A. Gupta, N. Hussein, and Y. Wang, “A linear classifier based approach for identifying security requirements in open source software development,” *Journal of Industrial Information Integration*, vol. 14, pp. 34–40, 2019.
- [73] A. Alebrahim and M. Heisel, “Towards developing secure software using problem-oriented security patterns,” in *Proceedings of the International Conference on Availability, Reliability, and Security*, pp. 45–62, Fribourg, Switzerland, September 2014.
- [74] J. Bürger, D. Strüber, S. Gärtner, T. Ruhroth, J. Jürjens, and K. Schneider, “A framework for semi-automated co-evolution of security knowledge and system models,” *Journal of Systems and Software*, vol. 139, pp. 142–160, 2018.
- [75] J. Davies, “Measuring subversions: security and legal risk in reused software artifacts,” in *Proceedings of the 33rd International Conference on Software Engineering*, pp. 1149–1151, Honolulu, HI, USA, April 2011.
- [76] D. Macdonald, S. L. Clements, S. W. Patrick et al., “Cyber/physical security vulnerability assessment integration,” in *Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–6, Washington, DC, USA, February 2013.
- [77] M. S. Kozłowski, “Cloud security monitoring and vulnerability management,” in *Critical Infrastructure Protection Research*, pp. 123–139, Springer, Berlin, Germany, 2016.
- [78] S. Panasenko, “Evaluation of distributed security systems server modules peak workload,” in *Proceedings of the 2013 International Conference on Anti-Counterfeiting, Security and Identification (ASID)*, pp. 1–4, Shanghai, China, October 2013.
- [79] M. R. Razian and H. M. Sangchi, “A threatened-based software security evaluation method,” in *Proceedings of the 2014 11th International ISC Conference on Information Security and Cryptology*, pp. 120–125, Tehran, Iran, September 2014.
- [80] P. Morrison, “A security practices evaluation framework,” in *Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, pp. 935–938, Florence, Italy, May 2015.
- [81] J. Zhang, Q. Wu, R. Zheng, J. Zhu, M. Zhang, and R. Liu, “A security monitoring method based on autonomic computing for the cloud platform,” *Journal of Electrical and Computer Engineering*, vol. 2018, Article ID 8309450, 9 pages, 2018.
- [82] B. Bordel, R. Alcarria, D. M. De Andres, and I. You, “Securing Internet-of-Things systems through implicit and explicit reputation models,” *IEEE Access*, vol. 6, pp. 47472–47488, 2018.
- [83] F. Hategekimana, J. M. Mbongue, M. J. H. Pantho, and C. Bobda, “Inheriting software security policies within

- hardware IP components,” in *Proceedings of the 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 53–56, Boulder, CO, USA, April 2018.
- [84] I. Alsmadi and F. Mira, “Website security analysis: variation of detection methods and decisions,” in *Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1–5, Riyadh, Saudi Arabia, April 2018.
- [85] A. R. S. Farhan and G. M. M. Mostafa, “A methodology for enhancing software security during development processes,” in *Proceedings of the 2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1–6, Riyadh, Saudi Arabia, April 2018.
- [86] A. Wanniarachchi and C. Gamage, “RECSRf: novel technique to evaluate program security using dynamic disassembly of machine instructions,” in *Proceedings of the 2019 21st International Conference on Advanced Communication Technology (ICACT)*, pp. 545–551, PyeongChang, South Korea, February 2019.
- [87] B. Cruz, S. Gómez-Meire, D. Ruano-Ordás, H. Janicke, I. Yevseyeva, and J. R. Méndez, “A practical approach to protect IoT devices against attacks and compile security incident datasets,” *Scientific Programming*, vol. 2019, Article ID 9067512, 11 pages, 2019.
- [88] M. Saito, A. Hazeyama, N. Yoshioka et al., “A case-based management system for secure software development using software security knowledge,” *Procedia Computer Science*, vol. 60, pp. 1092–1100, 2015.
- [89] M. Ahmad, V. Costamagna, B. Crispo, F. Bergadano, and Y. Zhauniarovich, “StaDART: addressing the problem of dynamic code updates in the security analysis of android applications,” *Journal of Systems and Software*, vol. 159, p. 110386, 2020.
- [90] S. M. Ghaffarian and H. R. Shahriari, “Software vulnerability analysis and discovery using machine-learning and data-mining techniques: a survey,” *ACM Computing Surveys*, vol. 50, no. 4, pp. 1–36, 2017.
- [91] N. Schagen, K. Koning, H. Bos, and C. Giuffrida, “Towards automated vulnerability scanning of network servers,” in *Proceedings of the 11th European Workshop on Systems Security*, pp. 1–6, Porto, Portugal, April 2018.
- [92] G. Spanos, L. Angelis, and D. Toloudis, “Assessment of vulnerability severity using text mining,” in *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, pp. 1–6, Larissa, Greece, September 2017.
- [93] P. Rotella, “Software security vulnerabilities: baselining and benchmarking,” in *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment*, pp. 3–10, Gothenburg, Sweden, June 2018.
- [94] C. Fruehwirth, S. Biffel, A. Schatten, D. Winkler, and W. D. Sunindyo, “Quantitative software security measurement in an engineering service bus platform,” in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 1, Bozen, Italy, September 2010.
- [95] L. Allodi, S. Banescu, H. Femmer, and K. Beckers, “Identifying relevant information cues for vulnerability assessment using CVSS,” in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, pp. 119–126, Tempe, AZ, USA, March 2018.
- [96] N. H. Pham, T. T. Nguyen, H. A. Nguyen, X. Wang, A. T. Nguyen, and T. N. Nguyen, “Detecting recurring and similar software vulnerabilities,” in *Proceedings of the 2010 ACM/IEEE 32nd International Conference on Software Engineering*, pp. 227–230, Cape Town, South Africa, May 2010.
- [97] S. E. Sahin and A. Tosun, “A conceptual replication on predicting the severity of software vulnerabilities,” in *Proceedings of the Evaluation and Assessment on Software Engineering*, pp. 244–250, Copenhagen, Denmark, April 2019.
- [98] T. Casey, P. Koeberl, and C. Vishik, “Threat agents: a necessary component of threat analysis,” in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, pp. 1–4, Oak Ridge, TN, USA, April 2010.
- [99] G. Grieco, G. L. Grinblat, L. Uzal, S. Rawat, J. Feist, and L. Mounier, “Toward large-scale vulnerability discovery using machine learning,” in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 85–96, New Orleans, LA, USA, March 2016.
- [100] Y. Lu, K. Mitropoulos, R. Ostrovsky, A. Weinstock, and V. Zikas, “Cryptographically secure detection of injection attacks,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2240–2242, Toronto, Canada, October 2018.
- [101] S. Hameed, F. I. Khan, and B. Hameed, “Understanding security requirements and challenges in Internet of Things (IoT): a review,” *Journal of Computer Networks and Communications*, vol. 2019, no. 11, pp. 1–14, 2019.
- [102] A. Evesti and E. Ovaska, “Comparison of adaptive information security approaches,” *International Scholarly Research Notices*, vol. 2013, Article ID 482949, 18 pages, 2013.
- [103] W. Shang, T. Gong, C. Chen, J. Hou, and P. Zeng, “Information security risk assessment method for ship control system based on fuzzy sets and attack trees,” *Security and Communication Networks*, vol. 2019, Article ID 3574675, 11 pages, 2019.
- [104] J. Xu and D. Feng, “Identification of ICS security risks toward the analysis of packet interaction characteristics using state sequence matching based on SF-FSM,” *Security and Communication Networks*, vol. 2017, Article ID 2430835, 17 pages, 2017.
- [105] G. Wangen, C. Hallstensen, and E. Snekkenes, “A framework for estimating information security risk assessment method completeness,” *International Journal of Information Security*, vol. 17, no. 6, pp. 681–699, 2018.
- [106] S. Moshtari, A. Sami, and M. Azimi, “Using complexity metrics to improve software security,” *Computer Fraud & Security*, vol. 2013, no. 5, pp. 8–17, 2013.
- [107] M. A. V. Staalduinen, F. Khan, and V. Gadag, “SVAPP methodology: a predictive security vulnerability assessment modeling method,” *Journal of Loss Prevention in the Process Industries*, vol. 43, pp. 397–413, 2016.
- [108] H. Suleiman and D. Svetinovic, “Evaluating the effectiveness of the security quality requirements engineering (SQUARE) method: a case study using smart grid advanced metering infrastructure,” *Requirements Engineering*, vol. 18, no. 3, pp. 251–279, 2013.
- [109] Y. Kim, J. Nam, T. Park, S. Scott-Hayward, and S. Shin, “SODA: a software-defined security framework for IoT environments,” *Computer Networks*, vol. 163, p. 106889, 2019.
- [110] H. Maziku, S. Shetty, and D. M. Nicol, “Security risk assessment for SDN-enabled smart grids,” *Computer Communications*, vol. 133, pp. 1–11, 2019.
- [111] W. Yang, J. Li, Y. Zhang, and D. Gu, “Security analysis of third-party in-app payment in mobile applications,” *Journal of Information Security and Applications*, vol. 48, p. 102358, 2019.

- [112] S. Banescu, M. Ochoa, and A. Pretschner, "A framework for measuring software obfuscation resilience against automated attacks," in *Proceedings of the 2015 IEEE/ACM 1st International Workshop on Software Protection*, pp. 45–51, Florence, Italy, May 2015.
- [113] B. Chernis and R. Verma, "Machine learning methods for software vulnerability detection," in *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, pp. 31–39, Tempe, AZ, USA, March 2018.
- [114] S. Ouchani and M. Debbabi, "Specification, verification, and quantification of security in model-based systems," *Computing*, vol. 97, no. 7, pp. 691–711, 2015.
- [115] M. Elsayed and M. Zulkernine, "Offering security diagnosis as a service for cloud SaaS applications," *Journal of Information Security and Applications*, vol. 44, pp. 32–48, 2019.
- [116] C. Bodei, S. Chessa, and L. Galletta, "Measuring security in IoT communications," *Theoretical Computer Science*, vol. 764, pp. 100–124, 2019.
- [117] M. Ge, J. B. Hong, W. Guttman, and D. S. Kim, "A framework for automating security analysis of the internet of things," *Journal of Network and Computer Applications*, vol. 83, pp. 12–27, 2017.
- [118] J. H. Lee and S. J. Kim, "Analysis and security evaluation of security threat on broadcasting service," *Wireless Personal Communications*, vol. 95, no. 4, pp. 4149–4169, 2017.
- [119] J. Jürjens, K. Schneider, J. Bürger et al., "Maintaining security in software evolution," in *Managed Software Evolution*, pp. 207–253, Springer, Cham, Switzerland, 2019.
- [120] D. Gupta, K. Chatterjee, and S. Jaiswal, "A framework for security testing," in *Proceedings of the International Conference on Computational Science and Its Applications*, pp. 187–198, Cagliari, Italy, July 2013.
- [121] A. Goldstein and U. Frank, "Components of a multi-perspective modeling method for designing and managing IT security systems," *Information Systems and E-Business Management*, vol. 14, no. 1, pp. 101–140, 2016.
- [122] V. Desnitsky and I. Kotenko, "Expert knowledge based design and verification of secure systems with embedded devices," in *Proceedings of the International Conference on Availability, Reliability, and Security*, pp. 194–210, Fribourg, Switzerland, September 2014.
- [123] M. Cheah, S. A. Shaikh, J. Bryans, and P. Wooderson, "Building an automotive security assurance case using systematic security evaluations," *Computers & Security*, vol. 77, pp. 360–379, 2018.
- [124] K. Beckers, I. Côté, S. Faßbender, M. Heisel, and S. Hofbauer, "A pattern-based method for establishing a cloud-specific information security management system," *Requirements Engineering*, vol. 18, no. 4, pp. 343–395, 2013.
- [125] S. Y. Enoch, M. Ge, J. B. Hong, H. Alzaid, and D. S. Kim, "A systematic evaluation of cybersecurity metrics for dynamic networks," *Computer Networks*, vol. 144, pp. 216–229, 2018.
- [126] J. Almasizadeh and M. A. Azgomi, "A stochastic model of attack process for the evaluation of security metrics," *Computer Networks*, vol. 57, no. 10, pp. 2159–2180, 2013.
- [127] R. A. Oliveira, N. Laranjeiro, and M. Vieira, "Assessing the security of web service frameworks against Denial of service attacks," *Journal of Systems and Software*, vol. 109, pp. 18–31, 2015.
- [128] A. R. Sai, J. Buckley, and A. Le Gear, "Assessing the security implication of Bitcoin exchange rates," *Computers & Security*, vol. 86, pp. 206–222, 2019.
- [129] M. Hussain, A. Al-Haiqi, A. A. Zaidan et al., "A security framework for mHealth apps on Android platform," *Computers & Security*, vol. 75, pp. 191–217, 2018.
- [130] G. Sciarretta, R. Carbone, S. Ranise, and A. Armando, "Anatomy of the facebook solution for mobile single sign-on: security assessment and improvements," *Computers & Security*, vol. 71, pp. 71–86, 2017.
- [131] M. S. Ahmed, E. Al-Shaer, M. Taibah, and L. Khan, "Objective risk evaluation for automated security management," *Journal of Network and Systems Management*, vol. 19, no. 3, pp. 343–366, 2011.
- [132] Z. Yan and C. Prehofer, "Autonomic trust management for a component-based software system," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 810–823, 2010.
- [133] H. H. Albreiki and Q. H. Mahmoud, "Evaluation of static analysis tools for software security," in *Proceedings of the 2014 10th International Conference on Innovations in Information Technology (IIIT)*, pp. 93–98, Al Ain, UAE, November 2014.
- [134] M. L. Hale, D. Ellis, R. Gamble, C. Waler, and J. Lin, "Secu Wear: an open source, multi-component hardware/software platform for exploring wearable security," in *Proceedings of the 2015 IEEE International Conference on Mobile Services*, pp. 97–104, New York, NY, USA, June 2015.
- [135] R. Al-Jaljouli, J. Abawajy, M. M. Hassan, and A. Alelaiwi, "Secure multi-attribute one-to-many bilateral negotiation framework for e-commerce," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 415–429, 2016.
- [136] B. Potteiger, W. Emfinger, H. Neema, X. Koutosukos, C. Tang, and K. Stouffer, "Evaluating the effects of cyber-attacks on cyber physical systems using a hardware-in-the-loop simulation testbed," in *Proceedings of the 2017 Resilience Week (RWS)*, pp. 177–183, Wilmington, DE, USA, September 2017.
- [137] K.-O. Detken, M. Jahnke, T. Rix, and A. Rein, "Software-design for internal security checks with dynamic integrity measurement (DIM)," in *Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, pp. 367–373, September 2017, Bucharest, Romania.
- [138] D. Migault, M. A. Simplicio, B. M. Barros et al., "A framework for enabling security services collaboration across multiple domains," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 999–1010, Atlanta, GA, USA, June 2017.
- [139] R. Bloomfield, K. Netkachova, and R. Stroud, "Security-informed safety: if it's not secure, it's not safe," in *Proceedings of the International Workshop on Software Engineering for Resilient Systems*, pp. 17–32, Kiev, Ukraine, October 2013.
- [140] D. W. Chadwick, W. Fan, G. Costantino et al., "A cloud-edge based data security architecture for sharing and analysing cyber threat information," *Future Generation Computer Systems*, vol. 102, pp. 710–722, 2020.
- [141] P. Doty, "U.S. homeland security and risk assessment," *Government Information Quarterly*, vol. 32, no. 3, pp. 342–352, 2015.
- [142] N. H. Pham, T. T. Nguyen, H. A. Nguyen, and T. N. Nguyen, "Detection of recurring software vulnerabilities," in *Proceedings of the IEEE/ACM International Conference on Automated Software Engineering*, pp. 447–456, Antwerp, Belgium, September 2010.
- [143] K. Oishi and T. Matsumoto, "Self destructive tamper response for software protection," in *Proceedings of the 6th ACM Symposium on Information, Computer and*

Communications Security, pp. 490–496, Hong Kong, China, March 2011.

- [144] S. Zhang, X. Meng, L. Wang, L. Xu, and X. Han, “Secure virtualization environment based on advanced memory introspection,” *Security and Communication Networks*, vol. 2018, Article ID 9410278, 16 pages, 2018.
- [145] H. Jo, J. Nam, and S. Shin, “NOSArmor: building a secure network operating system,” *Security and Communication Networks*, vol. 2018, Article ID 9178425, 14 pages, 2018.
- [146] Z. Xu, B. Chen, M. Chandramohan, Y. Liu, and F. Song, “Spain: security patch analysis for binaries towards understanding the pain and pills,” in *Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pp. 462–472, Buenos Aires, Argentina, May 2017.

Research Article

Private Predicate Encryption for Inner Product from Key-Homomorphic Pseudorandom Function

Yi-Fan Tseng, Zi-Yuan Liu , Jen-Chieh Hsu, and Raylin Tso

Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan

Correspondence should be addressed to Zi-Yuan Liu; zyliu@cs.nccu.edu.tw

Received 4 November 2020; Revised 7 December 2020; Accepted 21 January 2021; Published 12 February 2021

Academic Editor: David Megías

Copyright © 2021 Yi-Fan Tseng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Predicate encryption (PE), formalized by Katz et al., is a new paradigm of public-key encryption that conceptually captures the public-key encryption that supports fine-grained access control policy. Because of the nature of PE, it is used for cloud storage so that users can retrieve encrypted data without revealing any information about the data to cloud servers and other users. Although lots of PE schemes have been studied, the predicate-hiding security is seldom considered; that is, the user's secret key may leak sensitive information of the predicate. Additionally, the security of the current predicate-hiding PE schemes relies on the discrete logarithm assumption which cannot resist the quantum attacks in the future. In this paper, we propose a generic PE for inner product under symmetric-key setting, called private IPE, from specific key-homomorphic pseudorandom function (PRF). The rigorous proofs are provided to show that the construction is payload-hiding, attribute-hiding, and predicate-hiding secure. With the advantage of the generic construction, if the underlying PRF can resist quantum attacks, then, through our proposed generic construction, a quantum-resistant private IPE can be obtained.

1. Introduction

In recent years, cloud computing has become increasingly important as smartphones and Internet of Things devices are widely used in our life. Users typically upload their data to the cloud to achieve efficient computing and reduce storage requirements of their devices. Due to the fact that the uploaded data are sensitive, users may consider using authentication protocol [1–4] and encryption schemes [5, 6] to protect their data privacy in cloud environment. One novel approach is to encrypt data before it is uploaded to the cloud. However, encrypted data loses flexibility in data usage, such as fine-grained control over access to encrypted data. For example, a user may want to search for and download ciphertext that corresponds to certain attributes. If each piece of data is purely encrypted, the only way is to download all the ciphertexts and decrypt them for search. Unfortunately, this approach would be very inefficient. Therefore, how to efficiently control the access to encrypted data and ensure the privacy and security of data is an urgent issue for cloud computing.

Predicate encryption (PE) [7], formalized by Katz et al., is a general paradigm that conceptually captures the public-key encryption supporting fine-grained access control policy. In a PE scheme for a predicate function P , a secret key, issued by a trusted authority, is associated with a key attribute y , while the ciphertext is associated with a ciphertext attribute x . Specifically, the ciphertext can be decrypted using the secret key if and only if $P(x, y) = 1$. Therefore, PE can be used as access control mechanism for the previous cloud storage scenario and provide the flexibility for encryption schemes, which allows sender to encrypt data with more complicated access policy. For example, in a school scenario, the secret keys of each teacher and each student are associated with key attributes “teacher” and “student,” respectively. If the principal wants to encrypt a file that can only be decrypted by each student and teacher, he/she can use a PE supporting “belong to” functionality and encrypt this file with a ciphertext attribute “student or teacher.” Because the key attributes “teacher” and “student” belong to ciphertext

attribute “student or teacher,” the secret keys associated with these key attributes can decrypt the ciphertext.

Additionally, Katz et al. proposed the first PE supporting inner product predicate, called PE for inner product (IPE), whereas ciphertext can be decrypted if and only if the inner product of \mathbf{x} and \mathbf{y} is equal to 0. They further suggested that IPE can be used to build other more flexible schemes, such as (anonymous) identity-based encryption [8], hidden vector encryption [9, 10], CNF/DNF formulas [7], PE schemes supporting polynomial evaluation [11], and exact thresholds [12]. The most basic security requirement of IPE, called payload-hiding, stipulates that a ciphertext does not reveal any information of the plaintext if $P(\mathbf{x}, \mathbf{y}) = 1$. A stronger security requirement of PE is attribute-hiding, which stipulates that a ciphertext reveals nothing about the ciphertext attribute. Although a lot of attribute-hiding IPE schemes [13–16] have been studied, seldom schemes [17–19] focus on the predicate-hiding security. In more detail, a secret key may reveal some sensitive information of the predicate that belongs to the key holder. Actually, in public-key cryptosystem, since the encryption algorithm is publicly accessible, any user can adaptively generate a ciphertext. The user who has obtained a secret key can evaluate its predicate with possible ciphertexts; thus it is hard to achieve predicate-hiding in the public-key setting.

Shen et al. [18] first considered constructing the IPE under symmetric-key setting, a.k.a. private IPE, to achieve predicate-hiding security requirement. More precisely, in the work, when generating a secret key, generating a ciphertext requires a master secret key, so that not every user can adaptively generate a ciphertext to test which predicate is embedded in the secret key. Compared with IPE under public-key setting, private IPE is more suitable for cloud storage under self-use scenario. For example, as shown in Figure 1, Alice uses the cloud storage service to store her files. For privacy concern, she uses private IPE as an access control mechanism. Alice not only uploads an encrypted file $ct_{File,i}$ but also uploads another ciphertext $ct_{x,i} = \text{Encrypt}(SK, \mathbf{x}, M = 1)$ for a specific ciphertext attribute by using private IPE. When Alice wants to retrieve encrypted files, she can send the secret key for some key attribute, that is, $sk_y \leftarrow \text{KeyGen}(SK, \mathbf{y})$, to the cloud. The cloud can then evaluate the predicate on each ciphertext by performing decryption. If the predicate is satisfied, that is, $1 \stackrel{?}{=} \text{Decrypt}(ct_{x,i}, sk_y)$, the cloud returns the corresponding encrypted files of those ciphertexts.

After Shen et al.’s pioneering work [18], Yoshino et al. [19] provided a more practical IPE scheme that uses only three groups, whereas [18] required four groups. In addition, Kawai and Takashima [17] then introduced a predicate-hiding IPE, where the security is proven under the decision linear assumption without random oracles. However, the sizes of the secret keys of the above schemes [17–19] are linearly related to the lengths of the key attributes. Due to the fact that users may obtain many secret keys for decrypting different ciphertext, it is important to reduce the key size of secret key. In addition, Shor [20, 21] has shown that existing quantum algorithms can break the discrete logarithm and factoring assumptions. Therefore, the current private IPE

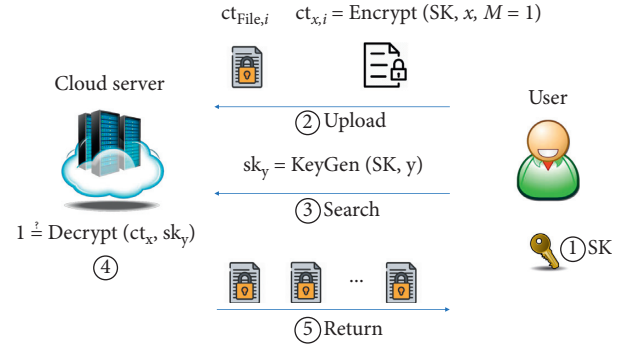


FIGURE 1: Private IPE scheme for cloud storage in self-use scenario.

schemes [17–19] are susceptible to quantum attack. Hence, how to construct a quantum-resistant private IPE scheme where the secret key is of constant size remains an open issue.

1.1. Our Contributions. In this paper, inspired by Alamedi et al.’s work [22], we propose a generic private IPE construction by utilizing specific key-homomorphic pseudo-random functions (PRF). By the advantage of the generic construction, the construction enjoys the security properties of the underlying primitives. Therefore, if the underlying key-homomorphic PRF is quantum-resistant, we further obtain a quantum-resistant private IPE scheme. In particular, in our construction, we require the underlying key-homomorphic PRF to have the following property for decryption correctness: the key space \mathcal{K} and the output space \mathcal{Y} are equal to \mathbb{Z}_q , for some prime q .

To obtain a private IPE scheme with constant-size secret key, we carefully use the key-homomorphic property of the key-homomorphic PRF to map each predicate attribute to the inner product of master secret key and secret key. That is, $sk_y = \sum_{i=1}^{\ell} (\sum_{j=1}^{y_i} F(a_i, h)) = F(\langle \mathbf{a}, \mathbf{y} \rangle, h)$, where $\mathbf{y} = (y_1, \dots, y_\ell)$ is a predicate vector and $(\mathbf{a} = (a_1, \dots, a_\ell), h)$ is the master secret key. Hence, the size of secret key is only $\log_2 q$, where q is the underlying modulo.

In addition, the rigorous security proofs are provided to demonstrate that if the underlying key-homomorphic PRF satisfies pseudorandomness (i.e., the output value of key-homomorphic PRF is indistinguishable from the value randomly chosen from \mathcal{Y}), the proposed construction satisfies the criteria of payload-hiding, attribute-hiding, and predicate-hiding privacy. The comparison of our construction with other state-of-the-art private IPE schemes is presented to show that our result is not only more secure but also more efficient with respect to the size of secret key.

In summary, this work introduces a generic construction to show how to obtain the first quantum-resistant private IPE scheme with a constant-size secret key.

1.2. Paper Organization. The rest of the paper is organized as follows. Section 2 recalls the definition of the PRF used in our generic construction. Moreover, Section 3 provides the definition and security requirement of the private PE. Next,

Sections 4 and Section 5 introduce and provide the security proofs of our generic construction, respectively. Section 6 compares our proposed construction with the related private IPE schemes. Finally, Section 7 concludes this study.

2. Pseudorandom Function (PRF)

In this section, we recall the definition of pseudorandom function from [23].

Definition 1 (pseudorandom functions [23]). A PRF $F: \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ is a keyed function defined over a key space \mathcal{K} , a domain \mathcal{X} , and a range \mathcal{Y} (these sets may be parameterized by the security parameter λ), whose output is indistinguishable from a truly random value. The security of a PRF can be defined by the two experiments $EXP(0)$ and $EXP(1)$ with an adversary \mathcal{A} . At first, a key k is uniformly randomly chosen from the key space \mathcal{K} . Given the description of the PRF, the adversary is then allowed to make queries to the following oracles:

- (i) *Evaluate*. Given $x \in \mathcal{X}$ from \mathcal{A} , the oracle returns $F(k, x)$ to \mathcal{A} .
- (ii) *Challenge*. Given $x \in \mathcal{X}$ from \mathcal{A} , where x has not been queried to evaluate Oracle, if $b = 1$, then the oracle returns $F(k, x)$, and if $b = 0$, then the oracle returns a random $y \xleftarrow{\$} \mathcal{Y}$.

Once the adversary is done querying the oracles, it outputs a bit $b' \in \{0, 1\}$. For $b = 0, 1$, we define W_b as the event where the adversary outputs $b' = 1$ in the experiment $EXP(b)$. The advantage of an adversary \mathcal{A} is defined as

$$\text{Adv}_{\mathcal{A}}^{\text{PRF}}(1^\lambda) = |\Pr[W_1] - \Pr[W_0]|. \quad (1)$$

We say that a PRF is secure if, for all PPT adversary \mathcal{A} , $\text{Adv}_{\mathcal{A}}^{\text{PRF}}(1^\lambda)$ is negligible.

Definition 2 (key-homomorphic PRF [24]). Let (\mathcal{K}, \circ) and $(\mathcal{Y}, *)$ be groups. Then, a keyed function $F: \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ is a key-homomorphic PRF:

- (i) F is a secure PRF.
- (ii) For every $k_1, k_2 \in \mathcal{K}$ and every input $x \in \mathcal{X}$, we have

$$F(k_1, x) * F(k_2, x) = F(k_1 \circ k_2, x). \quad (2)$$

Definition 3 (pseudorandom generators [24]). A pseudorandom generator (PRG) is an efficiently computable function $G: \mathcal{X} \rightarrow \mathcal{Y}$ with the following security, where (\mathcal{X}, \circ) and $(\mathcal{Y}, *)$ are groups. The security of a PRG is secure if, for any PPT algorithm \mathcal{A} , is negligible.

$$\text{Adv}_{\mathcal{A}}^{\text{PRG}}(1^\lambda) = |\Pr[\mathcal{A}(G(x)) = 1; x \leftarrow \mathcal{X}] - \Pr[\mathcal{A}(R) = 1; R \leftarrow \mathcal{Y}]|. \quad (3)$$

3. Private Predicate Encryption

Let $\{P = P_\ell\}_{\ell \in \mathbb{N}^c}$ for some constant $c \in \mathbb{N}$ be a predicate family, where $P_\ell: \mathfrak{A}_\ell \times \mathfrak{B}_\ell \rightarrow \{0, 1\}$ is a predicate function defined over a ciphertext attribute space \mathfrak{A}_ℓ and a key attribute space \mathfrak{B}_ℓ . The family index ℓ specifies the description of a predicate from the family. We would occasionally omit the index ℓ when the context is clear.

3.1. System Model. A private PE for predicate function $P: \mathfrak{A} \times \mathfrak{B} \rightarrow \{0, 1\}$ consists of four algorithms: *Setup*, *KeyGen*, *Encrypt*, and *Decrypt*. The details of the algorithms are shown as follows:

- (i) *Setup*($1^\lambda, 1^\ell$) $\rightarrow (pp, SK)$. Given the security parameters and the family index (λ, ℓ) , the algorithm outputs the system parameter pp and the secret key SK . Note that the description of \mathfrak{A} and \mathfrak{B} will be implicitly included in pp .
- (ii) *Encrypt*(pp, SK, \mathbf{x}, M) $\rightarrow ct_{\mathbf{x}}$. Given the system parameter pp , a secret key SK , a ciphertext attribute $\mathbf{x} \in \mathfrak{A}$, and a message M , the algorithm outputs a ciphertext $ct_{\mathbf{x}}$ for \mathbf{x} .
- (iii) *KeyGen*(pp, SK, \mathbf{y}) $\rightarrow sk_{\mathbf{y}}$. Given the system parameter pp , a secret key SK , and a key attribute $\mathbf{y} \in \mathfrak{B}$, the algorithm outputs the secret key $sk_{\mathbf{y}}$ for \mathbf{y} .
- (iv) *Decrypt*($pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}$) $\rightarrow (M/\perp)$. Given the system parameter pp , a ciphertext $ct_{\mathbf{x}}$, and a secret key $sk_{\mathbf{y}}$, the algorithm outputs a message M or an error symbol \perp .

Definition 4 (correctness). For all $\lambda, \ell \in \mathbb{N}$, $\mathbf{x} \in \mathfrak{A}$, and $\mathbf{y} \in \mathfrak{B}$, letting $ct_{\mathbf{x}} \leftarrow \text{Encrypt}(pp, SK, \mathbf{x}, M)$ and $sk_{\mathbf{y}} \leftarrow \text{KeyGen}(pp, SK, \mathbf{y})$, we have

$$\begin{aligned} M &\leftarrow \text{Decrypt}(pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}), \text{ if } P(\mathbf{x}, \mathbf{y}) = 1; \\ \perp &\leftarrow \text{Decrypt}(pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}), \text{ if } P(\mathbf{x}, \mathbf{y}) = 0, \end{aligned} \quad (4)$$

where $(pp, SK) \leftarrow \text{Setup}(1^\lambda, 1^\ell)$.

In this paper, we construct a private PE scheme supporting inner product predicate function defined over \mathbb{Z}_q^ℓ , where q is a large prime. That is,

- (i) ℓ denotes the dimension of the vector space.
- (ii) $\mathfrak{A} = \mathfrak{B} = \mathbb{Z}_p^\ell$.
- (iii) For all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^\ell$, $P: (\mathbf{x}, \mathbf{y}) = 1$ if $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.

Such encryption schemes are called ‘‘private PE for inner product’’ (private IPE), and $\mathfrak{A} = \mathbb{Z}_q^\ell$ and $\mathfrak{B} = \mathbb{Z}_q^\ell$ are called attribute vector space and predicate vector space, respectively.

3.2. Security Definitions. In private PE, there exist three types of adversary that want to retrieve the information of message, ciphertext attribute, and key attribute from ciphertext and secret key. Therefore, we model three security requirements of private PE, payload-hiding,

attribute-hiding, and predicate-hiding securities, to model the attacks from these adversaries.

The payload-hiding security [7] for predicate function $P: \mathfrak{A} \times \mathfrak{B} \rightarrow \{0, 1\}$ is defined as an interactive game between a challenger \mathcal{C} and an adversary \mathcal{A} . In payload-hiding models, a ciphertext reveals nothing about the encrypted message, and thus in some literature it is defined as IND-CPA security.

3.2.1. Payload-Hiding Game

- (i) *Setup*. The challenger \mathcal{C} runs $Setup(1^\lambda, 1^\ell)$ to generate a secret key SK and the system parameter pp . Then, it sends the system parameter pp to the adversary \mathcal{A} and keeps the secret key SK secretly.
- (ii) *Query Phase 1*. \mathcal{A} can query polynomially many times of the oracles described as follows:
 - (i) *KeyGen Oracle*: when \mathcal{A} issues a query with $\mathbf{y} \in \mathfrak{B}$, \mathcal{C} returns a secret key $sk_{\mathbf{y}} \leftarrow KeyGen(pp, SK, \mathbf{y})$.
 - (ii) *Encrypt Oracle*: when \mathcal{A} issues a query with $\mathbf{x} \in \mathfrak{A}$ and a message M , \mathcal{C} returns a ciphertext $ct_{\mathbf{x}} \leftarrow Encrypt(pp, SK, \mathbf{x}, M)$.
- (iii) *Challenge*. The adversary \mathcal{A} submits $\mathbf{x}^* \in \mathfrak{A}$ such that $P(\mathbf{x}^*, \mathbf{y}) = 0$ for all $\mathbf{y} \in \mathfrak{B}$, which has been queried to KeyGen Oracle in Query Phase 1, and two messages M_0, M_1 with the same length to the challenger \mathcal{C} . Then \mathcal{C} randomly chooses $b \in \{0, 1\}$ and returns a challenge ciphertext $c_{\mathbf{x}^*} \leftarrow Encrypt(pp, SK, \mathbf{x}^*, M_b)$.
- (iv) *Query Phase 2*. This phase is the same as Query Phase 1, except that \mathcal{A} is only allowed to make a query to KeyGen Oracle with $\mathbf{y} \in \mathfrak{B}$ such that $P(\mathbf{x}^*, \mathbf{y}) = 0$.
- (v) *Guess*. The adversary \mathcal{A} outputs a bit b' and wins the game if $b' = b$.

The advantage of an adversary for winning the payload-hiding game is defined as

$$\text{Adv}_{\mathcal{A}}^{PH}(1^\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|. \quad (5)$$

Definition 5 (payload-hiding for private predicate encryption). We say that private PE is payload-hiding if there is no probabilistic polynomial-time adversary \mathcal{A} winning the above payload-hiding game with a nonnegligible advantage.

Next, we define the ‘‘attribute-hiding’’ security for private PE, which can be also extended from the attribute-hiding definition for conventional PE [7]. Attribute-hiding security models that there is no adversary can obtain any information of the ciphertext attribute \mathbf{x} from the ciphertext. We then

define attribute-hiding via a security game between a challenger \mathcal{C} and an adversary \mathcal{A} .

3.2.2. Attribute-Hiding Game

- (i) Setup, Query Phase 1, Query Phase 2, and Guess are the same as those in the payload-hiding game.
- (ii) *Challenge*. The adversary \mathcal{A} submits two ciphertext attributes $\mathbf{x}^{(0)}, \mathbf{x}^{(1)} \in \mathfrak{A}$ such that $P(\mathbf{x}^{(0)}, \mathbf{y}) = P(\mathbf{x}^{(1)}, \mathbf{y})$ for all $\mathbf{y} \in \mathfrak{B}$, which has been queried to KeyGen Oracle in Query Phase 1, and a message M with the same length to the challenger \mathcal{C} . Then, \mathcal{C} randomly chooses $b \in \{0, 1\}$ and returns a challenge ciphertext $c_{\mathbf{x}^*} \leftarrow Encrypt(pp, SK, \mathbf{x}^{(b)}, M)$.

The advantage of an adversary for winning the attribute-hiding game is defined as

$$\text{Adv}_{\mathcal{A}}^{AH}(1^\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|. \quad (6)$$

Definition 6 (attribute-hiding for private predicate encryption). We say that private PE is attribute-hiding, if there is no probabilistic polynomial-time adversary \mathcal{A} winning the above attribute-hiding game with a nonnegligible advantage.

There is another weaker notion, called ‘‘weak attribute-hiding’’ [25]. The weak attribute-hiding game is the same as the above attribute-hiding game, except the following:

- (i) The adversary sends $(\mathbf{x}^{(0)}, M_0), (\mathbf{x}^{(1)}, M_1)$ to invoke the Challenge phase.
- (ii) The restriction on $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$ is modified to ‘‘ $P(\mathbf{x}^{(0)}, \mathbf{y}) = P(\mathbf{x}^{(1)}, \mathbf{y}) = 0$ for all $\mathbf{y} \in \mathfrak{B}$ which has been queried to KeyGen Oracle in Query Phase 1.’’

Furthermore, we define the ‘‘predicate-hiding’’ for private PE scheme via the following game, which models the notion that a secret key $sk_{\mathbf{y}}$ reveals nothing about the key attribute \mathbf{y} .

3.2.3. Predicate-Hiding Game

- (i) Setup, Query Phase 1, Query Phase 2, and Guess are the same as those in the payload-hiding game.
- (ii) *Challenge*. The adversary \mathcal{A} submits two key attributes $\mathbf{y}^{(0)}, \mathbf{y}^{(1)} \in \mathfrak{B}$ to the challenger \mathcal{C} , such that $P(\mathbf{x}, \mathbf{y}^{(0)}) = P(\mathbf{x}, \mathbf{y}^{(1)}) = 0$ for all $\mathbf{x} \in \mathfrak{A}$ which has been queried to Encrypt Oracle in Query Phase 1. Then, \mathcal{C} randomly chooses $b \in \{0, 1\}$ and returns a challenge secret key $sk_{\mathbf{y}^{(b)}} \leftarrow KeyGen(pp, SK, \mathbf{y}^{(b)})$.

The advantage of an adversary for winning the predicate-hiding game is defined as

$$\text{Adv}_{\mathcal{A}}^{PP}(1^\lambda) = \left| \Pr[b' = b] - \frac{1}{2} \right|. \quad (7)$$

Definition 7 (predicate-hiding for private predicate encryption). We say that private PE achieves predicate-hiding if there is no probabilistic polynomial-time adversary \mathcal{A} winning the above predicate-hiding game with non-negligible advantage.

4. A Private IPE from Key-Homomorphic PRF

In the following, we describe how to obtain a private IPE from a key-homomorphic PRF. In our construction, we require that $\mathcal{R} = \mathbb{Z}_q$, for some prime q . Additionally, we assume that the decryptor knows the value of predicate vector \mathbf{y} of his/her secret key $sk_{\mathbf{y}}$.

- (i) *Setup*($1^\lambda, 1^\ell$). Suppose that the message space is $\{0, 1\}^m$ for some positive integer $m = \text{poly}(\lambda)$. Given the security parameters (λ, ℓ) , where $\lambda, \ell \in \mathbb{N}$, the algorithm outputs the system parameter pp and the secret key SK as follows:
 - (i) Choose a prime $q = \text{poly}(\lambda)$.
 - (ii) Choose a key-homomorphic PRF $F: \mathcal{R} \times \mathcal{X} \rightarrow \mathcal{R}$, where \mathcal{X} is the domain and $(\mathcal{R}, +, \cdot)$ is a ring.
 - (iii) Choose $\mathbf{a} = (a_1, \dots, a_\ell) \in \mathcal{R}^\ell$.
 - (iv) Choose a pseudorandom generator $G: \mathcal{R} \rightarrow \{0, 1\}^m$.
 - (v) Choose $h \leftarrow \mathcal{X}$.
 - (vi) Output $pp = (F, G)$ and the secret key $SK = (\mathbf{a}, h)$.

Note that the descriptions of F and G are implicitly included in the system parameter pp .

- (ii) *Encrypt*(pp, SK, \mathbf{x}, M). Given the system parameter pp , a secret key SK , an attribute vector $\mathbf{x} = (x_1, \dots, x_\ell) \in \mathcal{R}^\ell$, and a message M , the algorithm runs the following steps:
 - (i) Choose random $\delta, \sigma \xleftarrow{\$} \mathcal{R}$.
 - (ii) $c_i = F(\sum_{j=1}^{\delta} (x_j + a_i, h) + \sigma)$ for $i = 1, \dots, \ell$.
 - (iii) $c_0 = M \oplus G(\sigma)$.
 - (iv) Output ciphertext $ct_{\mathbf{x}} = (c_0, \dots, c_\ell) \in \{0, 1\}^m \times \mathcal{R}^\ell$.
- (iii) *KeyGen*(pp, SK, \mathbf{y}). Given the system parameter pp , a secret key SK , and a predicate vector $\mathbf{y} = (y_1, \dots, y_\ell) \in \mathcal{R}^\ell$, the algorithm computes the following steps:
 - (i) $sk_{\mathbf{y}} = (\sum_{i=1}^{\ell} \sum_{j=1}^{y_i} F(a_i, h)) = F(\langle \mathbf{a}, \mathbf{y} \rangle, h)$.
 - (ii) Output $sk_{\mathbf{y}}$.
- (iv) *Decrypt*($pp, ct_{\mathbf{x}}, sk_{\mathbf{y}}$). Given the system parameter pp , a ciphertext $ct_{\mathbf{x}}$, and a secret key $sk_{\mathbf{y}}$, the algorithm computes the following steps:
 - (i) $ct' = \sum_{i=1}^{\ell} (y_i \cdot c_i) - sk_{\mathbf{y}}$.
 - (ii) Compute $\sigma = ct' \cdot (\sum_{i=1}^{\ell} y_i)^{-1}$.
 - (iii) Compute $M = c_0 \oplus G(\sigma)$.

Correctness. Let $ct_{\mathbf{x}}$ and $sk_{\mathbf{y}}$ be as above. Then,

$$\begin{aligned} ct' &= \sum_{i=1}^{\ell} (y_i \cdot c_i) - sk_{\mathbf{y}} \\ &= \sum_{i=1}^{\ell} \left(y_i \cdot \left(F\left(\sum_{j=1}^{\delta} x_j + a_i, h\right) + \sigma \right) \right) - sk_{\mathbf{y}} \\ &= F\left(\sum_{j=1}^{\delta} \langle \mathbf{x}, \mathbf{y} \rangle + \langle \mathbf{a}, \mathbf{y} \rangle, h\right) + \sum_{i=1}^{\ell} y_i \sigma - F(\langle \mathbf{a}, \mathbf{y} \rangle, h). \end{aligned} \quad (8)$$

If $\langle \mathbf{x}, \mathbf{y} \rangle = 0$, we have

$$ct' = F(\langle \mathbf{a}, \mathbf{y} \rangle, h) + \sum_{i=1}^{\ell} y_i \sigma - F(\langle \mathbf{a}, \mathbf{y} \rangle, h) = \sum_{i=1}^{\ell} y_i \sigma. \quad (9)$$

Then, we can compute $\sigma = ct' \cdot (\sum_{i=1}^{\ell} y_i)^{-1}$, and the plaintext can be decrypted by

$$c_0 \oplus G(\sigma) = M \oplus G(\sigma) \oplus G(\sigma) = M. \quad (10)$$

Our scheme accommodates approximate homomorphism [26], as long as the error term is bounded.

5. Security Proofs

5.1. Payload-Hiding Security. We prove the payload-hiding security of our scheme using the sequence-of-game approach [27]. Let $(c_0, c_1, \dots, c_\ell)$ be the challenge ciphertext given to the adversary in the payload-hiding game. Besides, let R_0 be a random element in $\{0, 1\}^m$ and let R_1, \dots, R_ℓ be random elements in \mathcal{R} . We define the following hybrid games differing in what challenge ciphertext is sent to the adversary:

- (i) *Game₀*. The challenge ciphertext is $(c_0, c_1, \dots, c_\ell)$. It is identical to the original payload-hiding game defined in Section 3.2.
- (ii) *Game_i*, $1 \leq i \leq \ell$. The challenge ciphertext is $(c_0, R_1, \dots, R_i, c_{i+1}, \dots, c_\ell)$.
- (iii) *Game_{\ell+1}*. The challenge ciphertext is $(R_0, R_1, \dots, R_\ell)$.

We remark that the challenge ciphertext in *Game_{\ell+1}* leaks no information about the encrypted message, since it is composed of $\ell + 1$ random elements, whereas the challenge ciphertext in *Game₀* is well formed. Therefore, the advantage of the adversary in the last game is 0. We then prove the indistinguishability between the adjacent games in the following lemmas.

Lemma 1. *If the underlying PRF F is secure, then $Game_{k-1}$ is indistinguishable from $Game_k$, for $k = 1, \dots, \ell$.*

Proof. Suppose that there is an adversary \mathcal{A} that is able to distinguish $Game_{k-1}$ from $Game_k$ with a nonnegligible advantage. Then we can build a challenger \mathcal{C}_1 to distinguish the experiment $EXP(0)$ from the experiment $EXP(1)$

shown in Section 2. After invoking the experiment $EXP(b)$ and receiving the description of the PRF F , the challenger \mathcal{C}_1 simulates a hybrid game for an adversary \mathcal{A} as follows:

Setup. The challenger first randomly chooses $a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_\ell$ from \mathcal{R} and h from \mathcal{X} and a pseudorandom generator G and then sends $pp = (F, G)$ to the adversary. Next, the challenger makes a Challenge query with h to the underlying experiment and obtains f as the response. The value of f will be used in the later simulation for KeyGen and Encryption Oracle.

Query Phase 1. In this phase, the adversary is allowed to make polynomially many queries to the following oracles.

- (i) KeyGen Oracle: taking as inputs a vector $\mathbf{y} = (y_1, \dots, y_\ell) \in \mathcal{R}^\ell$, the challenger computes

$$\begin{aligned} sk_{\mathbf{y}} &= \sum_{j=1}^{y_1} F(a_1, h) + \dots + \sum_{j=1}^{y_{k-1}} F(a_{k-1}, h) \\ &+ \sum_{j=1}^{y_k} f + \sum_{j=1}^{y_{k+1}} F(a_{k+1}, h) + \dots + \sum_{j=1}^{y_\ell} F(a_\ell, h) \quad (11) \\ &= \sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i F(a_i, h) + y_k f, \end{aligned}$$

and returns $sk_{\mathbf{y}}$ to the adversary. By implicitly setting a_j to the chosen key of the underlying experiment, it is easy to verify that $sk_{\mathbf{y}}$ is a valid secret key for \mathbf{y} .

- (ii) Encryption Oracle: taking as inputs a vector $\mathbf{x} = (x_1, \dots, x_\ell) \in \mathcal{R}^\ell$ and a message M , the challenger performs as follows:

- (1) Randomly choose δ, σ from \mathcal{R} .
- (2) Compute $c_k = F(\sum_{j=1}^{\delta} (x_k), h) + f + \sigma = F(\delta x_k, h) + f + \sigma$.
- (3) For $i = 1, \dots, k-1, k+1, \dots, \ell$, compute c_i the same as in the *Encrypt* algorithm since the challenger knows $a_1, \dots, a_{k-1}, a_{k+1}, \dots, a_\ell, h$.
- (4) Compute $c_0 = M \oplus G(\sigma)$.
- (5) Return $ct_{\mathbf{x}} = (c_0, c_1, \dots, c_\ell)$.

Challenge. The adversary submits two messages M_0, M_1 with the same length and a vector $\mathbf{x}^* = (x_1^*, \dots, x_\ell^*)$, such that $\langle \mathbf{x}^*, \mathbf{y} \rangle \neq 0$ for all \mathbf{y} queried to KeyGen Oracle. After receiving \mathbf{x}^*, M_0, M_1 , the challenger randomly chooses $\beta \leftarrow \mathcal{R}$ and then can compute the challenge ciphertext ct^* as follows:

- (1) Randomly choose δ, σ from \mathcal{R} .
- (2) For $i = 1, \dots, \ell$,
 - (i) if $i < k$, choose a random element $R_i \xleftarrow{\$}$ and set $c_i = R_i$.
 - (ii) if $i = k$, compute
$$c_k = F\left(\sum_{j=1}^{\delta} (x_k^*, h)\right) + f + \sigma = F(\delta x_k^*, h) + f + \sigma. \quad (12)$$
- (iii) if $i > k$, compute c_i the same way as that in the scheme since the challenger knows a_{k+1}, \dots, a_ℓ and h .

- (3) Compute $c_0 = M_\beta \oplus G(\sigma)$.

- (4) Return $ct^* = (c_0, c_1, \dots, c_\ell)$.

Query Phase 2. It is the same as Query Phase 1 except that the adversary is not allowed to make a query to KeyGen Oracle with \mathbf{y} such that $\langle \mathbf{x}^*, \mathbf{y} \rangle = 0$.

Guess. The adversary outputs a bit β' . Then the challenger outputs 1 if $\beta' = \beta$ and 0 otherwise. Before analyzing the advantages of the challenger in breaking the underlying PRF, we first discuss that the outputs of the oracles are well formed, no matter which experiment the challenger interacts with. Let S_i be the event where the adversary makes a right guess in $Game_i$. First, if the challenger is actually interacting with the experiment $EXP(0)$, then f is a random element in \mathcal{R} . In this case, the answer to a KeyGen Oracle,

$$sk_{\mathbf{y}} = \sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i F(a_i, h) + y_k f, \quad (13)$$

is an element of \mathcal{R} and the answer to an Encryption query $(c_0, c_1, \dots, c_\ell)$ is a vector in $\{0, 1\}^m \times \mathcal{R}^\ell$, and

$$\begin{aligned} \sum_{i=1}^{\ell} (y_i \cdot c_i) - sk_{\mathbf{y}} &= \sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i (F(\delta x_i + a_i, h) + \sigma) \\ &+ y_k (F(\delta x_k, h) + f + \sigma) \\ &- \left(\sum_{\substack{i=1 \\ i \neq k}}^{\ell} y_i F(a_i, h) + y_k f \right) \quad (14) \\ &= \sum_{i=1}^{\ell} y_i F(\delta x_i, h) + \sum_{i=1}^{\ell} \sigma \\ &= F\left(\delta \sum_{i=1}^{\ell} x_i y_i, h\right) + \sum_{i=1}^{\ell} \sigma \\ &= \sum_{i=1}^{\ell} \sigma \\ &\iff \langle \mathbf{x}, \mathbf{y} \rangle = 0. \end{aligned}$$

Therefore, the answers to KeyGen and Encryption queries are well formed.

Next, we analyze the advantage of \mathcal{C}_1 in breaking the underlying PRF. First, if the challenger is interacting with the experiment $EXP(0)$, then f is a random element in \mathcal{R} . Thus, c_1, \dots, c_k in the challenge ciphertext are random elements, and thus we are in $Game_k$. Thus, the probability that the challenger outputs 1 is

$$\Pr[S_k] = \Pr[\mathcal{C}_1 \text{ outputs } 1] = \Pr[\beta' = \beta] = \Pr[W_0]. \quad (15)$$

Second, if the challenger is interacting with the experiment $EXP(1)$, then f is the output of the PRF with input h . By implicitly setting the encryption key component a_k as the chosen key of the underlying experiment, we have $f = F(a_k, h)$, and thus the challenger answers the KeyGen

and Encryption queries correctly. As for the challenge ciphertext, we have that

$$c_k = F\left(\sum_{j=1}^{\delta} (x_k^*), h\right) + f + \sigma = F(\delta x_k^*, h) + F(a_k, h) + \sigma = F(\delta x_k^* + a_k, h) + \sigma, \quad (16)$$

is a valid ciphertext component. Since c_1, \dots, c_{k-1} are random elements from \mathcal{R} , we are in Game_{k-1} . In this case, the probability that the challenger outputs 1 is

$$\Pr[S_{k-1}] = \Pr[\mathcal{E}_1 \text{ outputs } 1] = \Pr[\beta' = \beta] = \Pr[W_1]. \quad (17)$$

Finally, combining the above two cases, we have that

$$|\Pr[S_{k-1}] - \Pr[S_k]| = |\Pr[W_1] - \Pr[W_0]| = \text{Adv}_{\mathcal{E}_1}^{\text{PRF}}(1^\lambda), \quad (18)$$

and hence, Game_{k-1} is indistinguishable from Game_k , if the underlying pseudorandom function is secure, for $k = 1, \dots, \ell$. \square

Lemma 2. *If the underlying PRG G is secure, then Game_ℓ is indistinguishable from $\text{Game}_{\ell+1}$.*

Proof. Given the description of the PRG G and a challenge $\psi \in \{0, 1\}^m$, the challenger \mathcal{E}_2 simulates the following hybrid game for an adversary \mathcal{A} :

Setup. The challenger first chooses a key-homomorphic pseudorandom function $F: \mathcal{R} \times \mathcal{X} \rightarrow \mathcal{R}$, a_1, \dots, a_ℓ from \mathcal{R} and h from \mathcal{X} and then sends (F, G) to the adversary.

Query Phase 1. The challenger is able to answer the KeyGen (Encryption, resp.) queries by following the KeyGen (Encrypt, resp.) algorithms to generate the secret keys sk_y (ciphertexts ct_x , resp.), since the challenger knows the secret key $SK = (a_1, \dots, a_\ell, h)$.

Challenge. The adversary submits two messages M_0, M_1 with the same length and a vector \mathbf{x}^* , such that $\langle \mathbf{x}^*, \mathbf{y} \rangle \neq 0$ for all \mathbf{y} queried to KeyGen Oracle. After receiving \mathbf{x}^* , M_0, M_1 , the challenger randomly chooses $\beta \leftarrow \{0, 1\}$ and then can compute the challenge ciphertext ct^* as follows:

- (1) Randomly choose $R_1, \dots, R_\ell \xleftarrow{\$} \mathcal{R}$.
- (2) For $i = 1, \dots, \ell$, set $c_i = R_i$.
- (3) Compute $c_0 = M_\beta \oplus \Psi$.
- (4) Return the challenge ciphertext $ct^* = (c_0, c_1, \dots, c_\ell)$.

Query Phase 2. It is the same as Query Phase 1 except that the adversary is not allowed to make a query to KeyGen Oracle with \mathbf{y} such that $\langle \mathbf{x}^*, \mathbf{y} \rangle = 0$.

Guess. The adversary outputs a bit β' . Then, the challenger outputs 1 if $\beta' = \beta$. Let S_i be the event where the adversary makes a right guess in Game_i . If the term $\psi = G(\sigma)$ is generated from the PRG G for some σ , then we are in Game_ℓ , and we have

$$\Pr[S_\ell] = \Pr[\mathcal{E}_2(\psi = G(\sigma)) = 1]. \quad (19)$$

If ψ is randomly chosen from $\{0, 1\}^m$, then we are in $\text{Game}_{\ell+1}$, and we have

$$\Pr[S_{\ell+1}] = \Pr\left[\mathcal{E}_2\left(\psi \xleftarrow{\$} \{0, 1\}^m\right) = 1\right]. \quad (20)$$

Finally, we have that

$$\begin{aligned} |\Pr[S_\ell] - \Pr[S_{\ell+1}]| &= \left| \Pr[\mathcal{E}_2(\psi = G(\sigma)) = 1] - \Pr\left[\mathcal{E}_2\left(\psi \xleftarrow{\$} \{0, 1\}^m\right) = 1\right] \right| \\ &= \text{Adv}_{\mathcal{E}_2}^{\text{PRG}}(1^\lambda) \end{aligned} \quad (21)$$

is negligible. \square

Theorem 1. *The proposed private IPE scheme achieves payload-hiding, if the underlying pseudorandom function is key-homomorphic and secure and the pseudorandom generator is secure.*

Proof. By combining Lemmas 1 and 2, we have

$$\begin{aligned} |\Pr[S_0] - \Pr[S_{\ell+1}]| &= \left| \sum_{i=1}^{\ell} (\Pr[S_{i-1}] - \Pr[S_i]) + (\Pr[S_\ell] - \Pr[S_{\ell+1}]) \right| \\ &\leq |\Pr[S_0] - \Pr[S_1]| + \dots + |\Pr[S_{\ell-1}] - \Pr[S_\ell]| + |\Pr[S_\ell] - \Pr[S_{\ell+1}]| \\ &= \underbrace{\text{Adv}_{\mathcal{E}_1}^{\text{PRF}}(1^\lambda) + \dots + \text{Adv}_{\mathcal{E}_1}^{\text{PRF}}(1^\lambda)}_{\ell} + \text{Adv}_{\mathcal{E}_2}^{\text{PRG}}(1^\lambda) \\ &= \ell \cdot \text{Adv}_{\mathcal{E}_1}^{\text{PRF}}(1^\lambda) + \text{Adv}_{\mathcal{E}_2}^{\text{PRG}}(1^\lambda). \end{aligned} \quad (22)$$

Note that $\Pr[S_0] = \text{Adv}_{\mathcal{A}}^{\text{PH}}(1^\lambda)$ since Game_0 is the payload-hiding game, and $\Pr[S_{\ell+1}] = 0$ since ct^* leaks no information about the encrypted message in $\text{Game}_{\ell+1}$.

Therefore, for any PPT adversary \mathcal{A} , there exist algorithms $\mathcal{C}_1, \mathcal{C}_2$ such that

$$\text{Adv}_{\mathcal{A}}^{\text{PH}}(1^\lambda) = \Pr[S_0] = |\Pr[S_0] - \Pr[S_{\ell+1}]| \leq \ell \cdot \text{Adv}_{\mathcal{C}_1}^{\text{PRF}}(1^\lambda) + \text{Adv}_{\mathcal{C}_2}^{\text{PRG}}(1^\lambda). \quad (23)$$

is negligible. \square

5.2. Attribute-Hiding Security. We then prove that our scheme achieves attribute-hiding. The proof is similar to the proof for payload-hiding security, and hence we will omit some content to avoid the unnecessary redundancy. Let $(c_0, c_1, \dots, c_\ell)$ be the challenge ciphertext given to the adversary in the attribute-hiding game. Besides, let R_1, \dots, R_ℓ be random elements in \mathcal{R} and let R_0 be a random element in $\{0, 1\}^m$. We define the following hybrid games differing in what challenge ciphertext is sent to the adversary:

- (i) Game_0 . The challenge ciphertext is $(c_0, c_1, \dots, c_\ell)$. It is identical to the original attribute-hiding game defined in Section 3.2.
- (ii) $\text{Game}_i, 1 \leq i \leq \ell$. The challenge ciphertext is $(c_0, R_1, \dots, R_i, c_{i+1}, \dots, c_\ell)$.
- (iii) $\text{Game}_{\ell+1}$. The challenge ciphertext is $(R_0, R_1, \dots, R_\ell)$.

In the last game, the challenge ciphertext is composed of $\ell + 1$ random elements, and hence the adversary obtains no information about the attribute vector from the challenge ciphertext. We then prove that the adjacent games are indistinguishable in the following lemmas.

Lemma 3. *If the underlying PRF F is secure, then Game_{k-1} is indistinguishable from Game_k , for $k = 1, \dots, \ell$.*

Proof. Suppose that there is an adversary \mathcal{A} that is able to distinguish Game_{k-1} from Game_k with a nonnegligible advantage. Then we can build a challenger \mathcal{C}_3 to distinguish the experiment $\text{EXP}(0)$ from the experiment $\text{EXP}(1)$ shown in Section 2. After invoking the experiment $\text{EXP}(b)$ and receiving the description of the PRF F , the challenger \mathcal{C}_3 simulates a hybrid game for an adversary \mathcal{A} as follows.

For Setup, Query Phase 1, Query Phase 2, and Guess, the challenger performs the same as in the proof of Lemma 1.

For Challenge phase, after receiving $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_\ell^{(0)})$, $\mathbf{x}^{(1)} = (x_1^{(1)}, \dots, x_\ell^{(1)})$, and M from the adversary, where

$$\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = 0 = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle \text{ or } \langle \mathbf{x}^{(0)}, \mathbf{y} \rangle \neq 0 \neq \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle, \quad (24)$$

for all \mathbf{y} queried to KeyGen Oracle in Query Phase 1, the challenger performs as follows:

- (1) Randomly choose $\beta \leftarrow_{\mathcal{S}} \{0, 1\}$.
- (2) Randomly choose δ, σ from \mathcal{R} .
- (3) For $i = 1, \dots, \ell$,

- (i) if $i < k$, choose a random element $R_i \leftarrow_{\mathcal{S}} \mathcal{R}$ and set $c_i = R_i$.
- (ii) if $i = k$, compute the following.

$$c_k = F\left(\sum_{j=1}^{\delta} (x_k^{(\beta)}), h\right) + f + \sigma = F(\delta x_k^{(\beta)}, h) + f + \sigma. \quad (25)$$

- (iii) if $i > k$, compute c_i the same way as that in the scheme since the challenger knows a_{k+1}, \dots, a_ℓ and h .

- (4) Compute $c_0 = M \oplus G(\sigma)$.
- (5) Return $ct^* = (c_0, c_1, \dots, c_\ell)$.

The analysis of the correctness of the simulation is similar to that in the proof of Lemma 1. Let S_i be the event where the adversary makes a right guess in Game_i . If f from the PRF game is a random element in \mathcal{R} , then we are in Game_k ; otherwise, we are in Game_{k-1} . Therefore, we have

$$|\Pr[S_{k-1}] - \Pr[S_k]| = |\Pr[W_1] - \Pr[W_0]| = \text{Adv}_{\mathcal{C}_3}^{\text{PRF}}(1^\lambda). \quad (26)$$

That is, Game_k is indistinguishable from Game_{k-1} , if the underlying pseudorandom function is secure, for $k = 1, \dots, \ell$. \square

Lemma 4. *If the underlying PRG G is secure, then Game_ℓ is indistinguishable from $\text{Game}_{\ell+1}$.*

Proof. The proof of this lemma is similar to the proof of Lemma 2, with the only difference being that the challenger received two vectors $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$ with a message M ; in Lemma 2, the challenger received two messages M_0, M_1 with a vector \mathbf{x}^* from the adversary.

Given the description of the PRG G and a challenge $\psi \in \{0, 1\}^m$, the challenger \mathcal{C}_4 simulates the following hybrid game for an adversary \mathcal{A} .

For Setup, Query Phase 1, Query Phase 2, and Guess, the challenger performs the same as in the proof of Lemma 1.

For Challenge phase, after receiving $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$, and M from the adversary, where

$$\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = 0 = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle \text{ or } \langle \mathbf{x}^{(0)}, \mathbf{y} \rangle \neq 0 \neq \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle, \quad (27)$$

for all \mathbf{y} queried to KeyGen Oracle in Query Phase 1, the challenger performs as follows:

- (1) Randomly choose $R_1, \dots, R_\ell \leftarrow_{\mathcal{S}} \mathcal{R}$.
- (2) For $i = 1, \dots, \ell$, set $c_i = R_i$.

(3) Compute $c_0 = M \oplus \psi$.

(4) Return the challenge ciphertext $ct^* = (c_0, c_1, \dots, c_\ell)$.

The analysis of the correctness of the simulation is similar to that in the proof of Lemma 3. Let S_i be the event where the adversary makes a right guess in $Game_i$. If ψ from the PRG game is a random element in $\{0, 1\}^m$, then we are in $Game_{\ell+1}$; otherwise, we are in $Game_\ell$. Therefore, we have that

$$\begin{aligned} |\Pr[S_\ell] - \Pr[S_{\ell+1}]| &= \left| \Pr[\mathcal{E}_4(\psi = G(\sigma)) = 1] - \Pr[\mathcal{E}_4(\psi \xleftarrow{\$} \{0, 1\}^m) = 1] \right| \\ &= \text{Adv}_{\mathcal{E}_4}^{\text{PRG}}(1^\lambda), \end{aligned} \quad (28)$$

$$\begin{aligned} |\Pr[S_0] - \Pr[S_{\ell+1}]| &= \left| \sum_{i=1}^{\ell} (\Pr[S_{i-1}] - \Pr[S_i]) + (\Pr[S_\ell] - \Pr[S_{\ell+1}]) \right| \\ &\leq |\Pr[S_0] - \Pr[S_1]| + \dots + |\Pr[S_{\ell-1}] - \Pr[S_\ell]| + |\Pr[S_\ell] - \Pr[S_{\ell+1}]| \\ &= \underbrace{\text{Adv}_{\mathcal{E}_3}^{\text{PRF}}(1^\lambda) + \dots + \text{Adv}_{\mathcal{E}_3}^{\text{PRF}}(1^\lambda)}_{\ell} + \text{Adv}_{\mathcal{E}_4}^{\text{PRG}}(1^\lambda) \\ &= \ell \cdot \text{Adv}_{\mathcal{E}_3}^{\text{PRF}}(1^\lambda) + \text{Adv}_{\mathcal{E}_4}^{\text{PRG}}(1^\lambda). \end{aligned} \quad (29)$$

Note that $\Pr[S_0] = \text{Adv}_{\mathcal{A}}^{\text{AH}}(1^\lambda)$ since $Game_0$ is the attribute-hiding game, and $\Pr[S_{\ell+1}] = 0$ since ct^* leaks no information about the encrypted message in $Game_{\ell+1}$.

is negligible. That is, $Game_\ell$ is indistinguishable from $Game_{\ell+1}$, if the underlying pseudorandom generator is secure. \square

Theorem 2. *The proposed private IPE scheme achieves attribute-hiding, if the underlying pseudorandom function is key-homomorphic and secure and the pseudorandom generator is secure.*

Proof. By combining Lemma 3 and Lemma 4, we have

Therefore, for any PPT adversary \mathcal{A} , there exist algorithms $\mathcal{E}_3, \mathcal{E}_4$ such that

$$\text{Adv}_{\mathcal{A}}^{\text{AH}}(1^\lambda) = \Pr[S_0] = |\Pr[S_0] - \Pr[S_{\ell+1}]| \leq \ell \cdot \text{Adv}_{\mathcal{E}_3}^{\text{PRF}}(1^\lambda) + \text{Adv}_{\mathcal{E}_4}^{\text{PRG}}(1^\lambda) \quad (30)$$

is negligible. \square

5.3. Predicate-Hiding Security. We first give the intuition for the proof. Let $[y_1, y_2, \dots, y_\ell]$ denote the challenge secret key generated using the vector $(y_1, y_2, \dots, y_\ell)$. Besides, let $\mathbf{y}^{(0)} = (y_1^{(0)}, y_2^{(0)}, \dots, y_\ell^{(0)})$, $\mathbf{y}^{(1)} = (y_1^{(1)}, y_2^{(1)}, \dots, y_\ell^{(1)})$ be the two vectors sent from the adversary in the Challenge phase. To prove the indistinguishability between the cases $[y_1^{(0)}, y_2^{(0)}, \dots, y_\ell^{(0)}]$ and $[y_1^{(1)}, y_2^{(1)}, \dots, y_\ell^{(1)}]$ given to the adversary, we define a sequence of games below and show the indistinguishability of any two adjacent games. Each

game differs in the challenge secret key given to the adversary. Let $y'_1, y'_2, \dots, y'_\ell$ be random elements from \mathcal{R} .

$Game_{0,i}$: $[y'_1, y'_2, \dots, y'_{k-1}, y_k^{(0)}, \dots, y_\ell^{(0)}]$ is given ($k = 1, \dots, \ell$)

$Game_{1,i}$: $[y'_1, y'_2, \dots, y'_{k-1}, y_k^{(1)}, \dots, y_\ell^{(1)}]$ is given ($k = 1, \dots, \ell$)

Note that $Game_{0,\ell}$ and $Game_{1,\ell}$ are identical, and $Game_{0,0}$ and $Game_{1,0}$ are the games where $[y_1^{(0)}, \dots, y_\ell^{(0)}]$ and $[y_1^{(1)}, \dots, y_\ell^{(1)}]$ are given to the adversary, respectively. We then give the following lemma to prove that

$$Game_{0,0} \approx Game_{0,1} \approx \dots \approx Game_{0,\ell} \approx Game_{1,\ell} \approx \dots \approx Game_{1,1} \approx \dots \approx Game_{1,0}. \quad (31)$$

Lemma 5. *If the underlying PRF F is secure, then $Game_{0,k-1}$ and $Game_{0,k}$ are indistinguishable, for $k = 1, \dots, \ell$.*

Proof. Suppose that there is an adversary \mathcal{A} which is able to distinguish $Game_{k-1}$ from $Game_k$ with a nonnegligible

advantage. Then we can build a challenger \mathcal{C} to distinguish the experiment $EXP(0)$ from $EXP(1)$ shown in Section 2. After invoking the experiment $EXP(b)$ and receiving the description of the PRF F , the challenger \mathcal{C} simulates a hybrid game for an adversary \mathcal{A} as follows.

TABLE 1: Comparison with other related private IPE schemes [17–19]. Here, the length of ciphertext attribute and key attribute is n . $|\mathbb{G}|$ and m represent size of an element of $|\mathbb{G}|$ and message, respectively. MSK, SK, CT, Qun. Res., GSD, C3DH, and 3FCOBGA stand for master secret key, secret key for some key attribute, ciphertext for some ciphertext attribute, quantum-resistant, general subgroup decision, composite 3-party (decisional) Diffie-Hellman, and 3-factor-based composite-order bilinear groups assumption, respectively.

	SSW09 [18]	YKNS12 [19]	KT13 [17]	Ours
Security	Selective	Selective	Adaptive	Adaptive
Order of \mathbb{G}	Composite	Composite	Prime	—
Assumption	A variant of GSD, C3DH, DLIN	3FCOBGA	DLIN	PRF
MSK size	$(4n + 4) \mathbb{G} $	$(4n + 4) \mathbb{G} $	$5n \mathbb{G} $	$n \log_2 q$
SK size	$(2n + 2) \mathbb{G} $	$(2n + 2) \mathbb{G} $	$6n \mathbb{G} $	$\log_2 q$
CT size	$(2n + 2) \mathbb{G} $	$(2n + 2) \mathbb{G} $	$6n \mathbb{G} $	$m + n \log_2 q$
Qun. Res.	No	No	No	Yes [†]

[†]If the underlying PRF is resistant to quantum attacks, then our proposed scheme is resistant to quantum attacks.

For Setup, Query Phase 1, Query Phase 2, and Guess, the challenger performs the same as in the proof of Lemma 1.

For Challenge phase, after receiving $\mathbf{y}^{(0)} = (y_1^{(0)}, \dots, y_\ell^{(0)})$, $\mathbf{y}^{(1)} = (y_1^{(1)}, \dots, y_\ell^{(1)})$ from the adversary, where

$$\langle \mathbf{x}, \mathbf{y}^{(0)} \rangle = 0 = \langle \mathbf{x}, \mathbf{y}^{(1)} \rangle \text{ or } \langle \mathbf{x}, \mathbf{y}^{(0)} \rangle \neq 0 \neq \langle \mathbf{x}, \mathbf{y}^{(1)} \rangle, \quad (32)$$

for all \mathbf{x} queried to Encrypt Oracle in Query Phase 1, the challenger performs as follows.

- (1) Randomly choose $y'_1, y'_2, \dots, y'_{k-1}$ from \mathcal{R} .
- (2) Compute

$$sk^* = \sum_{j=1}^{y'_1} F(a_1, h) + \dots + \sum_{j=1}^{y'_{k-1}} F(a_{k-1}, h) + \sum_{j=1}^{y_k^{(0)}} f + \sum_{j=1}^{y_{k+1}^{(0)}} F(a_{k+1}, h) + \dots + \sum_{j=1}^{y_\ell^{(0)}} F(a_\ell, h). \quad (33)$$

- (3) Return sk^* .

If the challenger is interacting with the experiment $EXP(1)$, then f is the output of the PRF with input h . By

implicitly setting the encryption key component a_k as the chosen key of the underlying experiment, we have $f = F(a_k, h)$, and thus we have

$$\begin{aligned} sk^* &= y'_1 F(a_1, h) + \dots + y'_{k-1} F(a_{k-1}, h) + y_k^{(0)} F(a_k, h) \\ &\quad + y_{k+1}^{(0)} F(a_{k+1}, h) + \dots + y_\ell^{(0)} F(a_\ell, h) \\ &= [y'_1, \dots, y'_{k-1}, y_k^{(0)}, \dots, y_\ell^{(0)}], \end{aligned} \quad (34)$$

and thus we are in $Game_{k-1}$. Otherwise, f is a random element in \mathcal{R} ; then we can rewrite $f = F(a_k, h) + \tilde{R}$ for some random element $\tilde{R} \in \mathcal{R}$. Besides, there must exist an element

\tilde{y} such that $\tilde{R} = (y_k^{(0)})^{-1} \tilde{y} F(a_k, h)$. By implicitly setting $y'_k = y_k^{(0)} + \tilde{y}$, we have

$$\begin{aligned} \sum_{i=1}^{y_k^{(0)}} f &= y_k^{(0)} f = y_k^{(0)} (F(a_k, h) + \tilde{R}) = y_k^{(0)} F(a_k, h) + y_k^{(0)} (y_k^{(0)})^{-1} \tilde{y} F(a_k, h) \\ &= y_k^{(0)} F(a_k, h) + \tilde{y} F(a_k, h) = (y_k^{(0)} + \tilde{y}) F(a_k, h) = y'_k F(a_k, h). \end{aligned} \quad (35)$$

Since f is a random element in \mathcal{R} , y'_k is also a random element in \mathcal{R} . That means $sk^* = [y'_1, \dots, y'_k, y_{k+1}^{(0)}, \dots, y_\ell^{(0)}]$, and thus we are in $Game_k$. Let S_i be the event where the adversary makes a right guess in $Game_i$. Therefore, we have

$$|\Pr[S_{k-1}] - \Pr[S_k]| = |\Pr[W_1] - \Pr[W_0]| = \text{Adv}_{\mathcal{G}}^{\text{PRF}}(1^\lambda). \quad (36)$$

That is, $Game_k$ is indistinguishable from $Game_{k-1}$, if the underlying PRF is secure, for $k = 1, \dots, \ell$. \square

Theorem 3. *The proposed private IPE scheme achieves predicate-hiding, if the underlying pseudorandom function is key-homomorphic and secure and the pseudorandom generator is secure.*

Proof. The proof for the indistinguishability between $Game_{1,k-1}$ and $Game_{1,k}$ is the same as that for the indistinguishability between $Game_{0,k-1}$ and $Game_{0,k}$, due to the symmetry of the game sequence. This completes the proof of the predicate-hiding. \square

6. Comparison and Analysis

To the best of our knowledge, although existing private IPE schemes [17–19] can resist payload-hiding, attribute-hiding, and predicate-hiding security, these schemes cannot resist quantum attacks because their security is based on discrete logarithm assumption. In this section, we compare our scheme with the existing private IPE schemes in terms of security properties and the size of master secret key, secret key, and ciphertext, as shown in Table 1.

The results show that our construction has higher security and efficiency in terms of secret key size because the size is not related to attribute length. In particular, the security of [18, 19] is only selective security; meanwhile that in [17] and our construction is adaptive security, making it more resistant to real attacks. In secret key size, our construction is of constant size, while the secret key sizes of [17–19] are linearly related to the key attribute length. In terms of ciphertext size, the encryption algorithm in schemes [17–19] only encrypts ciphertext predicate, while our proposed construction further encrypts message; therefore, the ciphertext size of our scheme is $m + n \log_2 q$, where m is the length of message. Finally, [17–19] are not resistant to quantum attacks, while our construction is resistant to quantum attacks if the underlying PRF is resistant to quantum attacks.

7. Conclusions and Future Works

With the development of cloud computing, the privacy of uploaded data needs to be concerned and protected. Private IPE is well suited to cloud computing scenario because it provides encryption for access control. In this paper, we propose a generic private IPE construction that achieves payload-hiding, attribute-hiding, and predicate-hiding security by utilizing specific key-homomorphic PRF. For

future works, because the current construction requires that the key space and output space of the underlying key-homomorphic PRF be \mathbb{Z}_q , how to provide construction with less restriction is an open problem that remains to be solved.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Ministry of Science and Technology, Taiwan (ROC), under Project nos. MOST 108-2218-E-004-001-, MOST 108-2218-E-004-002-MY2, MOST 109-2218-E-011-007-, and by Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU).

References

- [1] L. Zhou, X. Li, K.-H. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Generation Computer*, vol. 91, pp. 244–251, 2019.
- [2] S. Kumari, P. Chaudhary, C.-M. Chen, and M. K. Khan, "Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications," *IEEE Access*, vol. 7, pp. 39717–39720, 2019.
- [3] P. Wang, C.-M. Chen, S. Kumari, M. Shojafar, R. Tafazolli, and Y. N. Liu, "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Access*, vol. 2020, 2020.
- [4] C.-M. Chen, B. Xiang, K.-H. Wang, K.-H. Yeh, and T.-Y. Wu, "A robust mutual authentication with a key agreement scheme for session initiation protocol," *Applied Sciences*, vol. 8, no. 10, p. 1789, 2018.
- [5] J. Li, S. Wang, Y. Li et al., "An efficient attribute-based encryption scheme with policy update and file update in cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6500–6509, 2019.
- [6] M. Zhang, Yu Chen, and S. E. P. F. M. Jiajun Huang., "A searchable encryption scheme supporting privacy-preserving fuzzy multikeyword in cloud systems," 2020.
- [7] J. Katz, S. Amit, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *EUROCRYPT 2008*. LNCS, N. Smart, Ed., vol. 4965, pp. 146–162, Springer, Berlin, Heidelberg, 2008.
- [8] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," *CCS*, vol. 417–426, 2008.
- [9] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *TCC 2007*. LNCS, S. P. Vadhan, Ed., vol. 4932, pp. 535–554, Springer, Berlin, Heidelberg, 2007.
- [10] V. Iovino and G. Persiano, "Hidden-vector encryption with groups of prime order," in *Pairing 2008*. LNCS, S. D Galbraith and K. G. Paterson, Eds., vol. 5209, pp. 75–88, Springer, Berlin, Heidelberg, 2008.

- [11] M. Naor and B. Pinkas, "Oblivious transfer and polynomial evaluation," *STOC*, vol. 1999, pp. 245–254, 1999.
- [12] A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang, "Threshold ciphertext policy attribute-based encryption with constant size ciphertexts," in *ACISP 2012. LNCS*, W. Susilo, Y. Mu, and J. Seberry, Eds., vol. 7372, pp. 336–349, Springer, Berlin, Heidelberg, 2012.
- [13] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan, "Functional encryption for inner product predicates from learning with errors," in *ASIACRYPT 2011. LNCS*, D. H. Lee and X. Wang, Eds., vol. 7073, pp. 21–40, Springer, Berlin, Heidelberg, 2011.
- [14] T. Okamoto and K. Takashima, "Hierarchical predicate encryption for inner-products," in *ASIACRYPT 2009. LNCS*, M. Matsui, Ed., vol. 5912, pp. 214–231, Springer, Berlin, Heidelberg, 2009.
- [15] T. Okamoto and K. Takashima, "Adaptively attribute-hiding (hierarchical) inner product encryption," in *EUROCRYPT 2012. LNCS*, D. Pointcheval and T. Johansson, Eds., vol. 7237, pp. 591–608, Springer, Berlin, Heidelberg, 2012.
- [16] K. Xagawa, "Improved (hierarchical) inner-product encryption from lattices," in *PKC 2013. LNCS*, K. Kurosawa and G. Hanaoka, Eds., vol. 7778, pp. 235–252, Springer, Berlin, Heidelberg, 2013.
- [17] Y. Kawai and K. Takashima, "Predicate- and attribute-hiding inner product encryption in a public key setting," in *Pairing 2013. LNCS*, Z. Cao and F. Zhang, Eds., vol. 8365, pp. 113–130, Springer, Berlin, Heidelberg, 2013.
- [18] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *TCC 2009. LNCS*, O. Reingold, Ed., vol. 5444, pp. 457–473, Springer, Berlin, Heidelberg, 2009.
- [19] M. Yoshino, N. Kunihiko, K. Naganuma, and H. Sato, "Symmetric inner-product predicate encryption based on three groups," in *ProvSec 2012. LNCS*, T. Takagi, G. Wang, Z. Qin, S. Jiang, and Y. Yu, Eds., vol. 7496, pp. 215–234, Springer, Berlin, Heidelberg, 2012.
- [20] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *FOCS*, vol. 1994, pp. 124–134, 1994.
- [21] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.
- [22] N. Alapati, H. Montgomery, and S. Patranabis, "Ring key-homomorphic weak PRFs and applications," 2020.
- [23] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM*, vol. 33, no. 4, pp. 792–807, 1986.
- [24] D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan, "Key homomorphic PRFs and their applications," in *CRYPTO 2013. LNCS*, R. Canetti and J. A. Garay, Eds., vol. 8042, pp. 410–428, Springer, Berlin, Heidelberg, 2013.
- [25] J. Chen and J. Gong, "ABE with tag made easy," in *ASIACRYPT 2017. LNCS*, T. Takagi and T. Peyrin, Eds., vol. 10625, pp. 35–65, Springer, Berlin, Heidelberg, 2017.
- [26] D. Boneh, S. Eskandarian, S. Kim, and M. Shih, "Improving speed and security in updatable encryption schemes," 2020.
- [27] V. Shoup, "Sequences of games: a tool for taming complexity in security proofs," 2004.

Research Article

Improved Authenticated Key Agreement Scheme for Fog-Driven IoT Healthcare System

Tsu-Yang Wu ^{1,2,3}, Tao Wang ^{2,3}, Yu-Qi Lee ^{2,3}, Weimin Zheng ^{1,2}, Saru Kumari ⁴, and Sachin Kumar ⁵

¹College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

²School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China

³Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China

⁴Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India

⁵Department of Computer Science and Engineering, Ajay Kumar Garg Engineering College, Ghaziabad 201009, India

Correspondence should be addressed to Weimin Zheng; zhengwm901@126.com

Received 30 October 2020; Revised 4 January 2021; Accepted 15 January 2021; Published 31 January 2021

Academic Editor: Shehzad Chaudhry

Copyright © 2021 Tsu-Yang Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of things (IoT) has been widely used for various applications including medical and transportation systems, among others. Smart medical systems have become the most effective and practical solutions to provide users with low-cost, noninvasive, and long-term continuous health monitoring. Recently, Jia et al. proposed an authentication and key agreement scheme for smart medical systems based on fog computing and indicated that it is safe and can withstand a variety of known attacks. Nevertheless, we found that it consists of several flaws, including known session-specific temporary information attacks and lack of per-verification. The opponent can readily recover the session key and user identity. In this paper, we propose a secure authentication and key agreement scheme, which compensates for the imperfections of the previously proposed. For a security evaluation of the proposed authentication scheme, informal security analysis and the Burrows–Abadi–Needham (BAN) logic analysis are implemented. In addition, the ProVerif tool is used to normalize the security verification of the scheme. Finally, the performance comparisons with the former schemes show that the proposed scheme is more applicable and secure.

1. Introduction

A wireless sensor network (WSN) [1–5] (also called sensor network) is a multihop self-organizing network system formed by several inexpensive minisensor nodes distributed in the detection region by wireless communication. The aim of WSN is to gather and process the information of the sensing objects in the network coverage area and transmit it to the observer. The WSN is a significant foundation of the Internet of things and has been used in several fields, such as smart healthcare. Wireless medical sensor networks (WMSNs) [6] can be used to build universal medical systems, which can immediately verify patient emergency situations through the remote monitoring function and can increase the quality of patient medical treatment. In a WSN-based healthcare system, medical sensors are physically

applied on patients, and then the acquired data are forwarded to authorized entities in a secure manner. However, the sensors deployed in the wireless medical sensor network have limited storage and computing capabilities; therefore, when excessive data are collected, the real-time nature of all the data processing may not be guaranteed.

To resolve the aforementioned critical problems, the concept of a fog-driven IoT healthcare system [7–9] (Figure 1) is proposed to move computing functions to users and devices at more remote locations. The fog-driven IoT healthcare system consists of the three following layers: healthcare device layer, medical fog layer, and medical cloud layer. In fog computing [10–16], fog nodes (including routers, gateways, switchers, and access points) are distributed at the margin of the network and approach terminal facilities in a geographic location. By expanding cloud

services to the margin of the network, fog computing transforms cloud data centers into distributed platforms while preserving cloud services for users. Therefore, the waiting time for wireless medical sensor data processing is minimized [17–19], improving user experience and service quality.

Generally, sensor nodes are resource-constrained devices with computing, communication, and storage functions. In addition, sensor nodes are usually distributed in a sparsely populated environment. Because the nodes are vulnerable to threats from adversaries, the security of the deployed equipment cannot be guaranteed. Hence, the security of wireless sensor networks has become a significant challenge for researchers, particularly in WMSN because medical data, security, and privacy issues are more serious considering key patient private information. A few challenges need to be overcome to exploit the entire mechanism and run it efficiently. Maintaining the integrity of the medical data gathered from sensor nodes, providing only legitimate users with secure access to these data, and preventing misuse of data transmitted through public channels are the main challenges that need to be addressed and must be handled carefully. The integrity and confidentiality of data transmitted between the parties must be guaranteed [20].

To establish trust between communication parties and prevent counterfeiting, it is necessary to provide a unique identification [21] and authentication [22] to each user or fog node in the system. In addition, data transmitted through public channels and stored in fog nodes or cloud servers need to be encrypted to ensure data security and privacy [23–25]. However, owing to the mobility of deployed fog nodes and terminal devices, it is not practical to share session keys between them in advance. The authenticated key agreement (AKA) [26–29] is a sufficient scheme for user or node authentication and generating public session keys; however, it is rarely used for fog computing.

Recently, numerous AKA protocols [28–41] have been proposed in WSN, fog computing, and IoT environments. Turkanovic et al. [31] proposed an effective AKA scheme for heterogeneous WSNs, in which the user authenticates through the sensor node without communicating with the gateway node. However, Farash et al. [33] found that their protocol is vulnerable to theft attacks of smart cards and does not provide the untraceability and anonymity of sensor nodes to the user. Wang and Wang [32] indicated that the realization of anonymous authentication cannot be accomplished only through a symmetric cryptographic system. Therefore, it has always focused on designing AKA schemes based on asymmetry. Hayajneh et al. [34] proposed a lightweight authentication scheme based on the Rabin signature, which is used for the remote monitoring of patients by wireless sensor networks. In 2018, Amin et al. [35] proposed a lightweight AKA protocol that is applied to IoT devices in a distributed cloud computing environment. The mutual authentication between the user, service provider, and control server is implemented in their protocol, and a common session key is shared between the user and the server provider. In the scheme indicated above, only a symmetric cryptographic system is used to make the

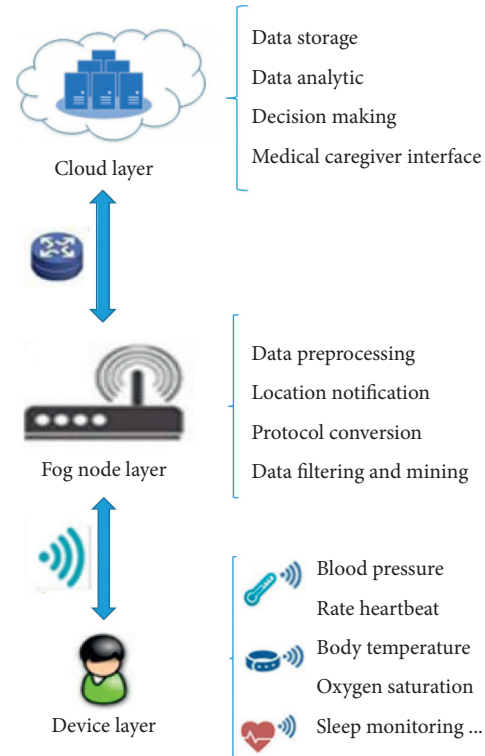


FIGURE 1: The concept of fog-driven IoT healthcare system.

scheme highly efficient. Yeh et al. [30] proposed the first AKA elliptic curve cryptography (ECC) wireless sensor network solution, leading to other researchers proposing an increasing number of ECC-based AKA protocols [36, 41–46].

Although several AKA schemes have been proposed for IoT environments, these protocols are rarely suitable for directly deployed fog computing environments. Hamid et al. [45] proposed a third-party single-round AKA protocol with bilinear pairing for this feature and indicated that it can ensure the privacy of medical data of the fog-based medical system. However, because the session key generated by this scheme is static, it cannot provide forward privacy. The key exchange mechanism of this scheme is based on Joux’s three-party Diffie–Hellman key exchange algorithm [43]; thus, it is also vulnerable to man-in-the-middle attacks. Recently, Jia et al. [46] proposed an AKA scheme for a fog-driven IoT healthcare system using bilinear pairs, in which the cloud server authenticates the IoT device as well as the fog node and generates a shared common session key between them. Based on the Bellare–Rogaway–Pointcheval (BRP) security model [42], they claim that the proposed scheme can resist various known attacks. Informal security analysis also indicates that this scheme retains user anonymity and untractability. Some important related works are summarized in Table 1.

In this study, we first analyzed Jia et al.’s scheme and revealed that it is vulnerable to a random number impersonation attack and key compromise impersonation attack. Then, we proposed an enhancement based on their proposal and remedied the shortcomings of their scheme. In our

TABLE 1: The summary of authentication schemes.

Scheme	Cryptographic techniques	Limitations
Ref. [31]	Smart card	Vulnerable to smart card theft attacks
	One-way hash function	Does not support anonymity Does not support untraceability
Ref. [35]	Symmetric encryption	Does not support anonymity
	One-way hash function	Vulnerable to impersonation attacks
Ref. [36]	Elliptic curve cryptography	Vulnerable to replay attacks
	Bilinear pairing	Does not support mutual authentication
Ref. [46]	One-way hash function	Insecure session key establishment
	Smart card	Does not support anonymity
Ref. [41]	Elliptic curve cryptography	Vulnerable to impersonation attacks
	Bilinear pairing	
	Identity-based cryptography	
	Bilinear pairing	
	One-way hash function	

proposed scheme, the mutual authentication and key agreement between the three entities can be achieved only by one round of communication. After the cloud server verifies the identity of the IoT devices and fog nodes, it generates shared common session keys between them. For a security analysis, we adopted the BAN logic, ProVerif, and an informal security analysis. These approaches can provide evidence indicating that our improvement can resist several well-known security threats.

2. Cryptanalysis of Jia et al.'s AKA Scheme

2.1. Review of Jia et al.'s AKA Scheme. Here, we briefly review the scheme proposed by Jia et al. [46], which mainly consists of the following four phases: system setup, user registration, and fog node registration, as well as authentication and key agreement.

2.1.1. System Setup. The cloud service provider (CSP) selects a nonsingular elliptic curve on the finite field F_p , where p is a large prime number, and $l = \log_2 p$ is the security parameter. Let G be a cyclic group of order n generated by a base point P . Then, CSP selects a random $s \in Z_n^*$ and computes $P_{\text{pub}} = s \cdot P$. (G, P, P_{pub}) are published as the public system parameters, while s remains hidden. Six secure hash functions $\{h_0, h_1, h_2, h_3, h_4, h_5\}$, are selected by CSP, where $h_0: G_1 \rightarrow \{0, 1\}^*$, $h_1: \{0, 1\}^* \times \{0, 1\}^* \rightarrow Z_p^*$, $h_2: \{0, 1\}^* \times Z_p^* \times Z_n^* \rightarrow Z_p^*$, $h_3: G_1 \times Z_p^* \times G_1 \times \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^* \rightarrow Z_p^*$, $h_4: G_1 \times G_1 \times G_1 \times G_1 \times \{0, 1\}^* \times \{0, 1\}^* \rightarrow Z_p^*$, and $h_5: G_2 \times G_1 \times G_1 \times G_1 \rightarrow Z_p^*$. We assume that the CSP is fully trusted and also holds a database to record registered users and fog nodes.

2.1.2. User Registration. U_i inputs respective identity ID_i and password PW_i , and then computes $RID_i = h_1(ID_i \| PW_i) \oplus r_i$, where $r_i \in Z_p^*$ is a random number chosen by U_i . Then, U_i sends (ID_i, RID_i) to CSP via a secure channel. After receiving the U_i request, CSP randomly chooses $x_i \in Z_p^*$ and computes $R_i = h_2(ID_i \| s \| x_i) \oplus RID_i$. The CSP then stores R_i in the smart card and the (ID_i, x_i) in its database and finally sends the smart card to the user over a secure channel. After

the user receives the smart card, U_i calculates $R_i^* = R_i \oplus r_i$ and replaces R_i on the card with R_i^* .

2.1.3. Fog Node Registration. Each fog node F_N must be registered with the CSP before deployment. F_N transmits its identity ID_j to CSP. Then, CSP randomly selects $y_j \in Z_p^*$ and computes $R_j = h_2(ID_j \| s \| y_j)$; CSP sends R_j to the fog node over a secure channel and stores (ID_j, y_j) into its database.

2.1.4. Authentication and Key Agreement. In this phase, CSP can help U_i and F_N to authenticate each other and establish a session key SK after executing the following steps:

- (a) U_i randomly chooses $a \in Z_n^*$ and computes $A = a \cdot P$, $\bar{A} = a \cdot P_{\text{pub}}$, $PID_i = ID_i \oplus h_0(\bar{A})$, $M_i = h_1(ID_i \| PW_i) \oplus R^*$, $|N_i = h_3(\bar{A} \| M_i \| A \| ID_i \| ID_j \| T_u)|$, where T_u is the current timestamp. U_i sends $\text{Msg}_1 = \{A, PID_i, N_i, T_u\}$ to F_N .
- (b) Upon receiving Msg_1 , F_N first checks that the freshness of the timestamp T_u meets the requirements. Then, F_N randomly selects $b \in Z_n^*$ and calculates $B = b \cdot P$, $\bar{B} = b \cdot P_{\text{pub}}$, $PID_j = ID_j \oplus h_0(\bar{B})$, $|L_j = h_3(\bar{B} \| R_j \| A \| PID_j \| ID_j \| T_f)|$, where T_f is the current timestamp. Finally, F_N sends $\text{Msg}_2 = \{A, B, PID_i, PID_j, N_i, L_j, T_u, T_f\}$ to the CSP.
- (c) After receiving Msg_2 , CSP first checks the validity of two timestamps T_u, T_f and then executes the following steps:
 - (i) CSP computes $\bar{A}' = sA$, $\bar{B}' = sB$, $ID_i' = PID_i \oplus h_0(\bar{A}')$, and $ID_j' = PID_j \oplus h_0(\bar{B}')$.
 - (ii) CSP searches its database to find entries that match (ID_i', x_i) and (ID_j', y_j) . If there are no matching entries, CSP denies the request and immediately terminates the session. Otherwise, CSP computes $M_i' = h_2(ID_i' \| s \| x_i)$, $R_j' = h_2(ID_j' \| s \| y_j)$, $N_i' = h_3(\bar{A}' \| M_i' \| A \| ID_i' \| ID_j' \| T_u)$, and $L_j' = h_3(\bar{B}' \| R_j' \| A \| ID_i' \| ID_j' \| T_f)$.

(iii) CSP checks whether $N_i = N'_i$ and $L_j = L'_j$. If one of these equations is not true, the CSP rejects the request and terminates. Otherwise, it randomly chooses $c \in Z_n^*$ and computes $C = c \cdot P$. $\text{Auth}_i = h_4(A \| B \| C \| \overline{A}' \| \text{ID}_i \| T_c)$, $\text{Auth}_j = h_4(A \| B \| C \| \overline{B}' \| \text{ID}_j \| T_c)$, $K_c = e(A, B)^C$, and $\text{SK}_c = h_5(K_c \| A \| B \| C)$; note, the current timestamp is T_c . Finally, CSP forwards $\text{Msg}_3 = \{C, \text{Auth}_i, \text{Auth}_j, T_c\}$ to FN_j .

- (d) Upon receiving Msg_3 , F_N checks the freshness of T_c and verifies whether $\text{Auth}_j = h_4(A \| B \| C \| \overline{B}' \| \text{ID}_j \| T_c)$. If the equation is not true, F_N terminates the session. Otherwise, F_N calculates $\text{SK}_f = h_5(K_f \| A \| B \| C)$, where $K_f = e(A, C)^b$. Then, F_N sends $\text{Msg}_4 = \{B, C, \text{Auth}_i, T_c\}$ to U_i .
- (e) Upon receiving Msg_4 , U_i checks the freshness of T_c and verifies whether $\text{Auth}_i = h_4(A \| B \| C \| \overline{A}' \| \text{ID}_i \| T_c)$. If not, U_i aborts the session. Otherwise, U_i computes $\text{SK}_u = h_5(K_u \| A \| B \| C)$, where $K_u = e(B, C)^d$.

2.2. Security Weakness of Jia et al.'s Scheme

2.2.1. Known Session-Specific Temporary Information Attack. Here, we demonstrate that Jia et al.'s scheme suffered from a known session-specific temporary information attack. This attack is indicated in Canetti and Krawczyk's (CK) adversary model [47]. We allow an attacker E to fully control the communications over the user, fog node, and CSP for "authentication and key agreement phase." Thus, E can intercept the messages and obtain the hidden information of a current session from either side over a public channel, which enabled the recovery of key information from the session, such as the session key and the entity's identity.

- (a) *Session key recovery.* Based on the CK adversarial model, we may assume that an attacker E can obtain a random number a of users U_i . Note, E can also be intercepted $\{A, B, \text{PID}_i, \text{PID}_j, N_i, L_j, T_u, T_f, C, \text{Auth}_i, \text{Auth}_j\}$ in the open channel. Then, E can compute $\text{SK}_u = h_5(A \| B \| C \| K_u)$, where $K_u = e(B, C)^a$. Note, we may assume that E can obtain b or c from F_N and CSP. The session key SK can also be computed by $e(A, C)^b$ and $e(A, B)^c$ because $\text{SK} = e(B, C)^d = e(A, C)^b = e(A, B)^c$ in Jia et al.'s scheme; note, a , b , and c are random numbers chosen by U_i , F_N , and CSP, respectively.
- (b) *Identity recovery (anonymity violation).* By the same assumption in (a), E can recover the U_i identity $\text{ID}_i = \text{PID}_i \oplus h_0(\overline{A}^*)$, where $\overline{A}^* = a \cdot P_{\text{pub}}$. Similarly, E can recover $\text{ID}_j = \text{PID}_j \oplus h_0(\overline{B}^*)$, where $\overline{B}^* = b \cdot P_{\text{pub}}$, while E obtains the F_N random value b .

2.2.2. Lack of Per-Verification. Step (a) of the authentication and key agreement phase lacks verifying the user input ID_i

and PW_i . This will increase the redundant computational cost, while the user inputs an incorrect ID_i or PW_i . The incorrect input will be identified by CSP in step (c) of the authentication and key agreement phase.

3. Our Improved Scheme

In this section, we propose an improvement based on Jia et al.'s scheme to overcome the previously indicated security weaknesses in Section 2. In our improvement, the system setup is the same as in Jia et al.'s scheme.

3.1. Modified User Registration. This phase is depicted in Figure 2.

- (a) U_i randomly chooses $r_i \in Z_p^*$, inputs the password PW_i and the identity ID_i to compute $\text{RID}_i = h_1(\text{ID}_i \| \text{PW}_i) \oplus r_i$. Then, U_i sends $(\text{ID}_i, \text{RID}_i)$ to CSP via a secure channel.
- (b) After receiving $(\text{ID}_i, \text{RID}_i)$, CSP randomly chooses $x_i \in Z_p^*$ and computes $q_i = h_2(\text{ID}_i \| s \| x_i)$, $R_i = q_i \oplus \text{RID}_i$, $D_i = h_2(q_i \| \text{ID}_i) \oplus \text{RID}_i$. The CSP then stores (R_i, D_i) in the smart card and the (ID_i, x_i) in its own database and finally sends the smart card to the user over a secure channel.
- (c) After the user receives the smart card, U_i calculates $R_i^* = R_i \oplus r_i$, $V_i = D_i \oplus r_i$, and replaces R_i, D_i with R_i^* and V_i .

3.2. Modified Fog Node Registration. F_N transmits its identity ID_j to the CSP. It randomly selects $y_j \in Z_p^*$ and computes $g_j = h_2(\text{ID}_j \| s \| y_j)$. Then, CSP sends g_j to the fog node via a secure channel and stores (ID_j, y_j) in its own database. This phase is shown in Figure 3.

3.3. Modified Authentication and Key Agreement. This phase is depicted in Figure 4.

- (a) U_i inputs ID_i and PW_i and computes $q_i = R_i^* \oplus h_1(\text{ID}_i \| \text{PW}_i)$, $V_i^* = h_2(q_i \| \text{ID}_i) \oplus h_1(\text{ID}_i \| \text{PW}_i)$. Then, whether $V_i^* = V_i$ is checked. If the equation is true, U_i randomly chooses $a \in Z_n^*$ and computes $v_u = a \cdot q_i$, $A = v_u \cdot P$, $\overline{A} = v_u \cdot P_{\text{pub}}$, $\text{PID}_i = \text{ID}_i \oplus h_0(\overline{A})$, $M_i = q_i = h_1(\text{ID}_i \| \text{PW}_i) \oplus R_i^*$, $N_i = h_3(\overline{A} \| M_i \| A \| \text{ID}_i \| \text{ID}_j \| T_u)$, where T_u is the current timestamp. Finally, U_i sends $\text{Msg}_1 = \{A, \text{PID}_i, N_i, T_u\}$ to F_N .
- (b) Upon receiving Msg_1 , F_N first checks that the freshness of the timestamp T_u meets the requirements. Then, it randomly selects $b \in Z_n^*$ and calculates $v_f = b \cdot g_j$, $B = v_f \cdot P$, $\overline{B} = v_f \cdot P_{\text{pub}}$, $\text{PID}_j = \text{ID}_j \oplus h_0(\overline{B})$, $L_j = h_3(\overline{B} \| R_j \| A \| \text{PID}_j \| \text{ID}_j \| T_f)$, where T_f is the current timestamp. Finally, F_N forwards $\text{Msg}_2 = \{A, B, \text{PID}_i, \text{PID}_j, N_i, L_j, T_u, T_f\}$ to the CSP.

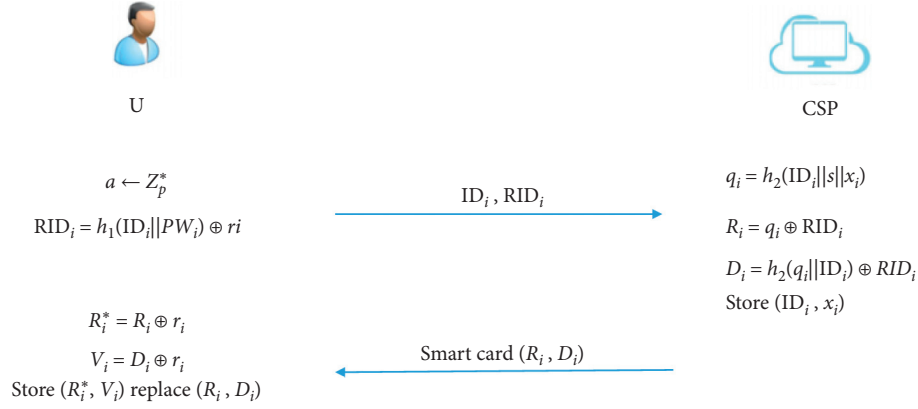


FIGURE 2: Modified user registration phase.

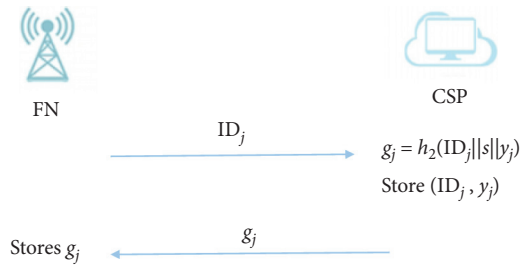


FIGURE 3: Modified fog node registration phase.

(c) After receiving Msg_2 , CSP first checks the validity of two timestamps T_u , T_f and then executes the following steps:

- (i) To compute $\bar{A}' = s \cdot A$, $\bar{B}' = s \cdot B$, $ID_i' = PID_i \oplus h_0(\bar{A}')$, $ID_j' = PID_j \oplus h_0(\bar{B}')$ and then searches for (ID_i', x_i) and (ID_j', y_j) in its database. If there are no matching entries, CSP denies the request and immediately terminates the session.
 - (ii) To compute $M_i' = h_2(ID_i' || s || x_i)$, $R_j' = h_2(ID_j' || s || y_j)$, $N_i' = h_3(\bar{A}' || M_i' || A || ID_i' || ID_j' || T_u)$, and $L_j' = h_3(\bar{B}' || R_j' || A || PID_i' || ID_j' || T_f)$. Then, it checks whether $N_i = N_i'$ and $L_j = L_j'$. If one of these equations is not true, the CSP rejects the request and terminates.
 - (iii) CSP randomly chooses $c \in Z_n^*$ and computes $z_c = h_2(y_j || s || x_i)$, $v_c = c \cdot z_c$, $C = v_c \cdot P$, $Auth_i = h_4(A || B || C || \bar{A}' || ID_i' || T_c)$, $Auth_j = h_4(A || B || C || \bar{B}' || ID_j' || T_c)$, $K_c = e(A, B)^{v_c}$, $SK_c = h_5(K_c || A || B || C)$, where T_c is the current timestamp. Finally, CSP sends $Msg_3 = \{C, Auth_i, Auth_j, T_c\}$ to F_N .
- (d) Upon receiving $Msg_3 = \{C, Auth_i, Auth_j, T_c\}$, F_N checks the freshness of T_c and verifies whether $Auth_j = h_4(A || B || C || \bar{B}' || ID_j' || T_c)$. If the equation is not true, then F_N immediately terminates the session. Otherwise, F_N calculates $K_f = e(A, C)^{v_f}$, $SK_f = h_5$

$(K_f || A || B || C)$, and forwards $Msg_4 = \{B, C, Auth_i, T_c\}$ to U_i .

- (e) Upon receiving Msg_4 , U_i checks the freshness of T_c and verifies if $Auth_i = h_4(A || B || C || \bar{A}' || ID_i' || T_c)$. If the equation is not true, U_i immediately terminates the session. Otherwise, U_i calculates $K_u = e(A, C)^{v_u}$, $SK_u = h_5(K_u || A || B || C)$.

4. Security Analysis of Our Improved Scheme

In this section, the security of our scheme is illustrated by the BAN logic, ProVerif, and an informal security analysis.

4.1. Formal Security Analysis Using BAN Logic. In this subsection, the sharing session SK calculated by CSP between U_i , F_N , and CSP is presented, which can be used to send request information to the server when the user wants to obtain data from the server. Note, the following notations and rules for the BAN logic can be found in previous studies [33, 35, 39, 48].

4.1.1. Related Rules

Messages meaning rule ($A \equiv A \xleftrightarrow{K} B, A \triangleleft \langle X \rangle K / A \equiv B \sim X$): if principal A believes that hidden K value is shared between principals A and B , and A receives the message X enciphered with K and then A believes that B is the sender of X .

Nonce verification rule ($A \equiv \#(X), A \equiv B \sim X / A \equiv B \equiv X$): if A believes that message X is fresh and that B has sent X , then A believes that B also believes in message X .

Jurisdiction rule ($A \equiv B \Rightarrow X, A \equiv B \equiv X / A \equiv X$): if A believes that B has jurisdiction over X and that B believes on statement X , then A believes on X .

Session key introduction rule $A \equiv \#(X), A \equiv B \equiv X / A \equiv A \xleftrightarrow{K} B$: if A believes that message X is fresh and that B also believes on X , then A believes they share the session key.

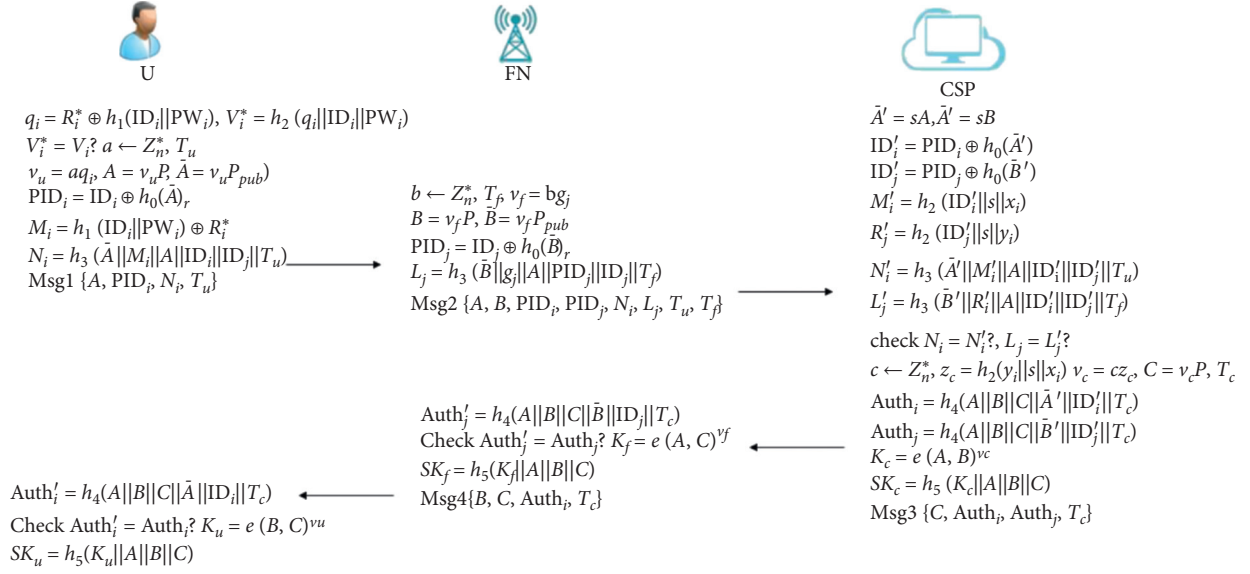


FIGURE 4: Modified authentication and key agreement phase.

Belief rule $(A | \equiv B | \equiv (XY)/A | \equiv B | \equiv X)$: if A believes that B believes formula (X, Y) , then A believes that B also believes the X or Y part of the formula.

4.1.2. Goals

- GOAL 1: $U_i | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$
- GOAL 2: $U_i | \equiv F_N | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$
- GOAL 3: $F_N | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$
- GOAL 4: $F_N | \equiv U_i | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$
- GOAL 5: $\text{CSP} | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$
- GOAL 6: $\text{CSP} | \equiv U_i | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$
- GOAL 7: $\text{CSP} | \equiv F_N | \equiv (U_i \xleftrightarrow{\text{SK}} F_N)$

4.1.3. Idealize the Communication Messages

- Msg1 $U_i \longrightarrow F_N: \{A, \text{PID}_i, N_i, T_u\}$
- Msg2 $F_N \longrightarrow \text{CSP}: \{B, \text{PID}_j, L_j, T_f, A, \text{PID}_i, N_i, T_u\}$
- Msg3 $U_i \longrightarrow \text{CSP}: \{A, \text{PID}_i, N_i, T_u\}$
- Msg4 $\text{CSP} \longrightarrow F_N: \{\text{Auth}_j, C, T_c\}$
- Msg5 $\text{CSP} \longrightarrow U_i: \{\text{Auth}_i, C, T_c\}$
- Msg6 $F_N \longrightarrow U_i: \{B, \text{Auth}_i, C, T_c\}$

4.1.4. Initial State Assumptions

- A1: $U_i | \equiv \#(a)$
- A2: $U_i | \equiv \#(A)$
- A3: $U_i | \equiv (A)$
- A4: $U_i | \equiv \#(B)$
- A5: $U_i | \equiv \#(C)$
- A6: $F_N | \equiv \#(b)$
- A7: $F_N | \equiv \#(B)$

- A8: $F_N | \equiv (B)$
- A9: $F_N | \equiv \#(A)$
- A10: $F_N | \equiv \#(C)$
- A11: $\text{CSP} | \equiv \#(c)$
- A12: $\text{CSP} | \equiv \#(C)$
- A13: $\text{CSP} | \equiv (C)$
- A14: $\text{CSP} | \equiv \#(A)$
- A15: $\text{CSP} | \equiv \#(B)$
- A16: $U_i | \equiv U_i \xleftrightarrow{\text{ID}_j} F_N$
- A17: $U_i | \equiv U_i \xleftrightarrow{(\text{ID}_i, \text{PID}_i, R_i)} \text{CSP}$
- A 18: $U_i | \equiv F_N = > B$
- A 19: $U_i | \equiv \text{CSP} = > C$
- A 20: $F_N | \equiv U_i \xleftrightarrow{\text{ID}_j} F_N$
- A 21: $F_N | \equiv F_N \xleftrightarrow{(\text{ID}_j, g_j)} \text{CSP}$
- A 22: $F_N | \equiv U_i = > A$
- A 23: $F_N | \equiv \text{CSP} = > C$
- A 24: $\text{CSP} | \equiv U_i \xleftrightarrow{(\text{ID}_i, \text{PID}_i, R_i)} \text{CSP} \alpha$
- A 25: $\text{CSP} | \equiv F_N \xleftrightarrow{(\text{ID}_j, g_j)} \text{CSP}$
- A 26: $\text{CSP} | \equiv F_N = > B$
- A 27: $\text{CSP} | \equiv U_i = > A$

If a is a random number chosen by U_i , we can obtain A1 and A2; when Msg1 sends form U_i to F_N , A22 is obtained. From A22, we obtain A9; when Msg3 sends form U_i to CSP, we obtain A27. From A27, we obtain A14. Similarly, because b is a random number chosen by F_N , we obtain A6 and A7; when Msg6 sends from F_N to U_i , we obtain A18. From A18, we obtain A4; when Msg2 sends from F_N to CSP, we obtain A26. From A26, we obtain A15. c is a random number chosen by CSP; we obtain A26 and A27; when Msg5 sends from CSP to U_i , we obtain A19. From A19, we obtain A5;

when Msg4 sends from CSP to F_N , we obtain A23. From A23, we obtain A10.

4.1.5. Main Proofs Using BAN Rules and Assumptions

(1) For GOAL 1 and GOAL 2. From message Msg6 and using the seeing rule, we obtain S1: $U_i \triangleleft \{B, \text{Auth}_i, C, T_c\}$. Using the seeing rule, we obtain S2: $U_i \triangleleft \{B\}$. Using A16, S2, and the message meaning rule, we obtain S3: $U_i | \equiv F_N | \sim \{B\}$. Using A4, S3, and the nonce verification rule, we obtain S4: $U_i | \equiv F_N | \equiv B$. Using A18, S4, and the jurisdiction rule, we obtain S5: $U_i | \equiv B$. Based on message Msg5 and the seeing rule, we obtain S6: $U_i \triangleleft \{\text{Auth}_i, C, T_c\}$. Using the seeing rule, we obtain S7: $U_i \triangleleft \{C\}$. According to A17, S7, and the message meaning rule, we have S8: $U_i | \equiv \text{CSP} | \sim \{C\}$. Using A5, S8, and the nonce verification rule, we obtain S9: $U_i | \equiv \text{CSP} | \equiv C$. Using A19, S9, and the jurisdiction rule, we obtain S10: $U_i | \equiv C$. Based on A2, A4, A5, A3, S5, S10, and the belief rule, we obtain S11: $U_i | \equiv \#(A, B, C)$ and S12: $U_i | \equiv (A, B, C)$. Because $A = v_u P, B = v_f P, C = v_c P$, we can obtain S13: $U_i | \equiv (v_u, v_f, v_c)$. Because $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, $\text{SK}_u = \text{SK}_f = \text{SK}_c = h_5(K_u || A || B || C) = h_5(K_f || A || B || C) = h_5(K_c || A || B || C)$. Using A2, A16_{SK}, S12, S13, and the belief rule, we obtain S14: $U_i | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 1).

Using A2, S14, and the session key introduction rule, we obtain S15: $U_i | \equiv F_N | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 2).

(2) For GOAL 3 and GOAL 4. From message Msg1 and using the seeing rule, we obtain S16: $F_N \triangleleft \{A, \text{PID}_i, N_i, T_u\}$. Using the seeing rule, we obtain S17: $F_N \triangleleft \{A\}$. According to A20, S17, and the message meaning rule, we have S18: $F_N | \equiv U_i | \sim \{A\}$. Employing A9, S18, and the nonce verification rule, we obtain S19: $F_N | \equiv U_i | \equiv A$. Using A22, S19, and the jurisdiction rule, we have S20: $F_N | \equiv A$. From message Msg4 and using the seeing rule, we have S21: $F_N \triangleleft \{\text{Auth}_i, C, T_c\}$. We obtain S22: $F_N \triangleleft \{C\}$ via the seeing rule. According to A21, S22, and the message meaning rule, we obtain S23: $F_N | \equiv \text{CSP} | \sim \{C\}$. Using A10, S23, and the nonce verification rule, we obtain S24: $F_N | \equiv \text{CSP} | \equiv C$. According to A23, S24, and the jurisdiction rule, we have S25: $F_N | \equiv C$. According to A7, A10, A9, A8, S20, S25, and the belief rule, we obtain S26: $F_N | \equiv \#(A, B, C)$ and S27: $F_N | \equiv (A, B, C)$. Because $A = v_u P, B = v_f P, C = v_c P$, we can obtain S28: $F_N | \equiv (v_u, v_f, v_c)$. Using A7, A20, S27, S28, and the belief rule, we obtain S29: $F_N | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 3).

By using A7, S29, and the session key introduction rule, we obtain S30: $F_N | \equiv U_i | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 4).

(3) For GOAL 5, GOAL 6, and GOAL 7. According to Msg2 and using the seeing rule, we obtain S31: $\text{CSP} \triangleleft \{B, \text{PID}_j, L_j, T_f, A, \text{PID}_i, N_i, T_u\}$. Using the seeing rule, we obtain S32: $\text{CSP} \triangleleft \{B\}$. Using A25, S32, and the message meaning rule, we obtain S33: $\text{CSP} | \equiv F_N | \sim \{B\}$. Using A15, S33, and the nonce verification rule, we obtain S34: $\text{CSP} | \equiv F_N | \equiv B$. Using A26, S34, and the jurisdiction

rule, we obtain S35: $\text{CSP} | \equiv B$. Based on Msg3 and the seeing rule, we obtain S36: $\text{CSP} \triangleleft \{A, \text{PID}_i, N_i, T_u\}$. We have S37: $\text{CSP} \triangleleft \{A\}$ via the seeing rule. According to A24, S37, and the message meaning rule, we obtain S38: $\text{CSP} | \equiv U_i | \sim \{A\}$. Using A14, S38, and the nonce verification rule, we obtain S39: $\text{CSP} | \equiv U_i | \equiv A$. According to A27, S39, and the jurisdiction rule, we obtain S40: $\text{CSP} | \equiv A$. According to A14, A12, A15, A13, S35, S40, and the belief rule, we obtain S41: $\text{CSP} | \equiv \#(A, B, C)$ and S42: $\text{CSP} | \equiv (A, B, C)$. Because $A = v_u P, B = v_f P, C = v_c P$, we can obtain S43: $U_i | \equiv (v_u, v_f, v_c)$. Using A12, S42, S43, and the belief rule, we obtain S44: $\text{CSP} | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 5).

Using A14, S44, and the session key introduction rule, we obtain S45: $\text{CSP} | \equiv U_i | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 6).

Using A15, S44, and the session key introduction rule, we obtain S46: $\text{CSP} | \equiv F_N | \equiv (U_i \stackrel{\text{SK}}{\leftrightarrow} F_N)$ (GOAL 7).

4.2. Informal Security Analysis. In this section, we demonstrate that our improved scheme can achieve the following well-known security requirements.

4.2.1. Known Session-Specific Temporary Information Attacks. The session key $\text{SK}_u = \text{SK}_f = \text{SK}_c = h_5(K_u || A || B || C) = h_5(K_f || A || B || C) = h_5(K_c || A || B || C)$ is generated utilizing the hidden values of $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, and $v_u = aq_i, v_f = bg_j, v_c = cz_c$; (A, B, C) can be intercepted on an open channel, but adversaries do not know (q_i, g_j, z_c) because they are the hidden values of U_i, F_N , and CSP, respectively, and, thus, cannot calculate (v_u, v_f, v_c) . Therefore, despite adversaries determining (a, b, c) , they cannot calculate (K_u, K_f, K_c) without (q_i, g_j, z_c) . Therefore, an opponent cannot recover SK using temporarily leaked session-specific information $\{a, b, c\}$.

(q_i, g_j) are the hidden values of U_i , and F_N , respectively; if only (a, b) is found, but not (q_i, g_j) , the adversaries cannot calculate $v_u = aq_i, v_f = bg_j$. $\bar{A} = v_u P_{\text{pub}}, \bar{B} = v_f P_{\text{pub}}, \text{PID}_i = \text{ID}_i \oplus h_0(\bar{A}), \text{PID}_j = \text{ID}_j \oplus h_0(\bar{B}); (\text{PID}_i, \text{PID}_j)$ can be intercepted on an open channel, but adversaries cannot retrieve $\text{ID}_i = \text{PID}_i \oplus h_0(\bar{A})$ and $\text{ID}_j = \text{PID}_j \oplus h_0(\bar{B})$ without (v_u, v_f) . If adversaries intercept (A, B) on an open channel, they do not know the key s of the CSP and, thus, cannot calculate $\bar{A}' = sA$ and $\bar{B}' = sB$, or retrieve $\text{ID}_i = \text{PID}_i \oplus h_0(\bar{A}'), \text{ID}_j = \text{PID}_j \oplus h_0(\bar{B}')$ without s .

4.2.2. Mutual Authentication. CSP authenticates U_i by verifying whether ID'_i equals to the ID_i saved in the CSP database and whether N'_i equals to N_i, N_i sent from U_i . U_i authenticates CSP by verifying whether Auth'_i equals to $\text{Auth}_i = h_4(A || B || C || \bar{A}' || \text{ID}'_i || T_c)$, which includes C calculated by CSP.

Similarly, CSP authenticates F_N by verifying whether ID'_j equals to the ID_j saved in the CSP database and whether L'_j equals L_j, L_j sent from F_N . F_N authenticates CSP by verifying whether Auth'_j equals to $\text{Auth}_j = h_4(A || B || C || \bar{B}' || \text{ID}'_j || T_c)$, which includes C calculated by CSP.

F_N authenticates U_i by verifying whether Auth'_j equals to Auth_j which includes A calculated by U_i , and U_i authenticates

F_N by verifying whether Auth'_i equals to $\text{Auth}_i = h_4(A \| B \| C \| \bar{A}' \| \text{ID}'_i \| T_c)$, which includes B calculated by F_N .

4.2.3. Impersonation Attack. To impersonate a legitimate user, the adversary has to obtain the identity ID_i , password PW_i , and $q_i = h_2(\text{ID}_i \| s \| x_i)$ of U_i or construct $A = v_u P$, $\text{PID}_i = \text{ID}_i \oplus h_0(\bar{A})$, and $N_i = h_3(\bar{A} \| M_i \| A \| \text{ID}_i \| \text{ID}_j \| T_u)$. First, the opponent is unable to guess the correct identity and password of U_i through “password-guessing attack.” Second, to construct $\{A, \text{PID}_i, N_i\}$, the adversary has to obtain the key s and parameter x_i . However, it cannot compute q_i without ID_i , s , and x_i , which are crucial for computing $\{A, \text{PID}_i, N_i\}$. Thus, the adversary cannot impersonate a legitimate user.

Similarly, to mimic a legitimate fog node, the opponent must obtain the identity ID_j and $q_j = h_2(\text{ID}_j \| s \| x_j)$ of F_N or construct $B = v_f P$, $\text{PID}_j = \text{ID}_j \oplus h_0(\bar{B})$, and $L_j = h_3(\bar{B} \| g_j \| A \| \text{ID}_j \| \text{ID}_i \| T_f)$; the adversary can obtain the identity ID_j , but it is impossible for the adversary to determine $g_j = h_2(\text{ID}_j \| s \| y_j)$, which is computed and assigned by CSP in F_N registration. g_j cannot be computed without s and y_j , which are crucial for computing $\{B, \text{PID}_j, L_j\}$. Thus, the adversary cannot impersonate a legitimate F_N .

The adversary is also unable to impersonate CSP. To compute $C = v_c P$, $\text{Auth}_i = h_4(A \| B \| C \| \bar{A}' \| \text{ID}'_i \| T_c)$, and $\text{Auth}_j = h_4(A \| B \| C \| \bar{B}' \| \text{ID}'_j \| T_c)$, s , x_i , and y_j are required to compute $C = v_c P = h_2(y_i \| s \| x_i) c P$. However, the adversary cannot obtain C unless it obtains all three factors at the same time. This is beyond the capacity of an adversary. Thus, the adversary cannot impersonate CSP.

4.2.4. Man-in-the-Middle Attacks. If the adversary obtains Msg1 or Msg2 from the public channel and modifies Msg1 or Msg2 to launch a man-in-the-middle attack, the identity authentication of CSP cannot be passed; the premise of the authentication of CSP is to determine the identity of U_i and F_N . From “(2),” we know that CSP will compute ID'_i and ID'_j and compare the values with ID_i and ID_j saved in the CSP database; if it is not equal, the session will immediately be terminated. From “(1),” we know that the adversary cannot obtain ID_i and ID_j . Meanwhile, from “(3),” we also know that the adversary cannot obtain the values of s , x_i , and y_j . Thus, the modified messages cannot pass the verification of $N'_i = N_i$ and $L'_j = L_j$ from CSP.

If the adversary obtains Msg3 or Msg4 from the open channel and modifies Msg3 or Msg4 to launch the man-in-the-middle attack, the authentication from U_i and F_N will still not be passed. As indicated by “(2),” we can see that if the messages are modified by the adversary, they cannot pass the verification of $\text{Auth}'_i = \text{Auth}_i$ and $\text{Auth}'_j = \text{Auth}_j$ from U_i and F_N .

4.2.5. Known Session Key Attacks. A scheme is considered vulnerable to known session key attacks if an adversary wants to use the old compromised session key to obtain sensitive parameters and keys for subsequent

communication sessions. In our scheme, $\text{SK}_u = \text{SK}_f = \text{SK}_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$, $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, $A = v_u P$, $B = v_f P$, $C = v_c P$, $v_u = a q_i$, $v_f = b g_j$, $v_c = c z_c$, is refreshed using random numbers $\{a, b, c\}$ and the attacker does not know $\{q_i, g_j, z_c\}$. Thus, owing to the computational difficulty of the elliptic curve Diffie–Hellman problem, it is impossible for the attacker to obtain the new SK information from the old SK and extract $\{a, b, c\}$ from $\{A, B, C\}$; thus, the scheme we proposed can withstand the known session key attack.

4.2.6. Compromise Impersonation Attacks. If the CSP long-term key s is compromised, the adversary may use s to impersonate a legitimate user to determine F_N and CSP. However, all attack sessions are terminated immediately, as follows. In a worst case scenario, the adversary may have access to the data $R_i^* = h_2(\text{ID}_i \| s \| x_i) \oplus h_1(\text{ID}_i \| \text{PW}_i)$, $V_i = h_2(h_2(\text{ID}_i \| s \| x_i) \| \text{ID}_i) \oplus h_1(\text{ID}_i \| \text{PW}_i)$, in the stolen smart card SC. Despite knowing s , the adversary does not know the hidden values of $\{\text{ID}_i, x_i, \text{PW}_i\}$ to compute $h_2(\text{ID}_i \| s \| x_i) = R_i^* \oplus h_1(\text{ID}_i \| \text{PW}_i)$ or $q_i = h_2(\text{ID}_i \| s \| x_i)$ directly. Thus, the adversary cannot generate $\text{Msg1} = \langle A, \text{PID}_i, N_i, T_u \rangle$ to masquerade U_i to launch a new session.

The adversary may intercept messages sent by U_i during authentication and key negotiation and attempt to impersonate the initiator of the session. However, the session will terminate immediately because the attacker cannot calculate $K_u = e(B, C)^{v_u}$ correctly without knowing the hidden values of $\{a, q_i\}$, despite knowing s .

4.2.7. Parallel Session Attacks. When the entity is in session, the adversary may try to replay the old messages to launch a new session attack; however, this is impossible. When an attacker replays $\{\text{M1}, \text{M2}\}$ to CSP, it can pass the verification of $N'_i = h_3(\bar{A}' \| M'_i \| A \| \text{ID}'_i \| \text{ID}'_j \| T_u)$, $L'_j = h_3(\bar{B}' \| R'_j \| A \| \text{ID}'_i \| \text{ID}'_j \| T_f)$. However, because the attacker does not know $\{a, b\}$ and $\{q_i, g_j\}$, it cannot compute one of $K_u = h_5(K_u \| A \| B \| C)$, $K_f = h_5(K_f \| A \| B \| C)$, $K_u = K_f = e(B, C)^{v_u} = e(A, C)^{v_f}$, and $v_u = a q_i$, $v_f = b g_j$, and the attacker session is immediately aborted.

4.2.8. Stolen Smart Card Attacks. If an attacker steals the smart card and extracts $R_i^* = h_2(\text{ID}_i \| s \| x_i) \oplus h_1(\text{ID}_i \| \text{PW}_i)$, $V_i = h_2(h_2(\text{ID}_i \| s \| x_i) \| \text{ID}_i) \oplus h_1(\text{ID}_i \| \text{PW}_i)$, he/she may impersonate U_i to F_N and CSP. However, the attacker does not know the sensitive parameter $\{\text{ID}_i, \text{PW}_i, x_i, s\}$ to generate the initiator message $\text{PID}_i = \text{ID}_i \oplus h_0(\bar{A})$, $N_i = h_3(\bar{A} \| M_i \| A \| \text{ID}_i \| \text{ID}_j \| T_u)$, thus cannot impersonate U_i to F_N and CSP. Hence, the proposed scheme can withstand stolen smart card attacks.

4.2.9. Password-Guessing Attacks. If an adversary obtains information regarding $\{A, B, \text{PID}_i, \text{PID}_j, N_i, L_j, C, \text{Auth}_i, \text{Auth}_j, T_u, T_f, T_c\}$ from the open channel, online password-guessing attacks may be launched. However, the adversary

will fail because $A = v_u P$, $PID_i = ID_i \oplus h_0(\bar{A})$, $N_i = h_3(\bar{A} \| M_i \| A \| ID_i \| ID_j \| T_u)$, $B = v_f P$, $PID_j = ID_j \oplus h_0(\bar{B})$, $L_j = h_3(\bar{B} \| g_j \| A \| PID_j \| ID_j \| T_f)$, $C = v_c P$, $Auth_i = h_4(A \| B \| C \| \bar{A}' \| ID_i' \| T_c)$, $Auth_j = h_4(A \| B \| C \| \bar{B}' \| ID_j' \| T_c)$, and PW_i are not included in these values. Therefore, PW_i remains secure.

If the smart card is compromised by an opponent, the parameter $\{R_i^*, V_i\}$ in the SC can be obtained through the power analysis attack method, and then off-line dictionary attacks can be made based on the relevant parameter $R_i^* = h_2(ID_i \| s \| x_i) \oplus h_1(ID_i \| PW_i)$, $V_i = h_2(h_2(ID_i \| s \| x_i) \| ID_i) \oplus h_1(ID_i \| PW_i)$, to guess the user password. However, because the values $\{x_i, s\}$ are only known by the CSP, the opponent cannot verify the accuracy of the guess value; therefore, all sensitive parameters are safe.

4.2.10. Privileged-Insider Attacks. When the attacker obtains U_i registration information (ID_i , RID_i , x_i) and the key s of CSP, the intent is to compute the session key $SK_u = SK_f = SK_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$, which is randomized using $\{a, b, c\}$ and $\{q_i, g_j, z_c\}$. By $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$ and $v_u = aq_i$, $v_f = bg_j$, $v_c = cz_c$, the attacker can compute $q_i = h_2(ID_i \| s \| x_i)$ and obtain (A, B, C) from the public channel. However, (a, b, c) are random numbers independently selected by U_i , F_N , and CSP, respectively, and are not available to the attacker; therefore, v_u and SK_u cannot be computed.

Similarly, when the attacker obtains the F_N registration information (ID_j , y_j) and the key s of CSP, the intent is to compute the session key SK_f ; the attacker can compute $g_j = h_2(ID_j \| s \| y_j)$ and obtain (A, B, C) from the public channel. However, (a, b, c) are random numbers independently selected by U_i , F_N , and CSP, respectively, and are not available to the attacker; therefore, v_f and SK_f cannot be computed.

The attacker also cannot compute SK_c ; $z_c = h_2(y_i \| s \| x_i)$ can be computed, but $v_c = cz_c$ cannot be computed without the c selected by CSP. Thus, the modified scheme can withstand privileged-insider attacks.

4.2.11. Replay Attacks. The adversary may attempt to replay old messages $\{Msg1, Msg2, Msg3, \text{ and } Msg4\}$. However, all communicated messages are refreshed and rely on the timestamp $\{T_u, T_f, T_c\}$ as well as random numbers $\{a, b, c\}$. Upon receiving the authentication request from the sender, the receiver first checks the freshness of the timestamp. If the timestamp is not fresh, the session is terminated immediately.

4.2.12. Perfect Forward Secrecy. Perfect forward secrecy indicates that if a long-term key is revealed to an attacker, the SK between U_i , F_N , and CSP, cannot be computed and remains secure. If an attacker attempts to calculate the session key, $SK_u = SK_f = SK_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$, which is randomized using numbers $\{a, b, c\}$ and $\{q_i, g_j, z_c\}$;

$K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, $v_u = aq_i$, $v_f = bg_j$, $v_c = cz_c$. The attacker obtains (A, B, C) from the public channel; however, the attacker needs to compute one of the parameters v_u, v_f, v_c , which cannot be obtained, thus SK cannot be calculated. Therefore, the improved scheme can provide perfect forward secrecy.

4.2.13. No Key Control. Each entity cannot control the key agreement process to calculate SK individually, where $SK_u = SK_f = SK_c = h_5(K_u \| A \| B \| C) = h_5(K_f \| A \| B \| C) = h_5(K_c \| A \| B \| C)$, $K_u = K_f = K_c = e(B, C)^{v_u} = e(A, C)^{v_f} = e(A, B)^{v_c}$, and $v_u = aq_i$, $v_f = bg_j$, $v_c = cz_c$, $A = v_u P$, $B = v_f P$, $C = v_c P$. The details are as follows:

(a, b, c) are random numbers independently selected by U_i , F_N , and CSP, respectively, and (A, B, C) are computed independently by each entity. If U_i does not know the values of B and C , which are contributed by F_N and CSP, SK_u cannot be computed. Similarly, F_N and CSP cannot compute SK_f and SK_c without the values of (A, C) and (A, B) .

4.2.14. Unknown Key-Share. From “(2),” we know that all three entities are mutually identifiable. If U_i and entity-1 establish the session key and send the request message of entity-1 by mistake to entity-2, it is impossible to pass the validation $ID_i' = ID_i$, $ID_j' = ID_j$, $N_i = h_3(\bar{A} \| M_i \| A \| ID_i \| ID_j \| T_u) = N_i' = h_3(\bar{A}' \| M_i' \| A \| ID_i' \| ID_j' \| T_u)$, and $L_j = h_3(\bar{B} \| g_j \| A \| PID_j \| ID_j \| T_f) = L_j' = h_3(\bar{B}' \| R_j' \| A \| ID_j' \| T_f)$, thus the session terminates immediately. Therefore, the proposed scheme can resist unknown key-share attacks.

4.3. Evaluation by ProVerif. In this section, we choose the widely accepted software tool ProVerif [49–53] to perform security simulation and testing of the scheme, which can fully guarantee the characteristics of confidentiality and authenticity.

The complete scheme shown in Figure 4 is implemented and validated in ProVerif. During the simulation, we assumed the two channels shown in Figure 5(a). The ch is a common channel used for the transmission of messages between entities in the authentication phase. The sch is a secure channel for user and fog node registration. Variables and constants are also defined in Figure 5(a). ID_i and ID_j are the identities of users and fog nodes, respectively, SK_u , SK_f , and SK_c are the keys negotiated between the three entities, respectively.

User and fog node are described by starting and ending events, and scheme authenticity is achieved by exposing the respective relationships between the start and end intervals of related events initiated by a particular participant. If no end event is reached, it means the scheme failed to terminate and the scheme is incorrect. Figures 5(b)–5(d) represent the user, fog node, and CSP implementation simulation processes, respectively, which are described in detail in Section 3 and executed in parallel.

The necessary queries are defined in Figure 5(a) to verify the security and correctness of the scheme. The query attacker simulates an actual attack to obtain the session key and secret random numbers, while the other three query in-events

```

(*-----channel-----*)
free ch:channel.(*public channel*)
free sch:channel[private].(*secure channel,used for registering*)
(*-----shared key-----*)
free SKu:bitstring [private].
free SKf:bitstring [private].
free SKc:bitstring [private].
(*-----constants and variables-----*)
free s:bitstring [private].(*the CSP's secret kay*)
(*free SKu:bitstring [private].
free SKf:bitstring [private].
free SKc:bitstring [private].*)
free ri:bitstring [private].
free a:bitstring [private].
free b:bitstring [private].
free c:bitstring [private].
const IDi:bitstring.(*user'sID*)
const IDj:bitstring.(*fognode'sID*)
const Ri:bitstring.
const gj:bitstring.
const Ppub:bitstring.
const P:bitstring.
(*-----functions & reductions & equations-----*)
fun h(bitstring):bitstring.(*hashfunction*)
fun mult(bitstring, bitstring):bitstring.(*scalar multiplication operation*)
fun con(bitstring, bitstring):bitstring.(*concatention operation*)
reduc forall m:bitstring, n:bitstring:getmess(con(m, n))= m.
fun x or(bitstring, bitstring):bitstring.(*XOR operation*)
equation forall m:bitstring, n:bitstring:xor(xor(m, n), n)= m.
fun clcommit(bitstring, bitstring, bitstring):bitstring.(*pairing operation*)
(*-----queries-----*)
query attacker(SKu).
query attacker(SKf).
query attacker(SKc).
query attacker(ri).
query attacker(a).
query attacker(b).
query attacker(c).
query var:bitstring:inj-event(Userend(var))=> inj-event(UserStarted(var)).
query var:bitstring:inj-event(FogNodeend(var))=> inj-event(FogNodeStarted(var)).
(*query var inj-event(endCSP)==> inj-event(startCSP).*)
(*-----events-----*)
event UserStarted(bitstring).
event Userend(bitstring).
event FogNodeStarted(bitstring).
event FogNodeend (bitstring).

```

(a)

FIGURE 5: Continued.


```

(*-----user'sprocess-----*)
let ProcessUser=
new IDi:bitstring;
new PWi:bitstring;
new ri:bitstring;
let RIDi= xor(h(con(IDi, PWi)), ri)in
out(sch,(IDi,RIDi));(*userregistration:1*)
in(sch,(xRi:bitstring));
let Ri'=xor(xRi,ri)in(*userregistration:3*)
!
(
event UserStarted(IDi);
new a:bitstring;
let qi=xor(xor(Ri',ri),RIDi) in
let A=mult(mult(a,qi),P) in
let A'=mult(a,Ppub) in
let PIDi=xor(IDi,h(con(A',qi))) in
let Mi=xor(h(con(IDi,PWi)),Ri')in
new Tu:bitstring;
let Ni=h(con(con(con(con(A',Mi),A),IDi),IDj),Tu) in
let Msg1=(A,PIDi,Ni,Tu) in
out(ch,Msg1);(*authentication:1*)
in(ch,(xB:bitstring,xxC:bitstring,xxAuthi:bitstring,xxTc:bitstring));
let xxxAuthi'=h(con(con(con(con(con(A,xB),xxC),A'),IDi),xxTc)) in
if xxxAuthi=xxxAuthi' then
let Ku=clcommit(xB,xxC,mult(a,qi)) in
let SKu=h(con(con(con(Ku,A),xB),xxC)) in
event Userend(IDi);(*authentication:5*)
0
).

```

(b)

```

(*-----fognode'sprocess-----*)
let ProcessFogNode=
new IDj:bitstring;
out(sch,IDj);(*fognoderegistaring:1*)
in(sch,xgj:bitstring);(*fognoderegistaring:3*)
in(ch,(xA:bitstring,xPIDi:bitstring,xNi:bitstring,xTu:bitstring));
!
(
new b:bitstring;
event FogNodeStarted(IDj);
let B=mult(mult(b,gj),P) in
let B'=mult(b,Ppub) in
let PIDj=xor(h(con(B',gj)),IDj) in
new Tf:bitstring;
let Lj=h(con(con(con(con(con(B',gj),xA),PIDj),IDj),Tf)) in
let Msg2=(xA,B,xPIDi,PIDj,xNi,Lj,xTu,Tf) in
out(ch,Msg2);(*authentication:2*)
in(ch,(xC:bitstring,xAuthi:bitstring,xAuthj:bitstring,xTc:bitstring));
let xxAuthj'=h(con(con(con(con(xA,B),xC),B'),IDj),xTc) in
if xAuthj=xxAuthj' then
let Kf=clcommit(xA,xC,mult(b,gj)) in
let SKf=h(con(con(con(Kf,xA),B),xC)) in
let Msg4=(B,xC,xAuthi,xTc) in
out(ch,Msg4);
event tFogNodeend(IDj);(*authencationg:4*)
0
).

```

(c)

FIGURE 5: Continued.

```

(*-----CSP'sprocess-----*)
let UserReg=
in(sch,(rIDi:bitstring,rRiDi:bitstring));
new xi:bitstring;
new yj:bitstring;
let qi=h(con(con(rIDi,s),xi)) in
let Ri=xor(qi,rRiDi) in
out(sch,Ri)(*user registering:2*)
let FogNodeReg=z
in(sch,(rIDj:bitstring));
new xi:bitstring;
new yj:bitstring;
let gj=h(con(con(rIDj,s),yj)) in
out(sch,gj)(*fognode registering:2*)
let CSPAuth=
in(ch,(xA:bitstring,xB:bitstring,xxPIDi:bitstring,xPIDj:bitstring,xxNi:bitstring,
xLj:bitstring,xxTu:bitstring,xTf:bitstring));
new xi:bitstring;
new yj:bitstring;
let A''=mult(s,xxA) in
let B''=mult(s,xB) in
let IDi'=xor(xxPIDi,h(A'')) in
let IDj'=xor(xPIDj,h(B'')) in
let Mi'=h(con(con(IDi',s),xi)) in
let gj'=h(con(con(IDj',s),yj)) in
let xxxNi'=h(con(con(con(con(con(A'',Mi'),xxA),IDi'),IDj'),xxTu)) in
let xxLj'=h(con(con(con(con(con(B'',gj'),xxA),IDi'),IDj'),xTf)) in
if xxxNi=xxxNi' then
if xLj=xxLj' then
new c:bitstring;
let zc=h(con(con(xi,s),yj)) in
let C=mult(mult(c,zc),P) in
new Tc:bitstring;
let Authi=h(con(con(con(con(con(xxA,xB),C),A''),IDi'),Tc)) in
let Authj=h(con(con(con(con(con(xxA,xB),C),B''),IDj'),Tc)) in
let Kc=clcommit(xxA,xB,mult(c,zc)) in
let SKc=h(con(con(con(Kc,xxA),xB),C)) in
let Msg3=(C,Authi,Authj,Tc) in
out(ch,Msg3).
(*-----authentication:3-----*)
let ProcessCSP=UserReg|FogNodeReg|CSPAuth.
(*-----main-----*)
process
let Ppub=mult(s,P) in
(!ProcessUser|!ProcessFogNode|!ProcessCSP)

```

(d)

FIGURE 5: ProVerif simulation. (a) Declarations. (b) User's process. (c) Fog node's process. (d) CSP's process and main.

```

-- Query not attacker(SKu[])
nounif mess(sch[],rIDj_2969)/-5000
Completing...
Starting query not attacker(SKu[])
RESULT not attacker(SKu[]) is true.
-- Query not attacker(SKf[])
nounif mess(sch[],rIDj_7537)/-5000
Completing...
Starting query not attacker(SKf[])
RESULT not attacker(SKf[]) is true.
-- Query not attacker(SKc[])
nounif mess(sch[],rIDj_12015)/-5000
Completing...
Starting query not attacker(SKc[])
RESULT not attacker(SKc[]) is true.

```

(a)

```

-- Query not attacker(ri[])
nounif mess(sch[],rIDj_16493)/-5000
Completing...
Starting query not attacker(ri[])
RESULT not attacker(ri[]) is true.
--Query not attacker(a[])
nounif mess(sch[],rIDj_20971)/-5000
Completing...
Starting query not attacker(a[])
RESULT not attacker(a[]) is true.
--Query not attacker(b[])
nounif mess(sch[],rIDj_25449)/-5000
Completing...
Starting query not attacker(b[])
RESULT not attacker(b[]) is true.
--Query not attacker(c[])
nounif mess(sch[],rIDj_29927)/-5000
Completing...
Starting query not attacker(c[])
RESULT not attacker(c[]) is true.

```

(b)

```

-- Query inj-event(Userend(var))
==> inj-event(UserStarted(var))
nounif mess(sch[],rIDj_34441)/-5000
Completing...
Starting query inj-event(Userend(var))
==> inj-event(UserStarted(var))
RESULT inj-event(Userend(var))
==> inj-event(UserStarted(var)) is true.
-- Query inj-event(FogNodeend(var_52))
==> inj-event(FogNodeStarted(var_52))
nounif mess(sch[],rIDj_39848)/-5000
Completing...
Starting query inj-event(FogNodeend(var_52))
==> inj-event(FogNodeStarted(var_52))
RESULT inj-event(FogNodeend(var_52))
==> inj-event(FogNodeStarted(var_52)) is true.

```

(c)

FIGURE 6: Verification result. (a) Query results for SK. (b) Query results for secrecy. (c) Query results for events.

TABLE 2: Comparison of security.

Security properties	Ref. [36]	Ref. [46]	Ref. [41]	Our scheme
Known session-specific temporary information attack	Yes	No	Yes	Yes
User anonymity and untraceability	Yes	No	Yes	Yes
Mutual authentication	No [55]	—	Yes	Yes
Impersonation attacks	No [55]	—	No [56]	Yes
Man-in-the-middle attacks	Yes	Yes	Yes	Yes
Known session key attacks	Yes	Yes	Yes	Yes
Compromise impersonation attacks	—	—	Yes	Yes
Parallel session attacks	—	—	—	Yes
Stolen smart card attacks	Yes	Yes	Yes	Yes
Password-guessing attacks	Yes	Yes	Yes	Yes
Privileged-insider attacks	Yes	—	—	Yes
Replay attacks	No [55]	Yes	Yes	Yes
Perfect forward privacy	Yes	Yes	Yes	Yes
No key control	—	Yes	—	Yes
Unknown key-share	—	—	—	Yes

TABLE 3: Computation time of basic operations.

Operation	Description	Times (ms)
TG_e	Bilinear pairing	17.4
TG_m	Scalar multiplication	13.5
TG_a	Point addition	0.48
T_h	Hash function	0.42
T_{fe}	Fuzzy extractor function [36]	17.1

TABLE 4: Performance comparisons (computation costs).

	Ref. [36]	Ref. [46]	Ref. [41]	Our scheme
Authentication and key agreement	$3TG_m + 19T_h + 1T_{fe}$	$3TG_e + 7TG_m + 18T_h$	$4TG_e + 10TG_m + 25T_h$	$3TG_e + 10TG_m + 21T_h$
Total	65.58 ms	154.26 ms	215.1 ms	196.02 ms

correspond to the start and end events of the three processes. If any of these queries result in false, it means that the scheme is incorrect. The results of the discussion query are shown in Figure 6.

It can be seen from the results in Figures 6(a) and 6(b) that the session key negotiated between entities and the secret random number selected by each entity are secure when dealing with security threats, which proves that the authenticity and confidentiality of our scheme are guaranteed during the execution process. The results in Figure 6(c) show that each process started and ended successfully, which proves the correctness of our scheme.

5. Performance Evaluation

In this section, the security features and defense against various attacks are compared between our scheme and the previous schemes [36, 41, 46] in Table 2. We can conclude that our scheme is more secure than the compared schemes. Note that “Yes” represents that the scheme can resist the indicated attack, whereas “No” represents that the scheme cannot, and “—” represents that the attack method indicated is not in the scope of the scheme.

Subsequently, we evaluate the performance of the proposed scheme from the perspective of computational and

communication costs. The improved scheme was implemented in JAVA with JDK version 1.3, and the simulation of the scheme was based on the JAVA paired cryptography library (JPBC) [54], version JPBC-2.0.0. A Windows 10 computer system was used as the experimental platform, which was configured with a quad-core 2.3 GHz Intel(R) Core i5-8300H processor and 16 GB memory. The software developed is the community version of IntelliJ IDEA 2020.2.1 and uses the widely accepted type A pairing, which is based on the curve $y^2 = x^3 + x$ structure in the field F_q of a specific $q = 3 \pmod{4}$. We have listed the symbols (TG_e , TG_m , T_h , TG_a) and time used in the performance comparison in Table 3. Table 4 presents the calculation costs for the different phases of the scheme.

As shown by the analysis in Table 4, the computing cost for our scheme is slightly higher than that of schemes [36, 46]; however, our scheme provides auxiliary security features, and the mandatory security objectives achieved by this scheme are greater than those achieved by other schemes [36, 41, 46]. Our solution provides security features that other solutions do not have, such as being able to resist replay attacks and impersonation attacks and providing user anonymity, mutual authentication, etc.

To calculate the communication and storage costs, we present that the length of the random nonce, password, and

TABLE 5: Performance comparisons (communication costs).

	Ref. [36]	Ref. [46]	Ref. [41]	Our scheme
Authentication and key agreement	$5 G_1 + 4 q $	$6 G_1 + 9 q + 5 T $	$3 G_1 + 4 q + 4 T $	$6 G_1 + 9 q + 5 T $
Total	5760 bit	7744 bit	3840 bit	7744 bit

TABLE 6: Performance comparisons (storage cost).

Scheme	Storage cost (bits)
Ref. [36]	320
Ref. [46]	640
Ref. [41]	736
Ours	640

identity is 160 bits, and the length of a point in the G_1 group is 1024 bits, denoted as $|G_1|$. The output length of the hash functions h_0, h_1, h_2, h_3 , and h_4 in Z_p^* is 160 bits, denoted as $|q|$. The output length of h_5 and the key length are both 256 bits. The length of the timestamp is 32 bits, denoted as $|T|$. The communication and storage costs of our scheme and related schemes are listed in Tables 5 and 6.

As shown in Tables 5 and 6, the communication and the storage overhead of our scheme are slightly higher. The slightly higher cost of our scheme is mainly due to the increase in computing overhead while providing stronger security. However, because the primary purpose of a scheme is to ensure the security and privacy of data, it is acceptable to have a slightly higher communication cost but stronger security. After analyzing Tables 4 and 5, our scheme is concluded to be better than the other schemes [36, 41, 46], which can provide stronger security and withstand various known attacks.

6. Conclusion

The usage of fog-driven IoT healthcare systems has brought significant convenience to people. The authentication of the healthcare system is also the most important. Recently, a growing number of scholars have taken a closer look at healthcare systems and developed stronger authentication protocols for their certification environments. In this study, we proposed a secure authenticated and key agreement scheme in fog-driven IoT healthcare systems; the defects of the original scheme were analyzed and security improvements were proposed. An analysis of the performance evaluation and informal security in comparison to other related schemes is also presented in this study, which indicates that our scheme provides more security features. Our solution uses pairing technology, and the time cost is slightly higher than other solutions. Future studies can improve on this limitation, but our solution provides security features that other solutions do not have, which is more suitable for the practical application of medical system based on the IoT.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The work was supported in part by National Key Research and Development Project, China, under Grant no. 2018YFC1201102 and the Natural Science Foundation of Fujian Province, China, under Grant nos. 2018J01636 and 2018J01638.

References

- [1] F. C. Chang and H. C. Huang, "A survey on intelligent sensor network and its applications," *Journal of Network Intelligence*, vol. 1, no. 1, pp. 1–15, 2016.
- [2] J. S. Pan, L. Kong, T. W. Sung, P. W. Tsai, and V. Snasel, "Alpha-fraction first strategy for hierarchical wireless sensor networks," *Journal of Internet Technology*, vol. 19, no. 6, pp. 1717–1726, 2018.
- [3] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, 2019.
- [4] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.
- [5] J. Wang, Y. Gao, C. Zhou, R. Simon Sherratt, and L. Wang, "Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs," *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.
- [6] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.
- [7] T. N. Gia, M. Jiang, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare internet of things: a case study on ECG feature extraction," in *Proceedings of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pp. 356–363, IEEE, Liverpool, UK, October 2015.
- [8] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT ehealth: promises and challenges of IoT in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.
- [9] A. M. Rahmani, T. N. Gia, B. Negash et al., "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.
- [10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of*

- the First Edition of the MCC Workshop on Mobile Cloud Computing ACM*, pp. 13–16, Helsinki, Finland, August 2012.
- [11] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: a survey," in *Wireless Algorithms, Systems, and Applications. WASA 2015. Lecture Notes in Computer Science*, K. Xu and H. Zhu, Eds., vol. 9204, pp. 685–695, Springer, Cham, Switzerland, 2015.
 - [12] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: architecture, use case, and security and forensic challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, 2017.
 - [13] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From cloud to fog computing: a review and a conceptual live vm migration framework," *IEEE Access*, vol. 5, pp. 8284–8300, 2017.
 - [14] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.
 - [15] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT," *IEEE Internet of Things Journal*, .
 - [16] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, .
 - [17] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.
 - [18] A. Q. Tian, S. C. Chu, J. S. Pan, H. Cui, and W. M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.
 - [19] S. C. Chu, X. Xue, J. S. Pan, and X. Wu, "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.
 - [20] Y. Huang, M. Hsieh, H. Chao, S. Hung, and J. Park, "Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 400–411, 2009.
 - [21] J. S. Pan, X. X. Sun, S. C. Chu, A. Abraham, and B. Yan, "Digital watermarking with improved SMS applied for QR code," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104049, 2021.
 - [22] R. Tso, "Two-in-one oblivious signatures," *Future Generation Computer Systems*, vol. 101, pp. 467–475, 2019.
 - [23] T.-Y. Wu, C.-M. Chen, K.-H. Wang, C. Meng, and E. K. Wang, "A provably secure certificateless public key encryption with keyword search," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 20–28, 2019.
 - [24] J. Zhang, H. Liu, and L. Ni, "A secure energy-saving communication and encrypted storage model based on RC4 for EHR," *IEEE Access*, vol. 8, pp. 38995–39012, 2020.
 - [25] J. M.-T. Wu, G. Srivastava, A. Jolfaei, P. Fournier-Viger, and J. C.-W. Lin, "Hiding sensitive information in eHealth datasets," *Future Generation Computer Systems*, vol. 117, pp. 169–180, 2021.
 - [26] C. M. Chen, L. Xu, T. Y. Wu, and C. R. Li, "On the security of a chaotic maps-based three-party authenticated key agreement protocol," *Journal of Network Intelligence*, vol. 1, no. 2, pp. 61–66, 2016.
 - [27] C. M. Chen, Y. Huang, E. K. Wang, and T. Y. Wu, "Improvement of a mutual authentication protocol with anonymity for roaming service in wireless communications," *Data Science and Pattern Recognition*, vol. 2, no. 1, pp. 15–24, 2018.
 - [28] S. Kumari, P. Chaudhary, C.-M. Chen, and M. K. Khan, "Questioning key compromise attack on Ostad-Sharif et al.'s authentication and session key generation scheme for healthcare applications," *IEEE Access*, vol. 7, pp. 39717–39720, 2019.
 - [29] P. Wang, C.-M. Chen, S. Kumari et al., "HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, p. 1, 2020.
 - [30] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.
 - [31] M. Turkanovic, B. Brumen, and M. Holbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.
 - [32] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.
 - [33] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, 2016.
 - [34] T. Hayajneh, B. J. Mohd, M. Imran, G. Almashaqbeh, and A. V. Vasilakos, "Secure authentication for remote patient monitoring with wireless medical sensor network," *Sensors*, vol. 16, no. 4, p. 424, 2016.
 - [35] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.
 - [36] S. Challa, A. K. Das, V. Odelu et al., "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Computers & Electrical Engineering*, vol. 69, pp. 534–554, 2018.
 - [37] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3133–3142, 2019.
 - [38] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.
 - [39] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, "An authenticated key exchange protocol for multi-server architecture in 5G networks," *IEEE Access*, vol. 8, pp. 28096–28108, 2020.
 - [40] C.-M. Chen, Y. Huang, K.-H. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, p. 1, 2020.
 - [41] M. Nikravan and A. Reza, "A multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things," *Wireless Personal Communications*, vol. 111, no. 1, pp. 463–494, 2020.
 - [42] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Advances in Cryptology—EUROCRYPT 2000. EUROCRYPT 2000*.

- Lecture Notes in Computer Science*, B. Preneel, Ed., Vol. 1807, Springer, Berlin, Germany, 2000.
- [43] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *Journal of Cryptology*, vol. 17, no. 4, pp. 263–276, 2004.
 - [44] C. T. Li, T. Y. Wu, C. L. Chen, C. C. Lee, and C. M. Chen, "An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system," *Sensors*, vol. 17, no. 7, p. 1482, 2017.
 - [45] H. A. Hamid, S. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22313–22328, 2017.
 - [46] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.
 - [47] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques—Advances in Cryptology (EUROCRYPT'02)*, pp. 337–351, Amsterdam, Netherlands, April 2002.
 - [48] M. Burrows, R. A. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 24, no. 20, pp. 18–36, 1975.
 - [49] B. Blanchet, M. Sylvestre, M. X. Allamigeon, V. Cheval, B. Smyth, and C. Stentzel, "ProVerif: cryptographic protocol verifier in the formal model," 2019, <http://prosecco.gforge.inria.fr/personal/bblanche/proverif/>.
 - [50] K. Mansoor, A. Ghani, S. Chaudhry, S. Shamshirband, S. Ghayyur, and A. Mosavi, "Securing IoT-based RFID systems: a robust authentication protocol using symmetric cryptography," *Sensors*, vol. 19, no. 21, p. 4752, 2019.
 - [51] B. A. Alzahrani, S. A. Chaudhry, A. Barnawi, A. Al-Barakati, and M. H. Alsharif, "A privacy preserving authentication scheme for roaming in IoT-based wireless mobile networks," *Symmetry*, vol. 12, no. 2, p. 287, 2020.
 - [52] S. A. Chaudhry, "Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106529, 2021.
 - [53] T. Y. Wu, Y. Q. Lee, C. M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, 2021.
 - [54] A. D. Caro and V. Iovino, "JPBC: java pairing based cryptography," in *Proceedings of the 2011 IEEE Symposium on Computers and Communications (ISCC)*, pp. 850–855, IEEE, Corfu, Greece, June 2011.
 - [55] Z. Ali, A. Ghani, I. Khan, S. A. Chaudhry, S. H. Islam, and D. Giri, "A robust authentication and access control protocol for securing wireless healthcare sensor networks," *Journal of Information Security and Applications*, vol. 52, Article ID 102502, 2020.
 - [56] S. Shamshad, K. Mahmood, and S. Kumari, "Comments on "a multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things," *Wireless Personal Communications*, vol. 112, no. 1, pp. 463–466, 2020.