EFFICIENT PRIVACY-PRESERVING DATA AGGREGATION AND REPLICATION FOR FOG-ENABLED IOT

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Supervisors

Dr. Sira Yongchareon

Dr. Jian Yu

Dr. Saeed ur Rehman

June 2021

By

Kinza Sarwar

School of Engineering, Computer, and Mathematical Sciences

Abstract

With the increasing popularity of Fog computing to provide computation, analysis and storage of data at the edge of IoT networks, the fulfilment of data privacy requirements over fog networks can be seen as one of the biggest security challenges. Data aggregation is considered an essential privacy requirement as it combines data from different IoT devices to protect the data leakage of an individual IoT device. It also reduces data redundancy while improving data analysis speed in Fog-enabled IoT networks. For preserving the privacy of data aggregation, the heavyweight cryptosystems are considered, which faces issues related to performance overhead and single point of failure risks due to data aggregation at a single fog node. In addition, no secure data replicas exist for data recovery and reliability in case of a data breach in Fog-enabled IoT applications. This thesis proposes an efficient privacy-preserving scheme for data aggregation to overcome the limitations of Fog-enabled IoT applications. This thesis also proposes an efficient privacy-preserving data replication scheme for data reliability and recovery.

The proposed data aggregation scheme is based on lightweight data encryption and data division method. This method effectively divides data according to Level of Privacy (LoP) and distributes the data among participating fog nodes for aggregation and storage processing, and reduces computational and memory overhead in the processing simultaneously. The proposed data aggregation scheme is further extended to optimize the time and energy consumption of the data division method. The multi-objective optimization method is defined for the data aggregation scheme based on the NSGA-III (non-dominated sorting genetic Algorithm III) to find optimal solutions concerning time consumption and energy consumption.

A data replica creation scheme and a data replica placement scheme are proposed to preserve the privacy of data replicas. The data replica creation scheme is based on a Level of Privacy (LoP) defined by data-owners and the service capacity of fog nodes. The proposed data replica placement scheme is based on the priority level of fog nodes.

Moreover, comprehensive simulations and systematic experiments are conducted to demonstrate and evaluate the effectiveness and efficiency of the proposed schemes compared with the state-of-the-art schemes. The results demonstrate that the proposed schemes can efficiently achieve data privacy in the fog computing paradigm and outperform other schemes in terms of performance efficiency.

Table of Contents

Abstract ii
List of Figures vi
List of Tables vii
Attestation of Authorship viii
Co-authored Work ix
Acknowledgements x
Dedication xi
Chapter 1 1
Introduction 2
1. Background 2
1.1. Fog Computing 3
1.2. Data Privacy 4
1.3. Data Aggregation 5
1.4. Data Replication 6
2. Research Problems 7
3. Research Questions 8
4. Research Contributions 9
5. Publications 11
6. Research Methodology 11
7. Thesis Structure 13
Chapter 2: A Survey on Privacy Preservation for Fog-Enabled IoT 15
Abstract 15
1. Introduction 16
2. Privacy requirement In Fog-enabled IoT 19
2.1. Evolution of Privacy in IoT and Fog-enabled IoT 19
2.2. Classification of Privacy Requirements in Fog-enabled IoT 21
2.2.1. Content Privacy 21
2.2.2. Context Privacy 23
3. Review of Privacy Preservation in IoT and Fog-enabled IoT 26
3.1. Review of Privacy Preservation in IoT 26
3.1.1. Content Privacy 27
3.1.2. Context Privacy 34
3.2. Review of Privacy Preservation in Fog-enabled IoT 39
3.2.1. Content Privacy 39
3.2.2. Context Privacy 45
3.3. Privacy Preservation of IoT Application in Fog Computing 48
3.3.1. Content Privacy 49
3.3.2. Context Privacy 51
4. Open Issues and Challenges 53
4.1. Privacy Privilege Escalation 53
4.2. Tracking Data Accuracy 55
4.3. Data-owner's usage pattern privacy 55
4.4. Rogue Fog node 56
4.5. Certificate Management 57
4.6. Communication Overhead 58
5. Conclusion 59
Chapter 3: Lightweight, Divide-and-Conquer Privacy-Preserving Data Aggregation in Fog Computing 60

Abstract 60
1. Introduction 61
2. Related Work 63
3. The Divide-and-Conquer Scheme 65
3.1. System and Adversary Model 65
3.2. Network Design and Setup 67
3.3. Level of Privacy (LoP) and Distribution Setup 68
3.4. Nodes Authentication 70
3.5. Data Encryption 70
3.6. Divide-and-Conquer Scheme 72
3.7. Data Aggregation and Decryption 73
4. Privacy and Performance Analysis 75
4.1. Experiment Setup 75
4.2. Formal Security Analysis 75
4.3. Performance Analysis 79
4.3.1. Computational Cost 80
4.3.2. Memory Cost 82
4.3.3. Communication Cost 84
5. Conclusion 85
Chapter 4: Joint Optimization of Time and Energy Consumption for Data Aggregation in Fog-enabled IoT Networks 87
Abstract 87
1. Introduction 88
2. Related Work 89
3. System Model and Problem Formulation 91
3.1. System Model 91
3.2. Time Consumption Model 92
3.3. Energy Consumption Model 96
3.4. Problem formulation and Constraints 97
4. Multi-objective optimization method (MUOM) in fog computing 97
4.1. Encoding 97
4.2. Fitness functions and Constraint 98
4.3. Initialization 99
4.4. Crossover and mutation 99
4.5. Selection for the next generation 100
4.6. Optimal selection using SAW and MCDM 101
4.7. Proposed MUOM overview 103
5. Experimental Evaluation 104
5.1. Fog computing Test-case Architecture 104
5.2. Simulation Setup 104
5.3. Evaluation Criteria 105
5.4. Performance Evaluation of Proposed MUOM 106
5.4.1. Impact of the Number of Fog nodes 106
5.4.2. Impact of the Execution and Transmission Power 107
5.4.3. Impact of Computing Capacity of Fog nodes 109
5.5. Comparison Analysis 109
5.5.1. Comparison of data sizes for time consumption and energy consumption 112
5.5.2. Comparison of power consumption 113
5.5.3. Comparison of workload imbalance 114

6. Conclusion 115
Chapter 5: Efficient Privacy-Preserving Data Replication for Fog-enabled IoT 117
Abstract 117
1. Introduction 118
2. Related Work 119
2.1. Data Replica Creation and Placement 119
2.2. Data Replica Creation and Placement in Fog-enabled IoT 122
2.3. Privacy in Data Replication 122
3. Data Replica Creation and Placement Schemes: Model and Solution 123
3.1. System Model 123
3.2. Adversary Model 124
3.3. Data Replica Creation Scheme based on Level of Privacy and Service Capacity 125
3.3.1. Miner Nodes Selection for Data Replica Creation 127
3.3.2. Data Replica Creation at Target Miner Node 128
3.4. Data Replica Placement Scheme Based on Level of Priority 129
3.4.1. Data Replica Placement at Fog nodes 131
3.5. Time Complexity Analysis of the Proposed Schemes 133
3.5.1. Time complexity Analysis of Data Replica Creation Scheme 134
3.5.2. Time complexity Analysis of Data Replica Placement Scheme 134
4. Simulation and Evaluation 135
4.1. Simulation Setup 135
4.2. Experimental Results and Analysis 135
4.2.1. Privacy Analysis of the Proposed Schemes 136
4.2.2. Performance Analysis for Data Replica Creation 137
4.2.3. Performance Analysis for Data Replica Placement 139
4.2.4. Time Series Analysis of the Proposed Replica Creation and Placement Schemes with Autoregressive Integrated Moving Average (ARIMA) 140
4.3. Performance Analysis of the Proposed Replica Creation Scheme vs DRC-AH and DRCA schemes 142
5. Conclusion and Future Work 145
Chapter 6 Conclusion and Future Work 146
6.1. Summary 147
6.2. Thesis Limitations and Recommendations for Future Research 150
References 152

List of Figures

Figure 1 Fog Computing Architecture 3
Figure 2 Research Methodology 12
Figure 3 Thesis Structure 14
Figure 4. Fog Computing Environment 17
Figure 5 Publications on Privacy Preservation in IoT and Fog-enabled IoT in the Literature from 2010-2020 19
Figure 6 Privacy Requirements in Fog-enabled IoT 21
Figure 7 Example of Data-aggregation 23
Figure 8 Fog-enabled IoT Applications 39
Figure 9 Content Privacy Models in Fog-enabled IoT 40
Figure 10 Context Privacy Models in Fog-enabled IoT 46
Figure 11 An overview system model of the proposed scheme 65
Figure 12 Sequence Diagram (SSD) for Divide-and-Conquer privacy-preserving data aggregation scheme 66
Figure 13 Computational overhead of scheme vs. Masker, ECBDA, APPA & LVPDA schemes 81
Figure 14 Memory size of scheme vs the ECBDA, Masker, APPA & LVPDA Schemes 83
Figure 15 Communication overhead of our scheme vs. the ECBDA, Masker, APPA, LVPDA schemes 85
Figure 16 System Model 92
Figure 17 Example of Encoding Chromosomes 98
Figure 18 Example of crossover operation 100
Figure 19 Example of mutation operation 100
Figure 20 Impact of the number of fog nodes. (a) Pareto front for optimal solutions. (b) Box plots of the time consumption in a varying number of fog nodes. (c) Box plots of the energy consumption in a varying number of fog nodes. 108
Figure 21 Impact of execution and transmission power. (a) Pareto front for optimal solutions. (b) Box plots of the time consumption in varying power values. (c) Box plots of the energy consumption in varying power values. 110
Figure 22 Impact of the computing capacity of fog nodes. (a) Pareto front for optimal solutions. (b) Box plots of the time consumption in varying computing capacity values. (c) Box plots of the energy consumption in varying computing capacity values. 111
Figure 23 Data size comparison for time consumption 113
Figure 24 Data size comparison for energy consumption 113
Figure 25 Power comparison for Time Consumption 114
Figure 26 Power comparison for Energy Consumption 114
Figure 27 The degree of workload imbalance. 115
Figure 28 The standard deviation of workload distribution 115
Figure 29 Fog-enabled IoT Network 124
Figure 30 The Architecture of Replica Creation Scheme 125
Figure 31 Architecture of Replica Placement 131
Figure 32 Flowchart for Data Placement Processes at Miner head 132
Figure 33 Number of Replica Vs Service Capacity 137
Figure 34 On Average Service Capacity for Different Priority level 139
Figure 35 Diagnostic Result of the Replica creation model 140
Figure 36 On Average Actual Vs Predicted Forecast at Fog Layer 1 141
Figure 37 On Average Actual Vs Predicted Forecast at Fog Layer 2 142
Figure 38 Computational Cost Comparison 143
Figure 39 Memory Cost Comparison 144
Figure 40 Communication Cost Comparison 145

List of Tables

Table 1 Privacy-based IoT Application Domains 27
Table 2 Techniques for Content Privacy Preservation in IoT 28
Table 3 Techniques for Context Privacy Preservation in IoT 35
Table 4 Mapping of Content Privacy models between IoT and Fog-enabled IoT 50
Table 5 Mapping of Context Privacy Models between IoT and Fog-enhanced IoT 52
Table 6 Issues/Challenges of Privacy solutions in Fog-enabled IoT 54
Table 7 Distribution setup 69
Table 8 Symbols used in Algorithms 70
Table 9 Average Computational overhead (in milliseconds) based on security parameters 80
Table 10 Data size variation results from Figure. 14 83
Table 11 Key notations and description 91
Table 12 Parameters Settings 104
Table 13 Summary of Symbols and Abbreviations 123
Table 14 Sensitive Parameters 135
Table 15 Level of Privacy Vs Total Number of Replicas 138
Table 16 Summary Statistics 142

Attestation of Authorship

I, Kinza Sarwar, hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signed: _____

Dated: 15/06/2021

Co-authored Work

All co-authors in the following table have approved these chapters for inclusion in Kinza Sarwar's doctoral thesis.

Chapter	Author %
Chapter 2: Kinza Sarwar, Sira Yongchareon, Jian Yu, Saeed ur Rehman. A Survey on Privacy Preservation in Fog-Enabled Internet of Things Manuscript accepted in ACM Computing Survey Journal	KS = 80 SY = 10 JY = 5 SR = 5
Chapter 3: Kinza Sarwar, Sira Yongchareon, Jian Yu, Saeed ur Rehman. Lightweight, Divide-and-Conquer Privacy-Preserving Data Aggregation in Fog Computing Manuscript published in Future Generation Computer System Journal (Elsevier)	KS = 80 SY = 10 JY = 5 SR = 5
Chapter 4: Kinza Sarwar, Sira Yongchareon, Jian Yu, Saeed ur Rehman. Joint Optimization of Time Consumption and Energy Consumption for Data Aggregation in Fog-enabled IoT Networks. Manuscript submitted to IEEE Internet of Things Journal	KS = 82 SY = 10 JY = 4 SR = 4
Chapter 5: Kinza Sarwar, Sira Yongchareon, Jian Yu, Saeed ur Rehman. Efficient Privacy-Preserving Data Replication in Fog-enabled IoT. Manuscript submitted to Future Generation Computer Systems Journal (Elsevier)	KS = 82 SY = 10 JY = 4 SR = 4
Kinza Sarwar (KS), Sira Yongchareon (SY), Jian Yu (JY), Saeed ur Rehman (SR)	

We, the undersigned, hereby agree to the percentages of participation to the chapters identified above:

Sira Yongchareon **Jian Yu** **Saeed ur Rehman** **Kinza Sarwar**

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Sira Yongchareon, for providing invaluable guidance and unflinching support throughout this research. He has taught me the methods and strategies to carry out and present the research work as clear as possible. I could not have imagined having a better mentor for my PhD study.

My special thanks to my second supervisor, Dr. Jian Yu, for his insightful comments and feedback on research work, which has always been valuable to research outcomes. I am also grateful to my third supervisor, Dr. Saeed ur Rehman, for his reviews and thoughtful suggestions throughout the research.

Last but not least, I would like to thank my family for their love, patience and support throughout this journey.

Dedication

To my husband Umar and son Wajdaan

whose support, tolerance and enthusiasm has enabled me to complete this research work

To my parents Muhammad Sarwar Afandi and Imtiaz Sarwar, and parents-in-law Abdul Ghafoor and Umi Salma Kanwal

whose ceaseless prayers and encouragement made the success of this work a reality

And my brothers Naveed Afandi and Shahzad Afandi

whose word of advice always inspired me

'Solve the problem or leave the problem. Do not live with the problem.'

---A Motivator

Chapter 1

The area of research and factors that motivated to conduct the proposed research are introduced in this Chapter. Also, the Chapter identifies the gaps in the existing area of research and then present contributions to bridge those gaps. Finally, the research methodology and structure of the thesis is provided at the end of this Chapter.

Introduction

Fog computing is becoming popular as it provides computing, security, networking, and storage capabilities to the Internet of Things (IoT) applications at the edge of the IoT network. The focus of this thesis is on data privacy in fog-enabled IoT networks. The key idea is to develop a framework for lightweight privacy-preserving data aggregation and replication with high-performance efficiency. At present, data aggregation and replication considering privacy concerns over fog networks have not been explored to establish an efficient and lightweight privacy-preserving framework. This thesis uses distributive computation and storage in the fog computing paradigm to optimize performance efficiency. Also, a data-owner-defined level of privacy is considered to strengthen the data privacy for data aggregation and replication of fog-enabled IoT networks.

Section 1 in this Chapter begins with the background of the factors that led to this research work. Section 2 addresses the research problems and introduces research questions in Section 3. The research contributions are discussed in Section 4, followed by research publication and research methodology in Section 5 and Section 6, respectively. Finally, the structure of the thesis is presented in Section 7.

1. Background

With the advancement in wireless communications and ubiquitous computing, a paradigm known as the Internet of Things (IoT) is gaining mainstream acceptance and research interest. IoT is a collection of physical devices, aka 'things' embedded with actuators, sensors, software, and electronics, to collect and exchange data with other devices using an internet connection [1]. The things can be personal, industry, and enterprise objects such as smartphones, smart appliances, wearable, digital cameras, tablets, vehicles, smart security, and smart lighting. These things are connected to the internet for data transmission, processing, and analysis, and hence things could be managed and controlled remotely [2].

The interconnectivity of smart things has resulted in the wide deployment of IoT networks worldwide. The IoT connects smart things for the interaction of human to machine (H2M), human to human (H2H), machine to machine (M2M) while providing ease of control, management, analysis, communication, and identification among the IoT devices [3]. The interconnectivity has significantly improved everyday life [4], such as home security, household activities, smart supply chain, infrastructure support, pervasive health care (smart hospitals), assisted living, smart meters for balancing bills, air quality management [5]. It is estimated that IoT devices connected to the Internet will reach 41.6 billion by the year 2025 [6]. Also, the amount of data generated by IoT devices is increasing in size [4]. This rapid growth in IoT devices

connectivity and data size has substantially increased the performance overhead of IoT devices to process, analyze and transmit data [7].

The computation power of IoT devices to process and analyze data is limited as a key purpose of IoT devices is to supply data about things while remaining autonomous [8]. IoT devices have sufficient processing power to supply data via the internet to a server. This processing power is, however, insufficient to fulfil requirements for heavy data processing and analyzing. Due to the IoT devices' limited battery and processing resources, the generated data is offloaded to cloud computing for computation, analysis, and long-term storage. Cloud computing allows data scalability to be scaled vertically and horizontally to meet heavy data processing and analysis requirements.

The Internet is neither sufficiently scalable nor efficient to deal with IoT data offloading. The IoT network requires an enormous amount of energy, time, and bandwidth for data offloading to a cloud server that is located remotely [9]. In addition, for heavy processing and analysis, IoT networks' overall energy and bandwidth overhead increase to offload data to cloud computing. CISCO researchers Bonomi *et al.* [10] proposed the fog computing concept in 2012 as an alternative paradigm to mitigate the limitations mentioned above for IoT data offloading to the cloud paradigm.

1.1. Fog Computing

Fog computing is an architecture that uses edge devices of a network to perform data communication, computation, and storage locally that are routed over the internet [11]. The

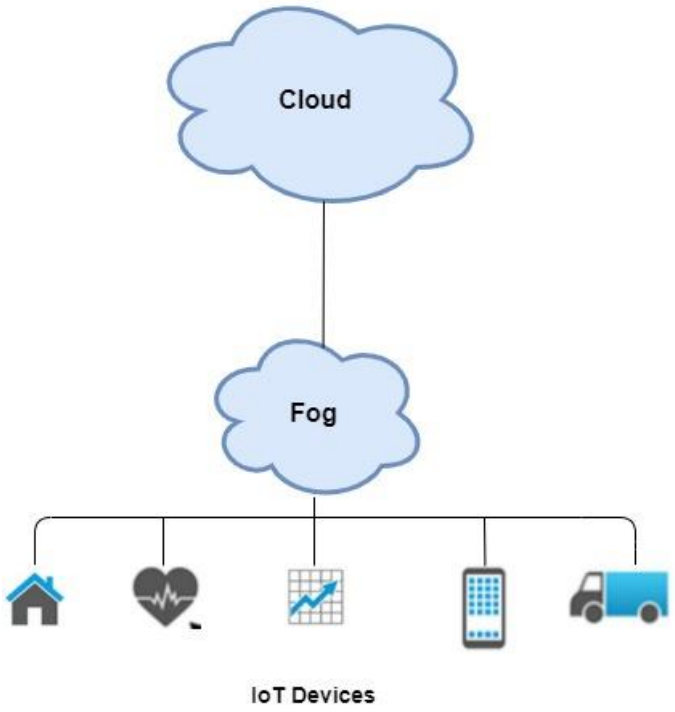


Figure 1 Fog Computing Architecture

word 'fog' is used for cloud periphery as it is distributed computing for peripheral devices connecting to a cloud. The fog computing paradigm partly shifts the cloud's processing and storage tasks to the edge of an IoT network. It can provide computing, networking, and storage capabilities to IoT devices with each fog node located near the IoT device [12]. Fog computing can be viewed as an extension of cloud computing, as shown in Figure 1. It reduces the amount of data transfer and processing to the cloud paradigm thus, alleviating much of the burden to fog servers themselves and improving the performance efficiency of IoT networks [13].

In cloud computing, data servers/centers are the main centralized components to process and store data. Due to this, cloud computing has high energy consumption, processing, and operational cost [14]. On the other hand, fog computing architecture consists of fog clusters in which fog devices cooperate to compute, communicate, network, and store tasks that consume low energy, processing, and operational cost. Also, the distance between cloud and IoT devices is the multi-hop distance [15]. In contrast, the distance between fog nodes and IoT devices is one or a few hops as fog nodes are located near the edge of the IoT network. Because of this distance, the fog paradigm's network and communication latency are always low compared to the cloud paradigm. A real-time application handling in the fog paradigm is achievable due to its low network and communication latency, whereas real-time interaction in cloud paradigm is difficult due to high latency.

As the concept of fog computing is introduced for IoT applications, the terminology fog-enabled IoT is used interchangeably [16]. The fog-enabled IoT paradigm allocates the processing of IoT data in a distributive manner to optimize the performance efficiency and bandwidth bottleneck of the IoT network.

1.2. Data Privacy

Data privacy is a branch of data security that governs how data is gathered, shared, and used in compliance with regulatory obligations and data consents. It is the right of a data-owner to have awareness and control of their personal data use [17]. However, data privacy is not limited to control and awareness. It also considers procedures compliance with data protection laws to collect, process, and share data securely [18]. When data that should be kept private gets exposed or breached by an adversary, damaging consequences may occur, including operational downtime, loss of sensitive data, financial loss, legal action, and reputational damage [19]. For example, a data breach of top-secret information at a government agency or proprietary data at a corporation put in the hands of a competitor can cause tremendous reputational damage and financial loss.

In IoT applications, data privacy is not negligible as the data-owners surrender their privacy in the form of data [20], bit by bit, without realization/awareness of what data content is being collected, exposed, and used. Data content not only contains general data fields of a data-owner, for example (name, telephone number, or address) but may also contain very sensitive information, including readings of habitual behavior and medical health reports [21]. For example, collecting data from hundreds of smart meters in a smart grid system can raise electricity consumers' privacy issues such as exposure to daily activity patterns and location tracking [22].

It is essential to preserve the data privacy of IoT devices before transmission to data receivers so that the data receiver cannot trace back to the source of data generation and misuse it. Strong privacy algorithms (e.g., public and private cryptography, privacy compliances) are required to preserve data privacy. Many kinds of IoT devices have limited memory storage, energy budget for power batteries, and constrained processing resources [23, 24]. Therefore, implementing efficient privacy algorithms on those IoT devices is critical. The cloud computing paradigm [25-31] has been considered for processing privacy algorithms to preserve the data privacy of IoT devices. However, transmitting data to the cloud paradigm increases the potential risks of data leakage. Also, the offloading of privacy tasks to the cloud increases the IoT network bottleneck.

In contrast, fog computing provides a promising medium for preserving the data privacy of IoT devices. The processing of privacy algorithms can be transmitted to fog nodes instead of remote cloud to reduce the network bottleneck. While fog nodes reside near IoT devices, a fog computing paradigm mitigates IoT devices' risk of data privacy leakage [13].

1.3. Data Aggregation

Data is constantly expanding and evolving with technology advancements, and to better understand data, the extraction and organization of key data trends are important. The data aggregation process helps in organizing, summarizing, and analyzing trends of data. It is a process of gathering data from multiple data sources and combining data sources into a comprehensive and consumable data set for further use. For example, an organization often collects and aggregates its online customer data to market the product [32]. The aggregated data includes the statistical analysis of online product purchases and demographics such as the number of purchases and average customer age. These statistics assist the marketing team to learn about the customer's digital experience and product purchase success. Different mathematical operations such as maximum, minimum, sum, percentile, ratio, and average [33] are performed for data aggregation. Maximum, minimum, and sum operations are categorized

as additive aggregation operations, and ratio, average, and percentile are non-additive operations.

Data aggregation is one of the requirements for data privacy in IoT applications. It combines data from different IoT sources, such as IoT devices, using aggregation operations to protect the data leakage of individual IoT devices [33]. It also significantly reduces the energy, computational and communication overhead to process data of IoT devices. Furthermore, it eliminates data redundancy and improves data analysis speed and efficiency [13]. For example, energy companies collect power consumption data from smart meters installed at customer sites and aggregate data to improve the overall efficiency and reliability of smart grid infrastructure [34]. Similarly, health data is aggregated for medical research in healthcare [35]. In the transport management system, aggregated traffic data is used to analyze the route network to improve transportation services [36].

Different public and private cryptography algorithms [37-40] have been used to preserve the privacy of aggregated data. In private cryptography algorithms [37, 38], IoT devices encrypt data and forward encrypted data to an aggregator. The aggregator aggregates the encrypted data and sends the resultant data file to a cloud for decryption/decoding. Although the performance efficiency is high using symmetric keys for encoding/ encryption compared to asymmetric keys, symmetric keys still result in a key-compromise attack. Asymmetric key algorithms [39, 40] overcomes key-compromise issues for preserving data aggregation privacy. However, in these algorithms [37-40], either data aggregation is performed at a single aggregator or performed at a remote cloud. A single aggregator/ cloud utilization increases IoT network bandwidth, performance overhead, data collision, and single point of failure risk.

Fog-enabled IoT can overcome such limitations using distributive aggregation and lightweight cryptosystems. Research [9, 13, 16, 41, 42] considered fog-enabled IoT solutions for aggregating and encrypting data. The solutions save the IoT network bandwidth, computational, and memory overhead using data aggregation at fog computing network and forwarding only aggregated results to the cloud.

1.4. Data Replication

Data is one of the most critical resources of any organization, and yet, in many organizations, data protection gets less priority as compared to data management and analysis. Data should be effectively protected from any loss and modification and secured for data availability [43]. Malware attacks, accidental deletion of data, or system hardware failures can lead to data loss. Creating copies of the data and storing it in multiple storage mediums can provide data protection. The process of creating copies and replicating them is known as data replication. It

ensures that the data backup exists for data recovery in case of a system or data breach, system hardware failure, or a catastrophe. Data replication improves network performance, data availability, accessibility, and reliability by making data replicas available at multiple storage mediums [44]. If any malware attack or system hardware fault destroys the data at one storage medium, the accurate data can be accessed from another storage medium.

Generally, data replication includes replica creation, placement, selection, and replacement processes [45]. First, the number of replicas to be created is determined in a replica creation process. Second, the best possible location for replica storage is determined in the replica placement and selection process. Finally, in the case of storage limitations, the replica replacement process can change the replica locality with a new replica [46]. For these processes, static and dynamic methods are used [47]. In the static method, data replicas are created during the data processing setup, and these replicas are unaffected by any changes in replica selection and placement [45]. However, the static method does not comply with any changes due to replica deletion and access patterns of data users [48]. On the other hand, the dynamic method creates replicas affected by the replica's creation/deletion and access pattern changes.

An adversary can either modify/delete data replicas to make data unavailable to end-users or acquire replicas to monitor data-owners patterns and sensor locations in wireless networks. Therefore, the privacy of replicas needs to be preserved as the original data. Data replication privacy has been considered in cloud-based schemes [48, 49]. These schemes consider non-cryptographic measures to replicate data without data encryption. An adversary exposing a few replicated data fragments (unencrypted) would be able to analyze and discover the data patterns and their meanings. Further, performing data replication on a cloud/centralized system increases the computational and storage burden at the computing end, which results in degraded data reliability, scalability, and high bandwidth overhead. Fog-enabled IoT schemes [50-53] have been considered to overcome the cloud/ centralized system issues for data replication.

2. Research Problems

An appropriate architecture to efficiently preserve data aggregation and replication privacy is required due to the significance of data aggregation and replication for IoT applications. Cloud-based schemes [37-40] have not been convenient because of the high network performance overhead, whereas fog-enabled IoT can provide a promising solution for efficient data aggregation and replication privacy.

The solutions [9, 12, 13, 54-56] in fog computing preserves data aggregation privacy for IoT applications. These schemes perform data aggregation on a fog node and then forward the

aggregation results to the cloud. Although these schemes reduce computation, communication, and latency overhead of an overall IoT network compared to cloud-based schemes, the utilization of multiple fog nodes for workload distribution of data aggregation has not been considered. Therefore, these schemes [9, 12, 13, 54-56] are vulnerable to the single point of failure risk and Denial of Service (DoS) attack. In addition, no other fog node is integrated into a network to minimize the fog node's failure probability during the data aggregation process. Furthermore, the schemes [9, 56] are based on heavyweight cryptosystems for preserving data privacy, such as pairing-based cryptography with third party consideration to generate private/ public keys, which increases the performance overhead of fog computing network. Data replication [50-53] has also been considered in fog-enabled IoT to improve the data's performance efficiency and reduce IoT network latency and turnaround time to cloud computing. However, these schemes have not considered preserving data privacy during the data replication process [50-53]. This thesis aims to provide a framework for efficient privacy-preserving data aggregation and replication to bridge the identified limitations in [9, 12, 13, 50-56].

3. Research Questions

The main objective of this thesis is to design a framework for efficient data aggregation and replication to preserve data privacy for fog-enabled IoT applications. Research Question *(RQ) 1* focuses on the design approach and factors for measuring the efficiency and effectiveness of the proposed framework to provide data privacy during data aggregation. *RQ2* aims to optimize the performance efficiency of the proposed data aggregation scheme in *RQ1* using an optimization method. Finally, *RQ3* focuses on the efficiency and effectiveness of a proposed framework for preserving the privacy of data replicas. The following research questions are addressed to design the schemes of the framework.

RQ 1: What design approach and factors can be used for creating a scheme to achieve effective and efficient privacy-preserving data aggregation for fog-enabled IoT?

Sub-questions:

1a. How to model the performance efficient privacy-preserving data aggregation scheme?

1b. Is the proposed privacy-preserving data aggregation scheme efficient to optimize the performance overhead with distributive data aggregation compared to traditional schemes?

1c. Is the proposed privacy-preserving data aggregation scheme effective to preserve the data-owner-defined level of privacy for data aggregation?

RQ 2: What design method can be used to optimize the performance efficiency of the proposed data aggregation scheme for fog-enabled IoT?

2a. How to model time and energy-efficient multi-objective optimization method for data aggregation?

2b. Is the proposed multi-objective optimization method efficient to optimize time and energy consumption compared to the traditional optimization methods?

RQ 3: What design approach and factors can be used for creating a scheme to achieve effective and efficient privacy-preserving data replication for fog-enabled IoT?

Sub-questions:

3a. How to model the performance efficient privacy-preserving data replication scheme using the proposed data aggregation scheme?

3b. Is the proposed privacy-preserving data replication scheme effective and efficient to preserve data replicas' privacy compared to traditional schemes?

4. Research Contributions

This thesis proposes a framework to achieve efficient data aggregation and replication while preserving a higher level of data privacy for fog-enabled IoT. The contributions of this thesis are outlined below:

Contribution 1: An in-depth analysis and classification of privacy requirements are presented for fog-enabled IoT applications. The mapping of the existing works to privacy classification is also provided to distinguish the benefits and improvements that fog-enabled IoT introduces for IoT applications. Finally, the state-of-the-art schemes' analysis is discussed to highlight the research challenges in preserving data privacy of IoT applications in fog computing with mapping to IoT solutions. This contribution is reported in Chapter 2.

Contribution 2: In this contribution, a lightweight Divide-and-Conquer scheme is proposed. The data aggregation scheme considers data processing distribution among fog nodes to preserve data aggregation privacy. A Data division strategy for the Divide-and-Conquer scheme is proposed based on the Level of Privacy (LoP) defined by a data-owner. The authority is provided to the data-owner to define LoP for their data privacy in fog-enabled IoT.

The performance efficiency of the proposed scheme is evaluated in terms of computational, memory, and communication overhead, considering different security parameters and comparison with the state-of-the-art schemes. Ouafi *et al.* & Gope *et al.* privacy

models [57, 58] are also considered to analyse the proposed scheme's effectiveness formally. This contribution is presented in Chapter 3.

Contribution 3: This contribution is for RQ2 to optimize the time and energy consumption of the data aggregation scheme. First, a multi-objective optimization problem is formulated with a joint objective to optimize time consumption and energy consumption for fog-enabled IoT. Inspired by Xu et al.'s task offloading formulation [59], time and energy consumption for data aggregation is then formulated. Second, the multi-objective optimization method is defined based on NSGA-III (non-dominated sorting genetic algorithm III) to develop optimal solutions for data aggregation. In the NSGA III method, the consideration of reference point methods: Simple Additive Weighting (SAW) [60] and Multi-Criteria Decision Method (MCDM) [61] selects the optimal solution for time and energy consumption of each fog node in a fog-enabled IoT network.

Comprehensive simulations and systematic experiments are conducted to demonstrate and evaluate the performance efficiency of the proposed scheme in terms of evaluation metrics, including the degree of workload imbalance and standard deviation. In this contribution, the performance efficiency of the proposed method is also evaluated with state-of-the-art methods. This contribution is reported in Chapter 4.

Contribution 4: A data replication scheme is proposed to efficiently process data replicas and effectively preserve the privacy of data replicas. This contribution is for RQ3, which is based on the proposed system model of the data aggregation scheme. First, a data replica creation scheme efficiently selects fog nodes for data replica creation. The replicas creation in the proposed scheme considers a basic replica creation model in [62], and the model is extended according to the requirements of the system model. The proposed scheme then generates data replicas based on the data-owner's Level of Privacy (LoP). Second, a data replica placement scheme is proposed to store data replicas in a distributive manner. The replica placement scheme considers a priority level based on the LoP and service capacity of fog nodes.

Third, the time complexity, privacy, and time series analysis are conducted for the proposed schemes. For privacy analysis, Shacham and Chen *et al.'s* privacy models [63, 64] is considered to perform the formal privacy analysis of the proposed schemes. For the time series analysis, the Autoregressive Integrated Moving Average (ARIMA) model [65] is adopted to verify the accuracy of the proposed replica creation and replica placement schemes. In this contribution, the performance efficiency evaluation of the proposed schemes is presented in terms of influential parameters, computational, memory, and communication overhead compared with the existing schemes. This contribution is presented in Chapter 5.

These four contributions of the thesis address all the RQs outlined in Section 3. In addition, the thesis methodology adopted for developing the RQs is presented in Section 6.

5. Publications

Journal Publication

- Sarwar, Kinza, et al. "Lightweight, Divide-and-Conquer privacy-preserving data aggregation in fog computing." *Future Generation Computer Systems* 119 (2021): 188-199.
- Sarwar, Kinza, Sira Yongchareon, Jian Yu, and Saeed ur Rehman. " A Survey on Privacy Preservation in Fog-Enabled Internet of Things". *ACM Computing Surveys.* (Accepted)

Conference Publication

- Sarwar, Kinza, Sira Yongchareon, and Jian Yu. "A brief survey on IoT privacy: taxonomy, issues and future trends." *International Conference on Service-Oriented Computing*. Springer, Cham, 2018.
- Sarwar, Kinza, Sira Yongchareon, and Jian Yu. "Lightweight ECC with Fragile Zero-Watermarking for Internet of Things Security." *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018.

Under Review Articles

- Sarwar, Kinza, Sira Yongchareon, Jian Yu, and Saeed ur Rehman. " Joint Optimization of Time Consumption and Energy Consumption for Data Aggregation in Fog-enabled IoT networks". *IEEE Internet of Things Journal.* (Submitted)
- Sarwar, Kinza, Sira Yongchareon, Jian Yu, and Saeed ur Rehman. " Efficient Privacy-Preserving Data Replication in Fog-enabled IoT ". *Future Generation Computer Systems.* (Submitted)

6. Research Methodology

This thesis adopts simulation and analytical modelling to achieve the objectives of the proposed framework. Figure 2 illustrates the research methodology.

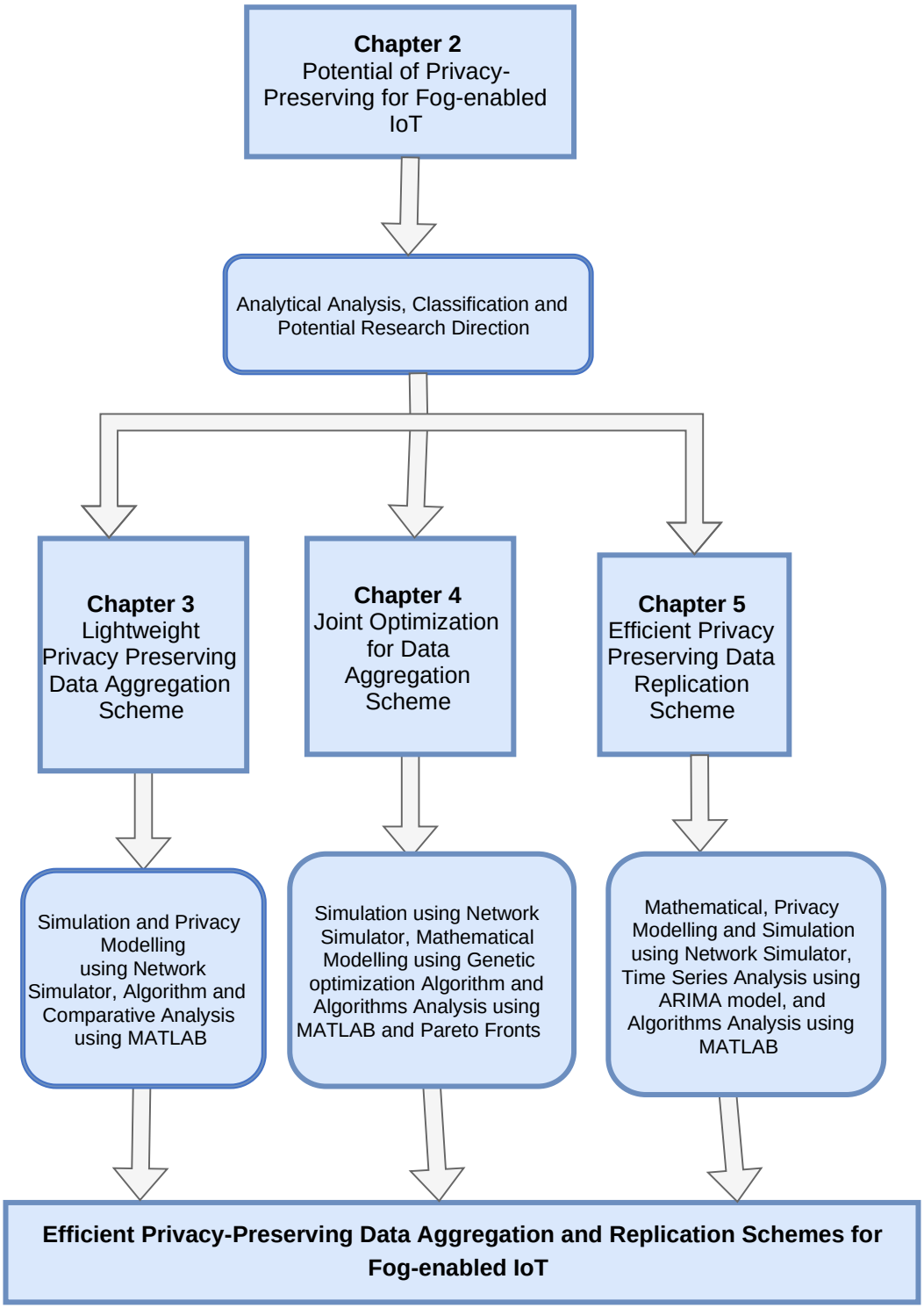


Figure 2 Research Methodology

Considering the in-depth analytical observations from the literature, Chapter 2 first explores the potential of privacy preservation for fog-enabled IoT in terms of performance efficiency and privacy effectiveness. Then Chapter 3 proposes a privacy-preserving data aggregation scheme based on analytical observations from the literature. Chapter 4 provides the time and energy consumption of the proposed data aggregation scheme and apply optimization algorithms to reduce the computational, memory, and communication

consumptions. Finally, Chapter 5 provides a design of a data replication scheme for the proposed data aggregation scheme considering the privacy of data replicas during data aggregation.

For implementing and evaluating the performance efficiency and effectiveness of the proposed schemes, this thesis adopts computer-based simulation, mathematical and privacy modelling, and statistical analysis. Different network scenarios are considered in Network Simulator to generated results for statistical analysis (Chapter 3 to 5). Mathematical modelling is also carried out in Network Simulator (Chapter 4 and 5). The genetic model is integrated with Network Simulator for mathematical modelling of data aggregation optimization (Chapter 4). Further, the MATLAB tool is used to evaluate the performance of the proposed schemes (Chapter 3 to 5). Correlation between sensitive parameters influencing the time and energy consumption is evaluated using Pareto fronts and box plots in MATLAB (Chapter 4). Box plots are also used to analyze the correlation between sensitive parameters influencing the data replication scheme (Chapter 5). In addition, Autoregressive Integrated Moving Average (ARIMA) model [65] is considered to verify the accuracy of the data replication scheme.

7. Thesis Structure

The thesis is communicated in six Chapters, and Figure 3 illustrates the overall structure of this thesis. Chapter 1 gives an overview of the research work. Chapter 2 presents a review of literature, which begins with an introduction of privacy requirements for IoT applications with the evolution of privacy-based schemes. The following sections in this Chapter provide the classification of privacy-based schemes in IoT and the mapping of schemes to fog-enabled IoT models. Finally, the open research challenges motivate the identification of the research gaps in preserving the privacy of IoT applications in fog-enabled IoT networks.

Chapter 3 presents the scheme for preserving the privacy of data aggregation. It provides an in-depth analysis of the proposed scheme with system and adversary models. The Chapter also gives the security analysis and performance evaluation comparison with the existing schemes. Chapter 4 presents the optimization scheme for the proposed data aggregation scheme. The Chapter outlines the system and problem formulation followed by the multi-objective optimization method. The performance evaluation considering statistical analysis and comparative analysis with state-of-the-art schemes is also presented in this Chapter.

Chapter 5 presents the data replication scheme with system and adversary models and time complexity analysis. It also provides experimental results, privacy, and performance analysis to evaluate the efficiency of the proposed scheme. Then Chapter 6 concludes the

research by summarizing the contributions and limitations of the proposed framework with recommendations for future research.

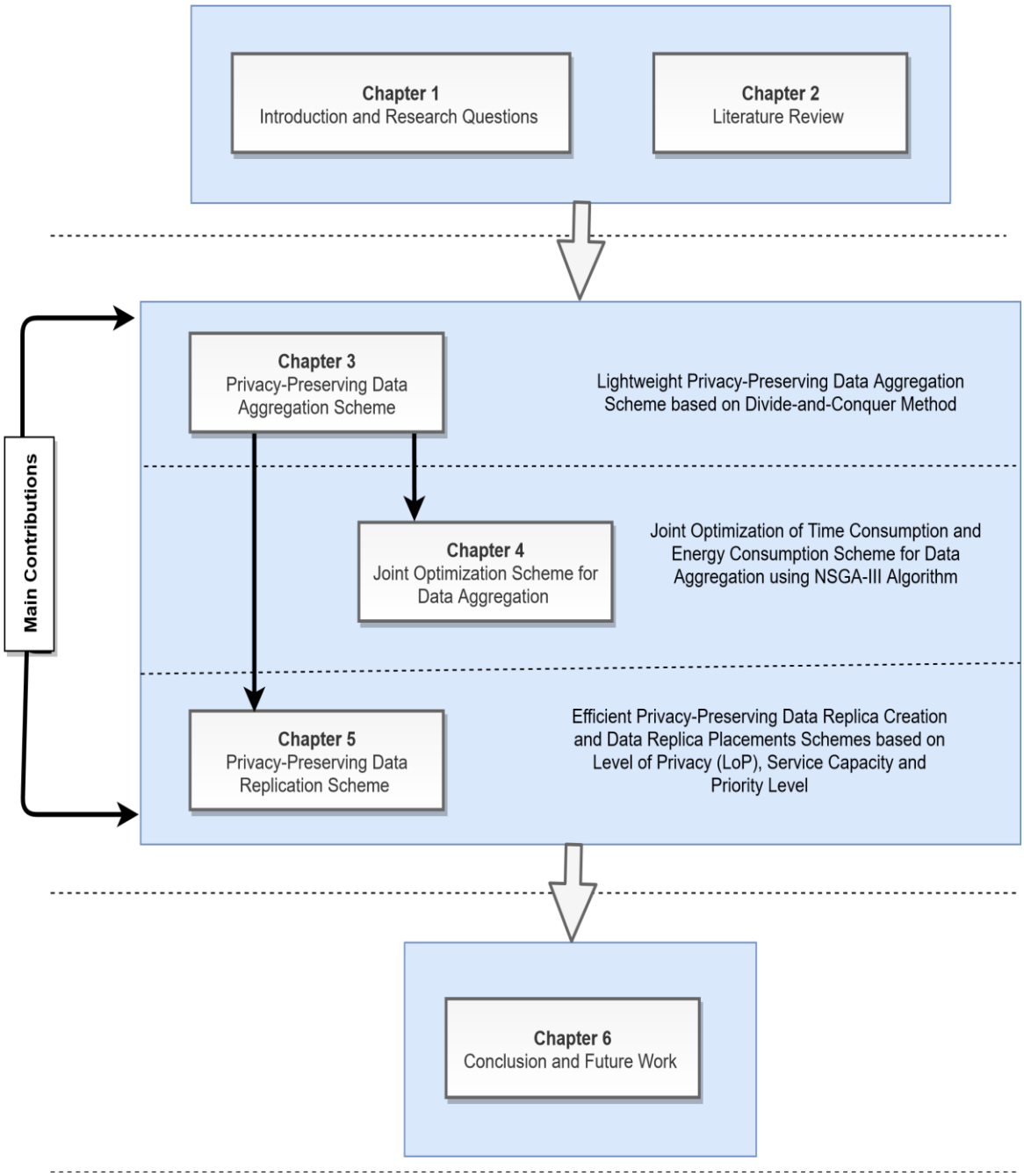


Figure 3 Thesis Structure

'We are all now connected by the Internet, like neurons in a giant brain'.

---Stephen Hawking

Chapter 2: A Survey on Privacy Preservation for Fog-Enabled IoT

Abstract

The Internet of Things (IoT) is a ubiquitous network that connects various kinds of network and mobile devices, sensors, data processing devices, and software platforms and applications to the Internet, such as vehicles, home appliances, and medical apparatuses. Despite the rapid growth and advancement in the IoT, there are critical challenges that need to be addressed before the full adoption of the IoT. Individual privacy is one of the hurdles towards the adoption of IoT. Individuals have concerns over the potential misuse of their data and identity in IoT applications. Several researchers have proposed different approaches to reduce or eliminate privacy risks. However, most of the existing solutions still suffer from various drawbacks, such as huge bandwidth utilization and network latency, heavyweight cryptosystems, and policies that are applied on sensor devices and in the cloud. To alleviate such drawbacks, recently, a concept of Fog-enabled IoT has been introduced, which extends cloud computing to the IoT network edge providing low latency, scalability, computation, and storage services. In this survey, we first aim to comprehensively review and classify the privacy requirements for a better understanding of privacy implications in IoT applications. Based on our classification, we highlight ongoing research efforts and limitations of the existing privacy-preservation techniques for the IoT applications and map the existing IoT schemes with fog-enabled IoT schemes to elaborate on the benefits and improvements that fog-enabled IoT brings to preserve privacy in IoT applications. Furthermore, based on our study, we enumerate research challenges in fog-enabled IoT and reveal future directions for fully preserving the privacy of IoT applications.

This Contribution has been accepted for publication in ACM Computing Surveys Journal

1. Introduction

The wide deployment of the Internet of Things (IoT) has resulted in the interconnectivity of smart things on a worldwide scale [4]. It is estimated that IoT devices connected to the Internet will reach 41.6 billion by the year 2025 [6]. Also, the amount of data generated by IoT devices is increasing in size [4]. Due to the device's limited battery, processing, and memory resources, the generated data is offloaded to cloud computing for computation, analysis, and long-term storage. Offloading IoT data to cloud computing increases network overhead, including latency and bandwidth. Compared with cloud computing, fog computing reduces the network overhead and congestion by alleviating the workload from a cloud to the edge of a network close to the IoT devices [66].

Fog computing is a distributed computing paradigm where IoT devices' computation and analysis are performed at the network edge [10], as shown in Figure 4. As the concept of fog computing is introduced for IoT applications, the terminology fog-enabled IoT is used interchangeably [16]. Along with the IoT resources distribution, fog-enabled IoT also considers the user access controls' and data policies' management, data ownership, and security credentials [67]. Further, collecting data from thousands of smart devices can be aggregated using multiple aggregators in fog-enabled IoT. Data aggregation upgrades the usability and performance of IoT applications.

Besides this, the fog-enabled IoT plays an essential role in mitigating the privacy issues of IoT applications. The concerns of data owners over the misuse of their data and identity place privacy as one of the critical challenges in IoT applications. IoT devices not only collect the data owner's identity data (name, telephone, number, or address) but monitors their activities, behaviors, health, genome, and social interactions [68]. For preserving such data collected by IoT devices, robust privacy algorithms (e.g., public and private cryptography) are required. Due to the limited IoT resources [23, 24], the implementation of efficient privacy algorithms on such IoT devices is critical. Keeping the IoT devices' limitations in mind, researchers in [25-31] considered remote service providers or Cloud to process privacy algorithms. The consideration of remote service providers or the Cloud increases the potential risks of IoT applications' information leakage and IoT network bottleneck.

In contrast, fog-enabled IoT provides a promising medium for preserving the privacy of data collected from IoT devices. While fog nodes reside nearby end devices, fog-enabled IoT mitigates the chances of eavesdropping on IoT data [13]. The processing workload and data storage of IoT devices with heavyweight security measures can be transmitted to fog nodes to reduce the performance and network overhead.

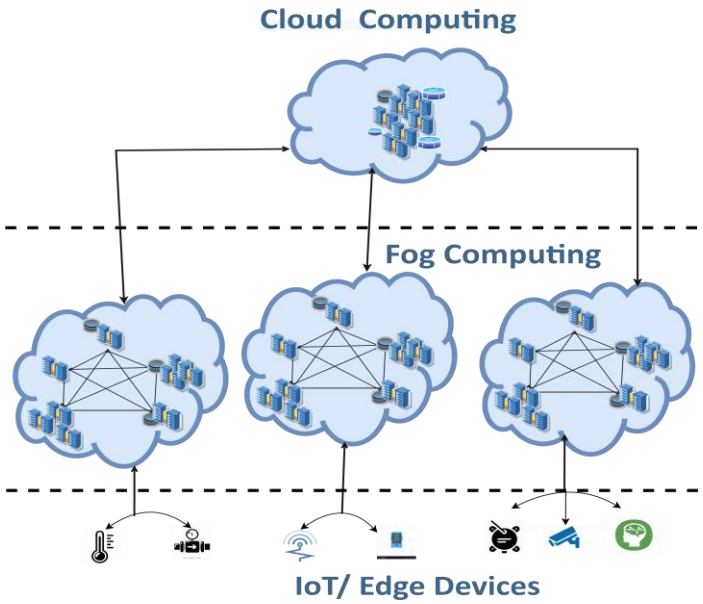


Figure 4. Fog Computing Environment

Several surveys have highlighted the security and privacy concerns of IoT applications in cloud-based IoT solutions [69-73]. Systematic literature reviews have addressed the awareness of generic security and privacy issues in IoT [69, 70]. These reviews analyzed the existing solutions and provided recommendations to involve security policies and standards. The reviews also addressed the Identity and location privacy in cloud-based IoT solutions for the Electronic Health Record (EHR) system's robustness.

Due to the advancement of technology, surveys [71-73] considered the additional security challenges of IoT applications in cloud-based IoT solutions. The authors discussed programming analysis techniques and security communities to best fit in for additional challenges. They claim that the focus of the programming analysis techniques is mostly on smart homes, and there is a lack of diverse IoT applications that need to be addressed in the future. Although this review is an in-depth Application Programming Interface (API's) analysis for privacy, the APIs limitations for IoT systems have not been considered. The limitations are the high-performance overhead without optimization strategies.

The studies [69-71, 74, 75] only focused on addressing overall security issues and countermeasures in cloud-based IoT applications, ignoring the in-depth analysis of privacy regarding issues and countermeasures. Recommendations from [69-71, 74, 75] emphasize that robust privacy-preserving algorithms, including lightweight attribute-based encryption and fully homomorphic techniques, should be applied to IoT application data in cloud. As discussed before, applying strong privacy algorithms in cloud computing results in high network latency, an increase in processing burden, and network communication vulnerabilities. Due to this, cloud computing may result in a weak privacy system for IoT applications. In all of the afore-discussed

literature reviews [69-75], there have been no discussions and recommendations on mitigating cloud-related concerns, for example, network vulnerabilities, high latency, processing burden, third-party involvement for providing strong data privacy.

Introduction to fog computing for IoT applications as a middleware can overcome the cloud-based IoT concerns as discussed in [76]. The study focused on solutions that fog computing has introduced for data security. No in-depth privacy analysis to mitigate the cloud-based IoT privacy issues in fog computing has been provided. Similarly, a review provided a discussion on security and privacy challenges in fog computing [77]. Another review highlighted potential security risks in microservices-based fog applications [78]. The reviews only focused on security-based risks, whereas the discussion on user's data privacy protection has not been highlighted. The reviews [79, 80] provided a security-based solution for IoT applications in fog computing. There is no detailed discussion on preserving privacy challenges and future trends to enhance privacy in fog-enabled IoT. Also, a review [81] provided encryption security technologies in cloud and fog-enabled IoT systems. This review summarizes the security systems by highlighting the future aspects of the user's privacy protection. The review is a generic introduction to architecture and layers in fog-enabled IoT. The exhaustive research to highlight the pros and cons of fog-enabled IoT security and privacy systems is missing.

The surveys [76-78, 81] either considered generic privacy issues and solutions in fog-enabled IoT or no in-depth classification and up-to-date summarization of privacy in fog-enabled IoT. Furthermore, the full utilization of privacy-based IoT platforms not only depends on addressing the generic privacy requirements with challenges, countermeasures, and concerns but also on analyzing in-depth privacy problems and solutions with an enhancement to IoT application domains. In our study, we comprehensively reviewed and analyzed literature related to IoT privacy preservation issues and solutions with a focus on fog-enabled IoT.

We summarize our main contributions as follows.

1. We present an in-depth analysis and classification of privacy requirements in fog-enabled IoT applications.
2. We identify the state-of-the-art solutions and research challenges in preserving privacy in IoT.
3. We map the existing works to our privacy classification for distinguishing the benefits and improvements that fog-enabled IoT introduces in IoT applications.
4. We highlight open research challenges and potential future directions.

The remainder of this paper is organized as follows. Section 2 introduces privacy requirements in IoT along with the evolution of privacy-based schemes in IoT and fog-enabled IoT. Section 3 provides an in-depth analysis and classification of privacy-based schemes in IoT

and the mapping of the state-of-the-art IoT-based models and fog-enabled IoT models. Section 4 identifies and discusses open research challenges, and the paper is concluded in Section 5.

2. Privacy requirement In Fog-enabled IoT

This section provides the classification of privacy-preserving requirements in fog-enabled IoT. The section also highlights the number of solutions proposed for each privacy-preserving requirement to give an overview of research contributions over a decade. In sub-section 2.1, we provide an overview of privacy preservation evolution regarding IoT and fog-enabled IoT. Then, we discuss a comprehensive review related to privacy requirements as well as threats, vulnerabilities, and attacks associated with these requirements in sub-section 2.2.

2.1. Evolution of Privacy in IoT and Fog-enabled IoT

Research in the privacy preservation of IoT data has gained significant attention in this era. Figure 5 depicts the research contributions of related scientific communities towards the evolution of privacy preservation in the IoT and fog-enabled IoT from 2010 to 2020. Before 2010, initial research has been carried out to identify and analyze the importance of security and privacy in the IoT application. The full realization of privacy for IoT applications has been considered after 2010. 2015 was a significant year for considering all privacy requirements in IoT applications. Before then, the main focus was on designing privacy standards and policies of IoT data.

In 2010, symmetric techniques for identity privacy of customer data has been proposed for IoT applications [82]. Only 0.1% of the proposed models discussed the privacy policies associated with IoT user's behaviors and actions, and location privacy. The enhancement in privacy policies

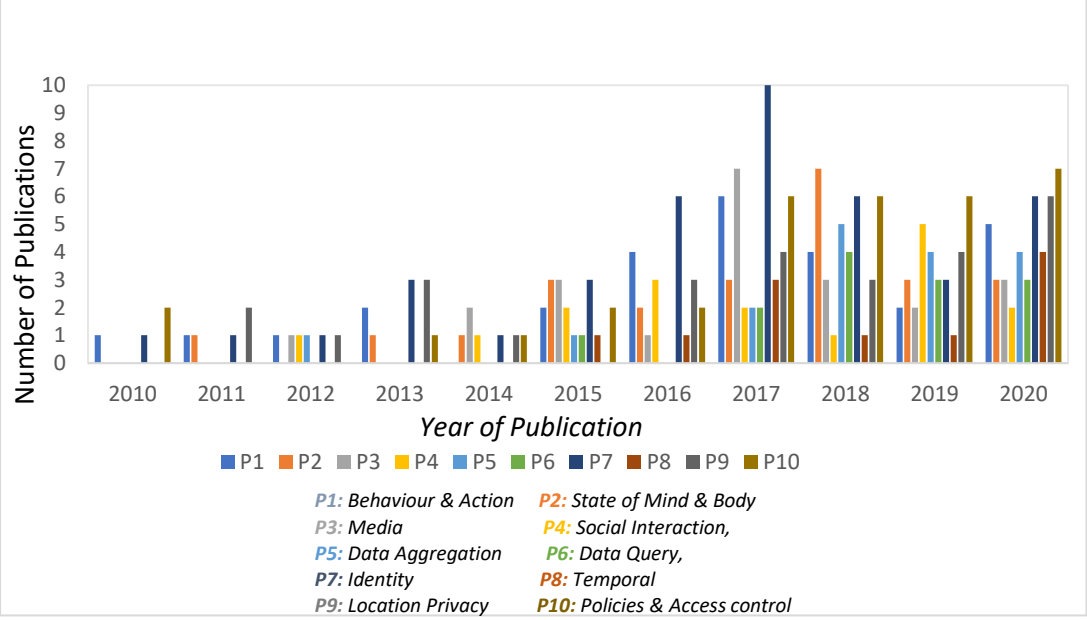


Figure 5 Publications on Privacy Preservation in IoT and Fog-enabled IoT in the Literature from 2010-2020

with legal obligations for IoT users' identity, location, and state of body and mind privacy has been deployed in 2011 [83-85]. In 2012, identity risk management with data aggregation, media, and social interaction privacy was introduced [37, 86, 87]. Approaching 2013 is when there were opening diversifying privacy ways in IoT with enhancements in privacy policies and methods for identity and location privacy, for example, oblivious methods and attributes signer techniques [88-92].

In 2014, a collaboration among cryptographic techniques has been carried out to achieve a state of body and mind, media, and social interaction privacy with high-performance computing efficiency [93-95]. Advanced levels of privacy measures with semantic ontologies for behavior and action, media, and query privacy have been introduced in 2015 [25, 26, 39, 96-101]. In 2016, there was a noticeable increase in the number of publications [33, 102-105] for identity, location, and behavioral and action privacy. These publications aimed to preserve the privacy of IoT data using fog computing for IoT applications. 2017 can be considered a technology advancement year, focusing on privacy considerations in IoT applications, including temporal media and identity privacy [56, 106-119]. Moreover, query and location privacy concepts have been applied in the fog-enabled IoT platforms [67, 120, 121].

Blockchain has been adapted for identity privacy in IoT applications in 2018 as the technology overcomes major limitations of centralized data processing and secret sharing [41]. Further, the range of research expanded for fog computing with the introduction of data ownership, forward secrecy, and enhancement in data query, data aggregation, state of body and mind, location privacy, and data owners' access control [7, 40, 113, 122-138]. In 2019, new studies [139-141] introduced related to strengthening the state of body and mind privacy for the healthcare sector. Also, the techniques for data transparency with the Health Insurance Portability and Accountability Act (HIPPA) and General Data Protection Regulations (GDPR) [139-141]. Social interaction, data aggregation, and identity privacy for IoT applications such as smart devices have been considered in fog computing [7, 139, 140, 142-145]. Data query and identity privacy using the blockchain concept had also been highlighted in other researches [7, 141, 142, 144-146]. The number of research for each privacy requirement has shown a tremendous increase in 2019 [147-171]. In early 2020, more focus has been on blockchain for preserving behavior and action, identity, and location privacy [148, 149, 154, 155, 159, 164-166, 169-171]. We can also depict from Figure 5 that in 2020 there is an increasing number of publications related to temporal, location and policies, and access control privacy [147, 150, 154, 155, 158, 159, 161, 165, 167-171]. From 2017 onwards, the number of publications considered platforms to deal with privacy requirements such as media, social interactions, data aggregation, data query, and temporal in both IoT and Fog-enabled IoT. The focus was not only on basic privacy

preservation (i.e. Identity, location, and behavioral and action privacy).

2.2. Classification of Privacy Requirements in Fog-enabled IoT

To understand the privacy issues and countermeasures of fog-enabled IoT in detail, the analysis of the privacy requirements is presented first, which serve as the building blocks of preserving privacy in IoT applications. Figure 6 depicts the privacy requirements based on the existing literature under our study and we group the requirements into two main categories: Content privacy and Context privacy. Discussions on each of the content privacy requirements with related concerns are elaborated in sub-sections 2.2.1. In sub-section 2.2.2, we discuss the requirements of content privacy in detail.

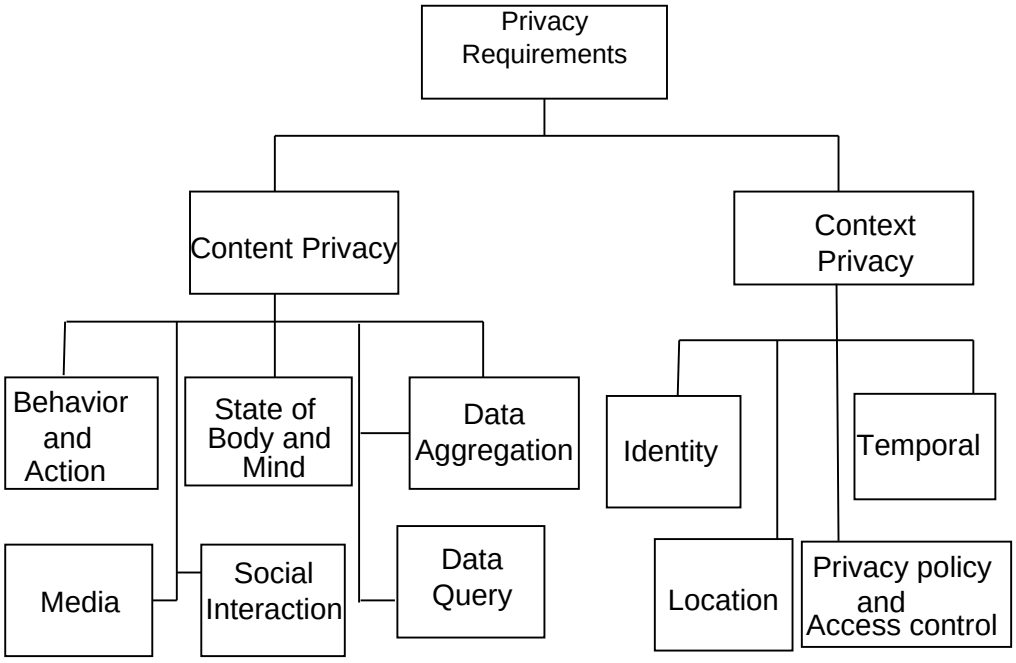


Figure 6 Privacy Requirements in Fog-enabled IoT

2.2.1. Content Privacy

In wireless communication, the terminology 'content' refers to the set of data that is used to deliver a particular message to the recipient. The data set may include particular actions, habits, health, behavior, or social conversations of an individual or group of individuals, or organizations' infrastructure [72]. The protection of such content from eavesdroppers and attackers enables content privacy. Content privacy needs to be protected from apparently two types of adversaries. One is an external adversary, which eavesdrops on data communication between sensor nodes [71]. Another one is an internal adversary as a node participating in the communication, manipulated and captured by an adversary [71]. Although a typical approach to content protection can be achieved using authentication and encryption mechanisms. These mechanisms alone cannot guarantee content privacy. Also, the adversary may have the potential access to a private set of data, including thought and emotion, body and state of mind,

and social interactions [172]. Therefore, it is of utmost importance to understand all the aspects of content privacy requirements at the early stage of fog-enabled IoT design, as follows:

- **Behavior and Action:** Content relevant to individuals' actions, habits, hobbies, and purchase patterns are aspects of behaviors and actions. The exploitation of such content may result in threatening data user's lifestyles [173-175]. For example, the data users' credit/debit cards or online shopping details may be shared with retailers to identify users' interests and profiles. The retailers can also use profile information to forward related advertisements to the users [87, 175, 176]. Introducing solutions that can mislead attackers from disclosing information about an individual's behaviors and actions is one of the prior requirements.
- **State of Body and Mind**: Privacy of state of body and mind is equally important as that of behavior & action. State of body and mind encompasses an individual's mental states, health, biometric, emotions, genome, and opinions protection from attackers [177, 178]. Violation of such privacy may lead to prosecutions by authoritarian regimes, discrimination by insurance companies and employers [172].
- **Media:** The scope of content privacy is not only limited to personal actions, health, and patterns but also image, video, and audio of individuals, businesses, and assets. Media includes camera footage, CCTV, and video uploads to the Internet [179, 180]. The distribution or creation of user-related media without users' consent can result in a privacy violation [172]. Consequently, users may not consider the network reliable for media sharing and be reluctant to upload and forward images, videos, or audios from smart devices to the Internet.
- **Social Interactions:** Individuals or group's conversations on a social media platform may expose a person's identity, interactions metadata, health, opinions, and conversation time durations [94, 175, 181]. In short, social interaction is the Florilegium of above mentioned three privacy requirements as it combines all of them. Therefore, achieving social interaction privacy is a challenging task in the IoT and Fog-enabled IoT network.

The following two situations are where content privacy is not sufficiently covered with these basic but still necessary requirements.

- **Data Aggregation:** A process of combining information from different data sources (which may belong to any of the requirements mentioned above) into a condensed message. The additive and non-additive aggregation methods consist of sum, average, maximum, and minimum [182, 183]. An example of a data-aggregation operation is shown in Figure 7, which consists of aggregators for collecting data from different sensors and aggregating data altogether. Data aggregation significantly reduces the performance overhead of sensor nodes for processing data. Data aggregation is an essential process for maintaining or increasing the durability and efficiency of sensor networks [33, 72, 160, 184-186]. Thus,

preserving the privacy of data during data aggregation is a considerable privacy requirement.

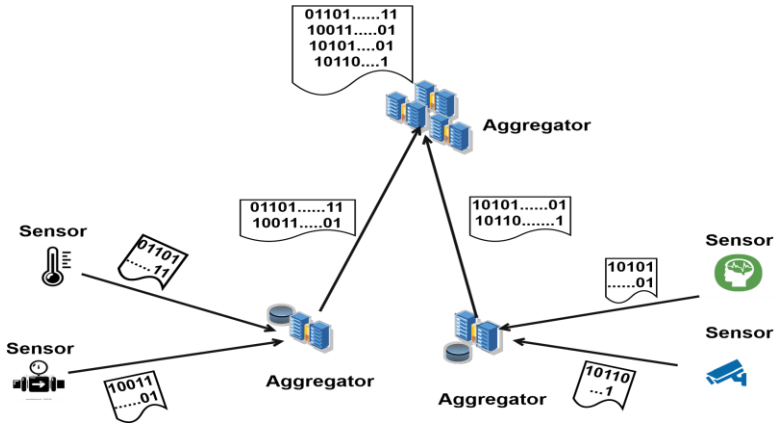


Figure 7 Example of Data-aggregation

- **Data Query:** A data-driven query is a process that a user requests on a data source/ processing node for their wanted data while ensuring that the data of sensor nodes is kept confidential and intact [72, 145, 146, 187]. Data users have concerns about their data queries, whether they are kept private or not. Attaining data query privacy has become an essential task in content privacy. Since the performance overhead grows exponentially with the widespread adoption of IoT sensor devices, achieving data query privacy has been more technically challenging [71, 145].

2.2.2. Context Privacy

Context privacy focuses on features of the communication that may be exploited to infer contents within the communication. The contextual features of the communications include the size and number of transmitted messages, the time and rate at which messages are being sent, the frequency spectrum used by the nodes, the source, and the destination of transmissions [71, 72, 145, 146]. IoT applications focus mainly on protecting the data set using encryption techniques during the data content transmission and ignoring identity and location privacy. The ignorance of such privacy results in revealing the transmission data's patterns, which can be traceback to the original data set [72, 122, 141, 188]. Due to the limited IoT resources, protecting context privacy is challenging. In order to understand context privacy in more detail, we divide context privacy requirements into four categories:

- **Identity Privacy:** Users wish that their identity remains anonymous during data transmission and processing in wireless sensor networks [75, 189]. A pseudonymity concept has been introduced to provide identity anonymity to users. The concept is based on a persistent

identifier to ensure that a service can be offered from initiation to completion without revealing any user's identity and location [70, 190]. Pseudonym generation and periodic updating can increase an intolerable computational cost for resource-constrained IoT nodes. Also, pseudonyms cannot resist physically dynamic tracing attacks for location identification. Fog-enabled IoT can be used as a paradigm to overcome pseudonyms location identification and computational constraints challenge for operating in a standardized manner across multiple IoT applications [70].

- **Temporal Privacy:** The packets transmitted to report a data generation or processing event have associated contextual spatial-information such as packet creation time and location with them [191]. An attacker can eavesdrop on the transmission of packets to gain a rough estimate of contextual Spatio-temporal information through traffic analysis even though the event data in the packet is encrypted [72, 181, 191, 192]. An attacker can use this Spatio-temporal information to disrupt the IoT network's proper functioning [124, 191]. Spatio-temporal information also estimates a correlation between the successive events in a series [193]. Initially, an attacker is unaware of the event creation time. However, eventually, by monitoring traffic patterns near the event's sink node, an attacker can deduce the temporal information related to an event [191, 194].

  Temporal privacy requires to prevent an attacker from analyzing the traffic traces reporting events to infer the exact Spatio-temporal information about the occurrence of an event [174, 192]. The concept of anonymity has been adopted to preserve temporal privacy. In this concept, data packets delay the information messages instead of batching all information together in packets transmitted over the network [192]. Packet delay using the buffering technique at the intermediate nodes has been considered in sensor networks [195]. The reason for buffering at the intermediate node is to obfuscate the temporal information from an attacker. Random delays reduced the correlation time between packet generation and transmission to sink nodes [191, 196]. In these techniques, protocol-aware attacker and node indistinguishability to preserve temporal privacy have not been considered. An additional protocol for maintaining buffer and delays has added more computational overhead to the network. Therefore, sustaining Spatio-temporal privacy with high-performance efficiency in a vast IoT paradigm is a critical task.

- **Location Privacy:** In IoT applications such as for Healthcare, Military, Assets, and Radiation, it is important to keep information private about the location of sensor nodes, which generates or transmits the data of the data-owner [105, 122, 197, 198]. If the source information is exposed to an attacker, it will allow an attacker to backtrace entire packet routes and locate source data usage in the network. The source location problem was first identified in the panda hunter game [199]. The location and activities of pandas were

continuously monitored using wireless sensor networks [198, 199]. Using wireless sensors antennas', an attacker could analyze the broadcast patterns between the sensor nodes. From this broadcast information, an attacker could trace back packet routes leading to the source node location and thus the pandas [106, 198].

Another approach for source location problems was introduced in military environments [199]. In this approach, soldiers were wearing sensor nodes to relay information packets to the sink node, such relay of information packet can be compromised by an attacker. To identify the location of soldiers, an attacker may use a spectrum analyzer to trace packets in the network. Shortest path routing [155] and phantom single-path routing [156] schemes have been introduced to tackle such issues. Packet routes between source and sink nodes are short, which can be easily traced back by an attacker [199].

Source node located near the sink node and using a single fixed routing path causes poor location privacy [200]. Using randomly selected routes and multiple sink nodes [108] introduces additional communication and packet delivery overhead. The concept of pseudonyms can also be used to hide source node locations [70]. The location cannot be directly hidden using pseudonyms as the traceback of packet routes can lead an attacker to observe the location and identify the real identity of the source [72, 201].

- **Privacy Policy and Access Control:** Full privacy protection cannot be achieved without identifying and specifying privacy policies, procedures, and individual access to protected data. Policies and procedures provide guidelines to understand the appropriate access, use, rights, and disclosure of protected information [202, 203]. Privacy policies may include descriptions or rules regarding individuals' access to protected data, amendment of data, requests for restrictions of data use, and disclosure of data [70, 204, 205].

  Setting up privacy policies and procedures for IoT devices with limited computational and memory resource availability is challenging. Because setting most of the policies and procedures in IoT applications are based on heavyweight-driven ontologies, rules, and behavioral modeling. Further, approaches for designing and implementing policy-driven rules, ontologies, and behavioral modeling for IoT devices are at infant stages. The nature of the IoT devices varies with the applications of use [205]. For example, some IoT devices act as mediators serving only data transmission tasks [204], whereas a few devices are used for data capturing and processing purposes. The context in which IoT devices align with policies and procedures regulation differs [137]. For example, an IoT device initially designed for lifestyle monitoring is only subjected to lifestyle policy regulations and constraints [206]. The device cannot incorporate with medical appliances policies.

  The policies and procedures regulation enforcement mechanism is unlikely to be possible to deploy across different IoT applications. For example, a front-end application may

provide access control to individuals for privacy setting modification. In contrast, cloud storage may only provide access guarantees per application, not to individual users of the application [137]. Therefore, setting up policies and access control for each IoT device and application in different domains with optimized performance efficiency is an essential task [70].

Based on the above discussion, we can understand that achieving privacy in fog-enabled IoT platforms successfully relies on the above-discussed privacy requirements. Keeping in mind resource allocation, storage, and memory capabilities while designing an IoT platform or application with full consideration of all the requirements is crucial. Furthermore, assuring that the design is invulnerable to security and privacy threats and attacks adds more complexity. Therefore, the following Section aims to analyze the state-of-the-art privacy-based schemes and provides an in-depth analysis of various considerations in designing effective privacy schemes for IoT applications in IoT and fog computing platforms.

3. Review of Privacy Preservation in IoT and Fog-enabled IoT

This section highlights the challenges that traditional IoT solutions face in preserving privacy and how fog computing can overcome those challenges of IoT applications. In this section, the literature review of state-of-the-art IoT and fog-enabled IoT schemes for preserving privacy is provided. The state-of-the-art schemes are discussed according to the classification of privacy requirements. The section consists of three subsections. In sub-section 3.1, we provide a comprehensive review of privacy-based schemes in IoT applications based on the privacy requirements classification discussed in Section 2, which summarizes research work being carried out without fog computing utilization. The second sub-section 3.2, then reviews fog-enabled IoT schemes. Lastly, sub-section 3.3 discusses the limitations of fulfilling privacy requirements in IoT and a systematic mapping between the existing works in IoT and fog-enabled IoT, and improvements that fog computing introduces to the IoT paradigm.

3.1. Review of Privacy Preservation in IoT

Preserving content and context privacy at IoT's four main axes such as embedded devices, real-time operating systems (RTOS), network protocols and the Internet, and offload storage or processing center is challenging [207]. The distribution of wireless sensor data between main axes makes it even harder for data owners, who can be an organization, an individual, or a group of individuals to retain their data privacy. The data is transmitted, processed, and stored in IoT using a different set of protocols and procedures, which may be vulnerable to a certain type of attack and may expose an individual's private data to an attacker. Therefore, we identify, discuss, and suggest privacy principles from the end-to-end view for IoT applications. Table 1

categorizes various application domains of privacy-based IoT schemes for the privacy requirements discussed in content and context privacy sub-sections 3.1.1 and 3.1.2, respectively.

Table 1 Privacy-based IoT Application Domains

Category	Sub-Category	Application Domain
Content Privacy	Behavior and Action	Smart city [56, 96, 118, 123, 208], Healthcare system [209], Authority management system [102, 125], Smart homes: [116]
	State of Mind and Body	Healthcare system: [85, 126, 209], Authority management system [125] , Video storage system: [127] , Smart city: [128, 139], Smart home: [138]
	Media	Smart parking system: [95], Smart city: [95, 180], Content-centric networks: [111], Smart home:[110, 112-114], Video storage system: [127]
	Social Interaction	Augmented reality devices: [94, 210, 211]
	Data Aggregation	Wireless sensor network: [93], Healthcare system: [103], Smart grid: [40]
	Data Query	Smart city: [212, 213], Smart home: [126]
Context Privacy	Identity	Smart city: [123, 128, 136, 208, 214, 215] , Smart home: [112-114], Wireless sensor network: [93, 104], Healthcare system: [216], Content centric networks:[111]
	Temporal	Smart city: [123]
	Location-	Smart city: [123], Wireless sensor network: [93], Smart homes: [112, 114, 116]
	Privacy Policy and access control	Smart home: [138], Smart city: [101, 119, 204, 217]

3.1.1. Content Privacy

Researchers and practitioners have used different approaches to achieve content privacy in IoT applications. We discuss here possible solutions adopted for each of the content privacy requirements classified in Section 2. Table 2 presents an overview of the schemes providing requirements of content privacy with features including privacy techniques, privacy level, performance levels including communication, computational and memory overheads. This table is discussed in detail in subsequent requirements of content privacy.

- **Behavior and Action:** For protecting the behavior and actions of individuals, anonymization and obfuscation solutions on IoT platforms have been provided [96, 103, 114, 116, 117]. For example, in a smart city, a security-aware automatic fare collection system has been introduced by anonymizing smart card unique identifications to conceal linkages with cardholder's tagging and location patterns [96]. The smart card and RFID readers are nearby, which reduces the communication and computational overhead of smart card

anonymization. Although the proposed solution has preserved the privacy of the location patterns between the smart card and RFID reader, data content between a public transport system and RFID readers is transmitted in raw format. No encryption Algorithm has been applied to keep RFID readings secure and private, which results in a low privacy preservation level, as pointed out in Table 2. The memory load on the back-end system increases for periodically generating keys to keep the communication link between the smart card and RFID reader secure.

Table 2 Techniques for Content Privacy Preservation in IoT

		Privacy Level			Communication overhead			Computational overhead			Memory overhead		
Reference	**Privacy Technique**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**
					Behavior and Action								
[96]	Mutual Authentication	✓			✓			✓					✓
[117]	SHA1		✓		✓			✓			✓		
[114]	AES, BAN logic		✓			✓				✓			✓
[116]	Diffie Hellman, SHA1			✓			✓				✓		
[103]	Pseudonym method	✓					✓	✓					✓
					State of Body and Mind								
[218]	Additive homomorphic		✓				✓		✓		✓		
[219]	SHA-256			✓	✓			✓			✓		
[220]	Compressed sensing	✓					✓	✓				✓	
					Media								
[221]	Quasi-fully		✓		✓				✓		✓		
[222]	Homomorphic Random Linear Network Coding (RLNC)		✓			✓		✓			✓		
[98]	Rivest–Shamir–Adleman Algorithm		✓				✓			✓		✓	
					Social Interaction								
[99]	Paillier homomorphic encryption, Diffie hellman key exchange			✓		✓				✓			✓
[211]	Image Processing	✓			✓								
[210]	AES-128				✓		✓			✓			✓
					Data Aggregation								
[39]	Elliptic curve		✓			✓		✓				✓	
[37]	Homomorphism		✓				✓	✓				✓	
[38]	Pseudorandom function, sink-rooted spanning tree		✓		✓			✓				✓	
					Data Query								
[213]	Attribute-based Encryption (ABE)			✓	✓					✓			✓
[126]	Homomorphic Encryption Noise Perturbation			✓			✓			✓			✓
[212]	Differential Privacy			✓			✓			✓			✓

In a smart city, a one-way hashing solution to preserve the privacy of RFID readers for anonymous authentication has been proposed [117]. The solution ensures that the data of individuals, which leads to habitual patterns, are kept private. The anonymous authentication solution is efficient in terms of performance, including computation, communication, and memory. The method used in the solution is that one-way hashing is a lightweight operation, which only generates authentication requests and IDs while reducing

unnecessary modular and hash operations. Lightweight one-way operations may compromise data privacy, e.g., an attacker can launch a birthday attack for the hash collisions and gain habitual patterns of an individual.

Instead of a one-way basic hashing solution, a lightweight AES and SHA1 based scheme has been utilized to design a framework for achieving content privacy in smart homes. Sensor devices and home gateway perform AES encryption operations to preserve the identity of home appliances and the sensor device's presence [114]. Due to the AES operations applied on sensor devices, the computation and memory overhead is high. A centralized service provider is also utilized to generate AES encryption keys, authentication keys, and data storage. An attacker can compromise the communication link between smart home devices and a service provider. Also, the involvement of a centralized unit makes the privacy level of the scheme medium. The scheme is also vulnerable to a single point of failure threat.

A decentralized privacy-preserving scheme to overcome a single point of failure threat of centralized processing and storage has been proposed [116]. In the scheme, Diffie Hellman key exchange and hashing instead of AES is used to achieve decentralized individuals' behavior and actions privacy in smart homes. For preserving privacy, sensor devices create their data blockchains and communicate with other sensor devices, smart home miners, and local storage to handle data chain transactions. Miners are responsible for transaction handling, and miners are interconnected to smart devices, local and cloud storage in a distributive manner. The blockchain transactions are encrypted and processed in a shared manner. Thus, miners reduce the computation and memory burden on a single smart device by sharing the workload. Although the concept of blockchain can achieve a high level of privacy with low computational and memory overhead in smart homes. Nevertheless, the distribution of transactions increases communication overhead. Also, Diffie Hellman's key exchange protocol gives weak security guarantees considering basic hashing for a distributed system in [116].

For preserving behavior and action privacy in a healthcare system, a multi-agent architecture concept with a pseudonyms method has been proposed [103]. The method provides authority to patients to select their data privacy level so that only selected private information is transmitted to medical servers [103]. Despite giving the authority to patients, the method cannot be fully utilized since patient data need to be transmitted to a central medical server for long-term storage. The central server increases DDOS attack vulnerability and leads to data loss caused by a single point of failure. The centralized authority has been considered for users' attribute's signature generation, leading to a single point of failure threat. A decentralized or distributed platform should be required to overcome such

limitations. It also incurs high communication overhead due to patient data monitoring, policy creation, data collection, and transmission to cloud/local storage based on patients' provided level of data privacy. Computational and memory overhead also increases at policy agent nodes due to the generation of rule-based XACML (eXtensible Access Control Markup Language) to define a patient's preferred policies.

In IoT applications, behavior and actions' data is replicated to various locations in the Cloud environment for data reliability, survivability and backup [48, 49].The replication of sensitive data network is vulnerable to many attacks, for example, an adversary can modify or delete replicas to make data unavailable to end-users. Therefore, preserving replicated data privacy at various locations is essential for maintaining the data reliability, authentication, and survivability.

Mansouri & Sharma et al. highlighted data protection concerns in a cloud environment [48, 49]. Sharma et al. proposed a scheme for data protection using data division into small fragments that can then be replicated to different locations in cloud using a fragment placement algorithm [49]. The scheme did not rely on cryptographic measures to encrypt data. Sharma et al. claimed that the non-cryptographic nature of the scheme makes it faster to perform replica placement operations in the cloud.

Similarly, Jayasaree and Saravanan considered a data security scheme [48] for data replicas. In the scheme, a particle swarm division algorithm has been adopted for optimizing the placement of the replicas in cloud computing. The scheme divides replicas into fragments and then distributes and stores them using the T-colouring concept [48]. Since the data fragments are not encoded, and they are distributed to different locations in the cloud, an adversary exposing few fragments would be able to analyze and discover data patterns and their meaning. Also, the privacy and reliability of the data are not guaranteed. The schemes [48, 49] are also vulnerable to DoS and authentication attacks.

- **State of Body and Mind:** Solutions for preserving the individual's state of body and mind privacy include data perturbation and distribution with Shamir's secret sharing techniques, additive homomorphic encryption, Arnold's scrambling, and logistic scrambling, and distributive role-based access controls [218-220, 223]. A data perturbation solution based on the AES algorithm has been adopted for preserving data privacy at sensors. The data is distributed among sensors using data fragmentation, and cloud aggregates fragments from single/multiple sensors [218]. The cloud is utilized to store the aggregated fragments of data, which incurs low memory overhead for computation and storage. However, aggregating fragments from multiple sensors incur high data transmission and communication overhead. The sensors are also vulnerable to jamming and DoS attacks due to the limited resources of sensors available for computation.

A solution [219] to improve privacy preservation and reduce communication overhead due to distributive computation has been proposed. In this solution, physiological parameters of the state of the body are preserved using compressive sensing with Arnold's and logistic scrambling. On the one hand, the solution preserves the privacy of physiological parameters by compressing the semi-tensor image parameters and then encrypting the parameters at the sender node, i.e. sensor. At the receiver node, the receiver deciphers the physiological image using scrambling, decompressing and, hash key.

On the other hand, computational and memory overhead is low due to the smaller measurement matrix for semi tensor images. Communication overhead is less because of only two entities, i.e. sender and receiver communication required for data transmission and decryption [219]. The limitation of the solution is that the original data can be altered without the receiver knowing the actual data in case of the sender node being compromised. A data compression solution based on data compression at the sensor node has been discussed in [220]. A distributive computation like [218] has also been considered in the solution. The proposed solution provided role-based access control to distributive sensors in a cluster. A cluster head is responsible for compressing the sensor's data, and then the base station aggregates all data from sensors and decompresses it for long-term storage [220].

Although the scheme reduces computational and memory overhead at cluster heads and base-station, compressed data is transmitted in a plain format making data privacy low and vulnerable to eavesdropping. For example, in a smart city, individuals using a running application may compromise their health data to eavesdropper by measuring the count of runs. Therefore, the count of runs can be made secured using encryption techniques, which stopped tracking and counting individual runs [223]. Also, medical regulations' in Health Insurance Portability and Accountability (HIPAA) have set up requirements for privacy policies that can be extended and implemented by service/ software vendors according to the need of data owners.

- **Media:** In the past, crowdsourcing, data minimization, obfuscation, media compression, and anonymization solutions have been proposed to keep media content as private as possible [98, 179, 180, 221, 222]. A crowdsourcing and data minimization scheme has been proposed in [221]. The scheme is based on the isolation of sensors from other systems to prevent the combination and correlation of personally identifiable data. Isolation partly allows systems to acquire sensor data, which is meant for that system to use further or store it. For example, in the case of identity verification, the entire video face data is not required to be stored in a system. Only vectors of face features are extracted, quantized, encrypted, and used for collateralizing and comparison with the previously stored feature vectors [221]. Facial

vectors are encrypted using a quasi-fully homomorphic method, which preserves the privacy of facial vectors. The performance efficiency of the scheme is high in terms of communication and memory, as only extended face vectors are outsourced to a database for verification and storage. The drawback of the scheme is that the computational overhead medium to low due to heavyweight polynomial computation of quasi-fully homomorphic method.

An obfuscation and anonymization method has also been introduced for facial vectors privacy, for example, in multiparty video caching via Content-Centric Networks (CCN), the facial identity of an individual request for accessing video content is kept private [222]. The method aims to obfuscate individual requests by dispersing the requests across several networks with distinct paths.

A similar obfuscation method for a secure media-based surveillance system has also been introduced for smart city platforms [179, 180]. Obfuscation is introduced using packet routing and video encoding to preserve the privacy of media data with high computational and communication performance efficiency [179, 180]. The obfuscation and anonymization methods proposed for media privacy can protect an individual's privacy, obfuscated or anonymized data is still stored on the central unit/cloud. The centralized storage system makes these methods vulnerable to a single point of failure threat leading to a DDoS attack, which increases the network bandwidth, communication overhead, and data recovery issues.

A distributive cooperated framework to mitigate the centralization issue of media privacy has been introduced for smart objects in a smart space [98]. Privacy is deployed using distributive cooperation between systems. The systems use a heavyweight Rivest-Shamir-Adleman (RSA) algorithm to implement privacy, which is not well suited for resource-constrained IoT devices. The distributive cooperated framework's real-world experiments, implementation, and comparative analysis are not provided [98].

- **Social Interaction:** Solutions have been proposed for protecting digitally capture images and live-streamed videos in social gatherings and lifelogging [99, 210, 211, 224]. For social interactions, the private details of a live video are preserved using a clip art image [211]. An automatic algorithm transforms the video into a clip art image using image processing. For transforming a video, the algorithm abstracts the visual details of a video and then detects a certain object. After detection, an algorithm replaces the object with a clip art image. The Algorithm preserves the private object of a video from the viewers. The algorithm is only able to preserve the privacy of a single object at a time.

  Further, the aesthetic quality of a clip art image as a replacement is low. A solution based on a privacy mediator has been introduced to preserve the privacy of multi-object in live

video analytics [210]. In the privacy mediator virtual machine at cloudlet, objects of a video stream are decoded and denatured according to the privacy policies. After denaturing, including modification of object with clip art image, the obscured bits of video are encrypted using AES-128 cryptosystem to preserve the privacy of video analytics. Due to the public-key cryptosystem and cloudlet centralized computation and storage, the network overhead increases, which incurs low performance efficiency of privacy mediator at the cloudlet.

A distributed secure computation ensures that individuals' privacy choices and visual features are kept protected publicly for social gatherings [99]. Image captured in a social gathering is encrypted using a pallier cryptosystem at a photo capturing sensor device. The capture agent and bystander agent are involved in enforcing privacy policies on sensor devices securely. Computation of Paillier cryptosystem and enforcement of privacy policies require efficient capture agent devices to perform computation with high-performance efficiency. However, due to the resource-constrained nature of IoT sensors, the efficiency and accuracy of privacy enforcement are not fully achieved in multiparty computation distribution.

- **Data Aggregation:** Symmetric key and asymmetric key homomorphism methods have been used for performing operations over encrypted data and then aggregating data securely in sensor nodes [37-40]. Sensors encrypt data using a symmetric cryptosystem, forwards encrypted data to an aggregator that encodes it, and then sends the resultant data file to a base station for decoding/decryption and long-term storage [37, 38].

  Although the homomorphic mechanism used by symmetric key optimizes the performance overhead with fast operations compared to the public key method, encrypting and decrypting data with the same key may result in a key-compromise attack. To overcome such an issue, a technique for data division and distribution to different aggregators has been proposed [39]. The elliptic curve cryptography method for encrypting data has been utilized, which provides the same level of performance efficiency as symmetric cryptosystems. However, aggregated data is forwarded to the base station, which causes network overhead and an increasing number of data collisions. The memory overhead of a centralized base station also increases to store aggregated data.

  Methods to mitigate data collision and centralize data processing were proposed with multi-party computation. For example, in a smart grid system, data privacy and reliability of smart meter readings have been tackled using homomorphism [40]. Although the scheme provides data privacy with scalability and performance efficiency in multi-party computation, it does not deal with users' privacy with lightweight cryptographic methods, limiting the use of the scheme in memory constraint IoT.

- **Data Query:** With data aggregation facilities, preserving the privacy of fine-grained search queries cannot be negligible. A few solutions have been discussed to preserve the privacy of fine-grained query data in IoT applications [126, 212, 213]. A protocol based on garbled circuits, partial homomorphic encryption, and secret sharing methods has been adopted to preserve the privacy of multi-user queries in smart homes [126]. In the protocol, the multi-users send a query to two cloud servers for access to certain genomic data of data-owner stored in cloud servers. The query generated by multi-users is encrypted using homomorphic encryption and is forward to two non-colluding servers. The servers communicate with the data-owner regarding the query and, on data-owner's approval, process the query with the secret sharing method. Both servers corporate with each other to answer a query securely. The server's cooperation and heavyweight cryptosystem, including homomorphic encryption and secret sharing in a protocol incurs high-performance overhead with network bottleneck.

  A protocol based on differential privacy technique preserves the privacy of data query in the IoT application domain of smart city [212]. A perturbation noise using a machine learning technique is added in a query to guarantee differential privacy. In machine learning, distributed training and testing is carried out for generating query responses. The distributed computing results in high communication overhead like cooperative computing in [126]. Another protocol for preserving the data query in smart city applications has been proposed [213]. The protocol is based on attribute-based encryption (ABE) has been used, which provides authority to data owners for attributes policy setting. Data-owners define the policy for preserving the privacy of data queries using ABE, which increases the privacy of data-owners identity and content of data query. However, the burden on resource constraint IoT devices to perform ABE complex operations, creates a performance bottleneck, delay in query responses, and vulnerability to a DDOS attack. Due to these limitations, computational and memory overhead on the network increases.

3.1.2. Context Privacy

In this section, we discuss the possible solutions proposed for the context privacy of IoT applications. Table 3 presents an overview of the context privacy requirements with features including privacy techniques, privacy, and performance levels (low, medium, and high), which schemes are providing for preserving privacy.

Table 3 Techniques for Context Privacy Preservation in IoT

		Privacy Level			Communication overhead			Computational overhead			Memory overhead		
Reference	**Privacy Technique**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**
					Identity								
[117]	Forward secrecy, hash		✓		✓			✓			✓		
[214]	Elliptic curve			✓		✓		✓			✓		
[136]	Elliptic curve			✓		✓			✓			✓	
[225]	Hash function	✓			✓				✓			✓	
[215]	Bitwise operation	✓			✓			✓			✓		
[226]	Elliptic curve, granulation computing			✓			✓	✓			✓		
					Temporal								
[100]	Perturbation Laplacian		✓			✓		✓			✓		
[107]	Priority queue	✓			✓			✓			✓		
					Location								
[92]	Fake packet injection	✓			✓			✓			✓		
[227]	MAC	✓			✓					✓	✓		
[84]	Fake bits injection	✓			✓			✓			✓		
[228]	Hash	✓					✓	✓			✓		
					Policy and Access Control								
[204]	Rivest Shamir Adleman (RSA) encryption			✓	✓					✓			✓
[229]	Attribute-based digital signatures			✓	✓					✓			✓

- **Identity Privacy**: Anonymous authentication methods have been proposed to preserve the identity of sensor devices, data users, and data owners [117, 214, 215, 226]. An authentication method based on a non-collision hashing function has been applied to sensor nodes for preserving the identity of nodes [117]. Gateway node uses a one-way hash function to hash the identity. It verifies sensor nodes and users' legitimacy using HMAC, thus providing a remedy against DoS or de-synchronization attacks. The method could be incorporated with the existing anonymous authentication protocols to significantly reduce computational and communication overhead and increase the identity privacy of sensor nodes. However, data of sensor nodes is stored on a base-station with no data recovery mechanism, which makes it vulnerable to a single point of failure attack.

  A method [215] based on bit operation has also been put forward for anonymous authentication of IoT devices such as RFID tags. The bitwise operation uses only two operations, i.e. XOR and left rotation to secure the identity of RFID tags. These operations are ultra-lightweight, which incurs low-performance overhead on limited resources RFID tags. However, XOR operation does not provide a high level of privacy of RFID tags as XOR is vulnerable to known-plaintext attack and identity traceability. A method based on Elliptic Curve Cryptography (ECC) to overcome the known-plaintext attack and improve the un-traceability of user's identity in RFID tags has been proposed [214]. In this method, a gateway node acts as a cluster head to verify the one-time alias identity of RFID tags and users

participating in the network. After successful verification, users' identity encoded in smart cards is encrypted to enhance users' anonymity and unlinkability.

The method is also able to provide the forward and backward identity secrecy with elliptic curve multiplication and addition. Further, the lightweight ECC encryption method can reduce the computational and memory cost of limited resource IoT devices. Although the lightweight method [214] provides anonymity using only symmetric key hashing, which is considered a lightweight solution. However, user identity is at stake if a symmetric key gets compromised. Further, a backend server may monitor the communications among RFID tags, which may lead to a single point of failure attack. The method is also vulnerable to DoS and collusion attacks leading to a loss of anonymity property. To overcome a single point of failure, DoS and collusion attacks, a distributive key management method is proposed in [136]. Similar to the [214] method, the ECC encryption method is used in [136]. The method effectively employs with anonymity and privacy of customers using distributive key management and authentication security practices. The use of high-level protocol and multiplication and addition of elliptic curve point for authentication and distributive key management makes the method performance, including computational, memory, and communication middleweight thus unsuitable for resource constrained IoT devices.

A smart cart solution in the shopping system mitigates performance overhead issues, including computational and memory [226]. The solution is based on the anonymity model using an elliptic curve and granulation computing for RFID data. Also, individuals can set their data privacy preferences for data usability with minimum computational and memory overhead. The authors also claim the solution as the first anonymity model considering the anonymity of quasi-identifier attributes of a smart cart. However, the solution does not reduce the communication overhead due to interaction between RFID reader, smart cart, server, and checkout points of sale. Also, the solution does not deal with any other security property, including data integrity and confidentiality.

For authentication of RFID anonymously, another lightweight authentication solution has been proposed [225]. The solution preserves identity and location privacy, forward secrecy, data availability, and high scalability. The solution is based on a hash function for anonymous RFID authentication in a distributive IoT environment. A backend unexhaustive query server performs a search to identify the RFID tags anonymously quickly. The scheme can compute position information for authentication only if data packets have been sent through legal RFID tags. Therefore, compromise of position computation message may result in tag privacy, forward and backward traceability problems. A fully secure design for wireless sensors network is needed to mitigate a single point of failure, performance overhead, third-party involvement, and backward traceability problems.

- **Temporal Privacy:** Researchers propose few solutions to deal with sensor nodes' temporal privacy, the trivial one is to use a time-driven model [124, 191]. In this model, the packet time-to-live is defined. If an interval is set short, it reduces the lifetime of a packet; on the other hand, if it is too long, then real-time processing and holding packet capabilities of the network are affected. The limitations affiliated with time-driven models have been highlighted with solutions to tackle them in wireless sensor networks [107, 195, 225]. The time-driven solution [195, 225] is based on random packet delays during packet transmission to the central station. The utilization of random data overloads buffer size at the intermediate nodes between the source and sink node, which delays the packet on-time delivery [195, 225].

  A temporal perturbation solution based on the concept of Laplacian in a real-time monitoring system to mitigate the use of random delays for preserving temporal privacy has been discussed in [100]. The Laplacian distribution enables a receiving node to aggregate data from multiple smart meter sources, due to which an attacker cannot infer sensor nodes' correct timestamp. The solution can preserve the privacy of timestamps with a minimum performance overhead. However, an attacker can still traceback to an original sending time of the sensor's sent data, which may result in monitoring data patterns [100]. A priority queue-based solution has been proposed for temporal privacy preservation to overcome Laplacian, random delays, and time-driven model limitations [107]. Healthcare data generated by sensor nodes such as EEG, ECG, blood pressure and heartbeat, etc., are priorities in the sink node and the highest priority data is sent to the server to guarantee on-time data delivery [107]. With on-time delivery, data forward is not kept secure during transmission, which may lead to a man-in-the-middle attack.

  From the state-of-the-art research in privacy-based IoT applications, we find that most attackers mainly focus on targeting time-dependent transitions to capture the data transmission patterns, which may lead to identity and content exposure. Therefore, researchers should consider a diversity of temporal-based vulnerabilities for preserving content and context privacy.

- **Location Privacy:** There have been few efforts made to hide the location of a sensor, for example, a concept of randomizing routing paths for the source and destination [195, 230] to prevent location-traceback attacks. A concept of injecting bogus traffic bits to misled an attacker [84, 92] from identifying source node location has been provided. Similarly, a source node encloses locations in an innocuous message to hide data transmissions between source and destination nodes [227]. The source node encodes location into a beacon frame and then applies MAC layer encryption mode on that location and data. These schemes [84, 92, 195, 227, 230] misled an attacker from location identification and provided low-performance

overhead. Due to an increase in network traffic, the data collision and DDoS attack at the destination node may expose source location. Therefore, privacy preservation of location of source node is low, which makes these schemes not suitable for preserving the location privacy of IoT devices.

Another solution based on authentication of vehicle locations in VANETs has been proposed [228]. The solution mitigated man-in-the-middle, and data compromise attacks for VANETs. An authentication method, including a cooperative message of sensor nodes with unlinkability to a sensor location, has been introduced to tackle these attacks. Third-party authority is involved in providing authentication using an evidence-token approach to all sensor nodes in the VANET network. After token creation, each sensor node communicates to other sensor nodes with a hash of a token. This communication is considered as a cooperative message directly between sensor nodes and not directly through the third-party authority. The third-party authority is only responsible for generating a token, not carrying out cooperative messaging between sensor nodes. Token-based authentication of vehicles through third-party authority increases the communication overhead of VANETs. Hence the solution is exposed to DoS and single point failure attacks.

Most of these aforementioned solutions were designed to protect either data source locations or destination locations or a particular type of attacker (passive). A platform that can simultaneously secure the location privacy of source and destination nodes and all possible active and passive attacks is of paramount importance. Further, as the IoT network density increases, a solution aiming to inject fake network traffic becomes disruptive with frequent packet retransmission and collision.

- **Privacy Policy and Access Control:** Policies and access control-driven interfaces have been designed to preserve users' privacy [101, 204, 229]. For example, a heavyweight method for policy creation has been proposed [204]. The method introduces adaptability and user transparency in assisting living healthcare for configuring privacy requirements. Policy rules in this method are based on behavior and matching relations based on the activity patterns of users. RSA and AES methods are proposed to focus on privacy-preserving of users' behavior and activity patterns with policy standards. The use of RSA makes the method heavyweight, and it is considered to be costly, not suitable for resource-constrained IoT sensor devices.

  Another solution is based on ciphertext-policy attribute-based encryption [229]. The solution defines multiple policies with users' attributes for encryption and then outsources data to cloud storage. The solution has also been able to mitigate collision attacks by defining multiple policies. Although the solution provides a high level of privacy using attribute-based encryption, the use of attribute-based signatures and defining policies makes it

heavyweight. Due to the heavyweight policy method, the solution incurs a high computational overhead like the RSA method proposed [229]. Therefore, the solution is not able to support the scaling needs of IoT devices.

Similarly, a heavyweight policy solution based on semantic ontologies has been introduced [101]. A concept of semantic ontologies has been introduced for policy behavioral modelling, policy decision-based language, policy rule evaluation, and enforcement rule monitoring. Likewise, the attribute-based encryption method, designing ontologies for policies requires much energy, which cannot be done in resources-constrained IoT devices.

3.2. Review of Privacy Preservation in Fog-enabled IoT

Based on our privacy classification, here we discuss the contributions of existing solutions in Fog-enabled IoT according to the application domains of IoT. Sections 3.2.1 discusses content privacy models, and Section 3.2.2 discusses context privacy models. The application domains in which fog-enabled IoT solutions are provided are illustrated in Figure 8.

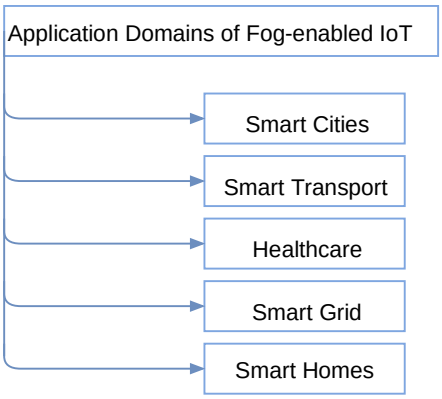


Figure 8 Fog-enabled IoT Applications

3.2.1. Content Privacy

Here we provide the analysis of existing solutions adopted to preserve content privacy requirements in fog-enabled IoT platforms. Existing solutions are organized according to the application domains of IoT. Figure 9 gives an overview of content privacy models used in fog-enabled IoT, which are discussed in detail in subsequent sections.

- **Behavior and Action:** Recently, solutions [41, 67, 120, 133, 231] to preserve privacy in a manner that keeps individuals' behaviors or actions private and secure have been proposed. Mainly, the proposed solutions have attempted to mitigate traditional IoT-based schemes problems w.r.t high computational, storage cost, and transmission delays to cloud. For smart city applications, system models using user-level key management, differential privacy, and blockchain has been proposed in fog-enabled IoT [41, 67, 120, 133]. A differential privacy-

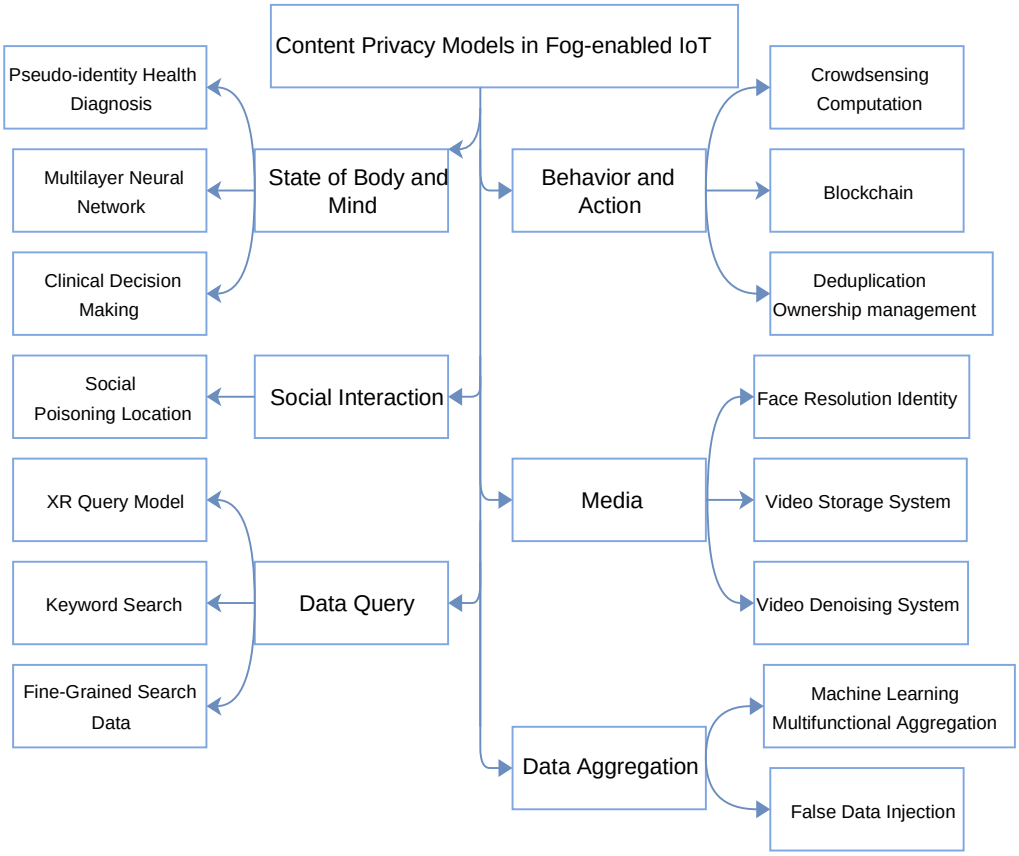


Figure 9 Content Privacy Models in Fog-enabled IoT

based query model preserves the privacy of user's actions stored in fog computing supported data center [120]. Information of user's actions includes datasets obtained from vehicle activities performed by users. In this scheme, the query model extracts the information structure from the fog computing supported data centers. Then the extracted information is preserved using the Laplacian mechanism in differential privacy. The scheme efficiently extracts datasets and executes query models with privacy preservation of datasets. Due to the Laplacian mechanism applied on a query model, the scheme resists fog node and network edge recognition attacks.

Another scheme for a smart city system has been proposed to provide distributive data storage among fog nodes using the blockchain technique [41]. The scheme is adopted to minimize the centralized storage limitations of IoT applications. In the scheme, each user manages its security keys, and data is encrypted using security keys. The encrypted data is distributed among fog nodes, and each fog node stores only an encrypted fragment of data. Thus, the scheme achieves complete data privacy without any third-party involvement for security keys generation and control of the data.

An efficient and secure deduplication scheme [133] to provide data ownership in fog storage has been proposed. In the scheme Merkle (hash) tree mechanism has been adopted

for data deduplication. Merkle tree authenticates proof-of-ownership (PoW) of data, and then according to the security keys of data-owners, deduplication is applied on encrypted data. Data owners are provided access control for defining user-level key management to deduplicate and manage data. The privacy of individual actions using deduplication without the trusted third-party requirement has alleviated the computational and memory burden from the cloud and minimized the communication cost between IoT sensors and cloud. Also, forward secrecy is maintained in the scheme by prohibiting data owners from accessing their previously outsourced data content to fog nodes.

Another scheme based on secure access control in fog-enabled IoT has been proposed [67]. The encryption of data using an attribute signature is done at fog nodes. The data owner can define data policies and perform encryption, and end-users can decrypt data with editing and re-encryption properties. The scheme provides access control to data owners and considers access control of data users to amend data for further processing. The scheme is secure against known attacks. Like scheme [133], the proposed scheme incurs less computational and communication overhead as it alleviates the burden from cloud to fog nodes. Further, the scheme does not consider the third party for key generations.

The behavior privacy of IoT devices in the smart city has been preserved using ontologies [232]. The ontology model preserves the privacy of IoT devices by changing the privacy behavior of data devices in smart cities dynamically. First, the privacy rules are defined in the ontology model. Then the server defines and applies privacy rules for each IoT device. Applied privacy rules on data are forward to the ontology server for further refinement. Cloud storage is utilized for the long-term storage of processed data from the edge of the network.

The users' behavior using fog computing in smart transport systems has been considered in [231]. A road surface monitoring based on crowdsensing behavior of vehicles. The vehicles are monitored using fog computing as fog provides services close to roadside units (RSU) of vehicles. The scheme has also been useful to detect anomaly behavior of a vehicle by keeping the detected data private and the vehicle’s identity anonymous. The scheme is also based on an efficient data aggregation method, which is a certificateless signcryption method. This method is used to aggregate vehicle data securely at fog nodes with minimum communication and computational overhead. Aggregated data is further forward to cloud for long-term storage.

- **State of Body and Mind:** The schemes to preserve the privacy of state of body and mind for healthcare systems have been proposed in fog-enabled IoT [129, 140, 233]. For example, in the clinical decision-making system, the privacy issues of personal information leakage while monitoring a patient's health status [129]. Furthermore, clinical decision-making systems

have been considered as the first step towards integrating health and genome data. According to cloud privacy policies and standards in improving patients' trust and reliability in Healthcare, a decision-making system is designed in fog computing. The proposed system preserves healthcare privacy, adhering to standards as defined for cloud computing to fully achieve the trustworthiness of fog clusters.

Another scheme [233] to preserve the privacy of health diagnoses in the healthcare system has been provided. The health diagnosis is encrypted at the edge of the IoT network using homomorphic encryption and two trapdoor cryptosystems. The mobile user submits an encrypted request for a result of health diagnosis to the edge of a network. After checking the authenticity of a mobile user, the result of the corresponding diagnosis is sent to a mobile user. The scheme is effective in terms of providing accurate diagnosis results to mobile users. Computation at the edge of a network reduces the network bottleneck of the healthcare system.

An E-healthcare framework to record a patient's state of body privacy in the healthcare system has also been presented [140]. For preserving the privacy of patient data, ECC and pseudo-identity have been adopted. The ECC ensures the undetectability of patients' health concerns, and pseudo-identity kept the identity of the message showing health conditions private. Not only medical records are kept private, but also reliability of the data requester to view health data is ensured.

- **Media:** For an application domain of a smart city, face identification privacy and video storage privacy have been proposed [109, 143]. Face identification and resolution include the facial features of an individual for obtaining identity information [109]. In the scheme, ECC, Diffie Hellman (DH) key exchange, and AES has been applied to preserve the privacy of facial features. DH is adopted to generate a session key for a session between fog nodes and the management server. To keep the session information secure, the ECC mechanism is used at fog nodes. After the generation of session keys, the face identification is encrypted using the AES cryptosystem. The encrypted face identification is verified by the management server using the SHA-1 algorithm. By applying a strong cryptosystem, the scheme can provide resilience against forgery and man-in-the-middle attacks. The scheme provides resilience while having less performance overhead on data transmission when compared to cloud-based systems [26].

  Another scheme for an application domain of smart city has been proposed in a fog layer to enhance video surveillance with high-performance efficiency [143]. In the scheme, video denoising has been provided in which the noise added to the data generated at sensors is removed by applying noise filters and signal-to-noise ratios. Video denoising is performed in a distributive manner at fog nodes. Thus, reducing the performance overhead on a single

fog node or resource-constrained sensors. fog nodes consist of embedded systems, which provide the required processes of denoising video generated from sensors, encryption of denoised video, and then compression of denoised video. After these processes, encrypted video is sent to cloud for decryption, decompression, and post-denoising. These schemes provide an opportunity for researchers to explore fog computing for media privacy at the edge of the sensors transmitting media data.

- **Social Interaction:** In the application domain of smart homes, a scheme to preserve the location of the user's social interaction from a poisoning attack has been introduced [234, 235]. At the edge of the IoT network, the scheme first utilizes the learning model to infer the social relationship of a user. Then constructs a social graph, which helps to identify the poisoning location and secure the social network from that location. The scheme efficiently secures the social network of users at the edge of the network without the need for a remote cloud for computation. The privacy preservation of social interaction for application domains of IoT in fog computing is at an infant stage. To strengthen content privacy including the mental health of data-owners, it is important to consider the privacy of users' social networks and relationships.
- **Data Aggregation:** Preserving the identity of individuals during data aggregation for different IoT application domains has also been considered in fog computing [9, 13, 16, 41, 42]. For example, in a smart city design, a fog node aggregates data from smart meters and forwards the aggregated data to the cloud for long-term storage [9]. The solution is based on Castagnos-Laguillaumie, short-signature, and bilinear pairing cryptosystem that can provide secure aggregation. Using data aggregation, the proposed solution saves the bandwidth overhead between fog nodes and cloud server as an only aggregated reading of smart meters is forward to cloud for storage. The solution also provides anonymous authentication of sensor nodes, fog nodes, and cloud using pseudonyms.

  To minimize the storage limitations of smart city applications, distributive storage of data among fog nodes using blockchain technology has been proposed [41]. In the scheme, data generated by IoT devices are aggregated at multi-interfaced base stations at the edge of the IoT network. The base stations act as a forwarding controller of aggregated data to fog nodes. The aggregated data is encrypted and distributed among fog nodes. Each fog node stores only an encrypted fragment of data. The authors claim that their technique is better in reducing response time delays, increasing throughput, and detecting real-time attacks as compared to existing models. One of the aims of using blockchain technology in their model is to provide complete data privacy in fog computing.

  For minimizing the storage and computational limitations of smart city applications, a divide-and-conquer scheme has been proposed for data aggregation in fog computing [42].

The scheme preserves the privacy of data using encryption, division, and distribution of data. The encryption is performed using a lightweight AES cryptosystem. Encrypted data is divided into blocks and distributed to fog nodes in different fog clusters. On end-user device requests to access aggregated sensors' data, fog nodes act as an aggregator and aggregate the data blocks in the proximity of their cluster. The blocks are aggregated using the blockchain concept. The aggregators then forward aggregated data to the end-user device. The end-user device performs the final aggregation of data coming from all aggregators and decrypts the data. The scheme can reduce computational and memory costs using distributive computation.

A machine learning approach has also been introduced to achieve data privacy during data aggregation in a smart city [13]. The approach provides additive and non-additive aggregation on sensor data, and the aggregated results from fog nodes are sent to the cloud. In the machine learning technique, the learning model is trained, which predicts the results of aggregation query and supports aggregation functions, including additive and non-additive. Authors claimed that using their machine learning technique at fog nodes minimizes performance overhead compared to cryptographic heavyweight technique, such as Paillier homomorphic encryption.

In the application domain of smart grid, readings of smart meters are aggregated using the chinese remainder theorem [16]. The theorem provides false data injection resistance with efficient data aggregation supporting fault tolerance. The proposed solution can mitigate the heterogeneity and hybrid data type limitations of data aggregation solutions' [236-238] by combining heterogeneity and hybrid data into one ciphertext. The solution also preserves the privacy of aggregated data at fog nodes using Paillier encryption. The aggregated data is secure against external attacks as Paillier encryption is IND-CPA (indistinguishable under the chosen plaintext attack). Also, the authenticity of smart meters and fog nodes is provided using one-way hash chains.

- **Data Query:** For application domains of IoT, fog nodes can be utilized to provide a solution for owner-enforced keyword searches [120, 134, 144, 239]. In a smart city application, keyword searching is done by a data user who wants to access and check a data file kept at the fog node. The data user interacts with the fog node by sending a query for a keyword search in a dataset. Only authorized users with satisfying the access policy of the data file can obtain a matching query result. The query result is encrypted at the fog node using ElGamal-ciphertext and sent to the data user. To obtain data plaintext, the user performs decryption using exponentiation and multiplication operations on the ciphertext. Due to query encryption and authorization of data users, the solution can resist swapping and chosen keyword attacks. The solution also provides trapdoor unlikability to data plaintext

and generator of data query.

Another solution provides unlinkability to the data owner and query generator [120]. The solution uses the query model to capture the structural information of fog nodes containing data and provides datasets, which can be mapped to real data vectors. Also, the query model satisfies the differential privacy of queries using Laplacian noise. The solution guarantees query protection and efficient query model computation.

A function query solution in fog computing-based smart grid has been proposed for efficient and secure communication and availability of data aggregation simultaneously [144]. Function queries on data usage can be launched securely by the service provider and data-users. Data is encrypted using a double trapdoor cryptosystem and forwarded to cloud for further billing queries while letting a data owner and users control their data usage. The proposed scheme is preserving the privacy of function queries against probabilistic polynomial-time (PPT) adversaries.

Similar to function queries, the XRQuery solution has been proposed for efficient retrieval of privacy information using query service [134]. This solution is proposed for the application domain of the healthcare system. The query generated by the doctor/ staff is kept private using homomorphic encryption and XOR operations in the XQuery model. The model can resist an external attack on a data query since the private key is shared between the fog node and IoT device in encrypted form. In the XQuery model, the query is encrypted and then shared with fog devices.

3.2.2. Context Privacy

Contextual privacy-based solutions in the fog-enabled IoT that can overcome vulnerabilities of existing IoT-based and cloud-based solutions are discussed in this Section. The discussion is grouped based on the following context privacy requirements: Identity, location, and privacy policies. Figure 10 gives an overview of context privacy models proposed in fog computing for IoT applications.

- **Identity Privacy:** For a smart transport system, the identity privacy of vehicles has been provided in schemes [67, 231]. A scheme based on certificateless sign-encryption technique provides identification privacy of vehicles [231]. The scheme protects the identity using low computational cost as compared to the existing traditional IoT schemes. The scheme can mitigate the limitations of IoT-based vehicular mobile sensors, which collect data and detect anomaly behavior of vehicles [240]. The scheme stores data in a cloud server thus leading to data transmission delays and extensive bandwidth requirements. Using the scheme in fog-enabled IoT network, data transmission delays can be decreased and the need for a

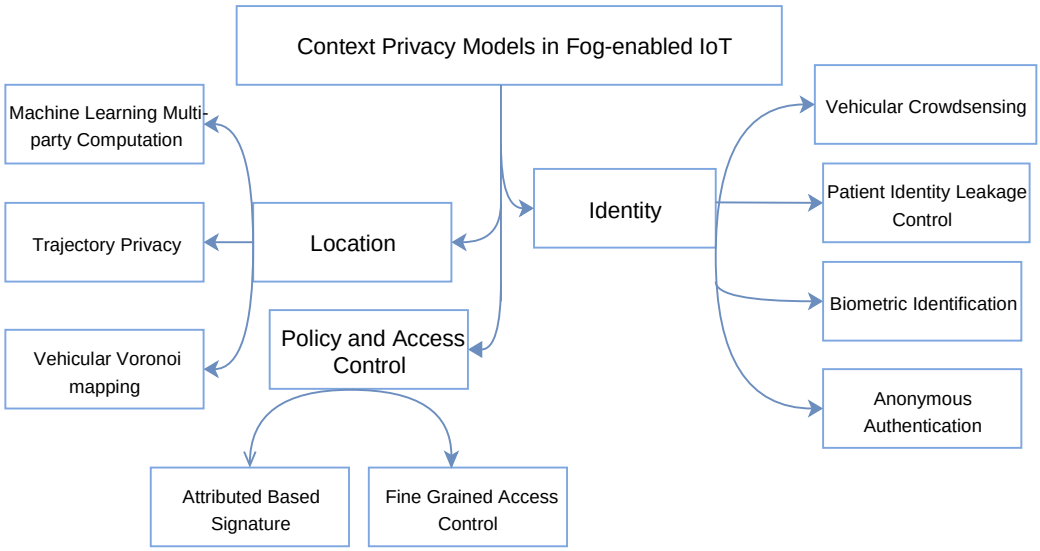


Figure 10 Context Privacy Models in Fog-enabled IoT

cloud or trusted third authority to generate certificates can be removed. Furthermore, vehicular devices' identities can be kept anonymous with resilience to key escrow attacks [231].

Another scheme based on anonymous user authentication to preserve the identity of users has been proposed for smart transport systems [67]. The scheme is based on CP-ABE and ABS techniques for data encryption and identity privacy. Due to these techniques, data users can sign data messages with a claim policy and attributes without revealing his/her identity. The data generated from vehicular mobile sensors are encrypted and outsourced to cloud through a nearby fog node. The authorized user can access and decrypt data if his/her attributes satisfy the access policy defined for CP-ABE. Also, the scheme provides authority to authorized users to edit decrypted data and re-outsource it with his/her ABS. Therefore, the scheme provides access control and privacy policies editing to data users. The scheme also alleviates the computational burden to authenticate signatures and encryption from cloud to fog nodes, nearby mobile sensors.

For the application domain of the healthcare system, the scheme [142] for detecting and protecting the identity leakage of patients, doctors, and staff has been proposed in fog computing. The scheme provides privacy-related API functions to deal with the detection of identity leakage. In monitoring and detecting identity leakage, the proportion of privacy leakage through logs, network transmission, and SMS messages has been considered. Based on the leakage results, the protection mechanism, including API functions is applied to data. The scheme can improve the context privacy for telehealth and telemedicine infrastructures.

Also, the fog computing has been introduced to protect face identification compared to biometric identification in a cloud server [28-30, 109]. Cloud server-based schemes are

vulnerable to identity forgery and man-in-the-middle attacks and response time to identify faces from databases is high. Whereas in a fog-enabled IoT scheme, an AES mechanism is used for the privacy of facial identity, which incurs less computational and communication overhead for response time from databases [109]. The scheme has also applied ECC to encrypt session keys and DH key exchange to generate session keys. By applying a strong cryptosystem, the scheme can provide resilience against forgery and man-in-the-middle attacks.

- **Location privacy:** For preserving the privacy of data source and sink location, few schemes have been carried out in fog computing [13, 67, 101]. For a smart transport system, trajectory privacy in the fog and a location-based server have been proposed [67]. The scheme can preserve the privacy of data source location and provide unlinkability between the source and sink devices and their data owners. The scheme is based on ABS for generating a signature for authentication to keep unlinkability between source and sink devices. Further data is encrypted using ABE, and cloud centralized unit has been considered for only secure data collaboration with ABE encryption.

  Another protocol for location privacy in transport systems has been proposed [121]. A protocol provides location-based proximity detection using the decision tree theory and Paillier encryption to protect the location privacy of individual users. The protocol can achieve the privacy preservation of data sharing among users using Paillier encryption. Decision tree theory is used for protecting the location of data users from service providers. Another protocol [241] for preserving the location privacy of smart transport systems at the edge of the vehicular network has been proposed. A lightweight ECC and homomorphic encryption are used to provide location privacy of vehicles. The vehicles are selected for collaborative downloading of map data from Roadside units (RSU) using a fuzzy logic strategy. Further, the location of the vehicle is kept secure from other vehicles and third-party using pseudo-identity, only trusted authority (TA) can calculate the identity of the vehicle.

  Similarly, a solution for protecting the location of vehicles has been proposed [242]. In this solution, the cloud server divides the map location of vehicles using the Voronoi diagram. Each user of the vehicle sends its region in the Voronoi map to the edge node. Then the edge nodes divide users into a group for the same Voronoi region. Further, users' exact location is embedded with Laplacian noise to preserve the location from edge nodes. The edge node knows the Voronoi grid of each user but unaware of the users' accurate location. Thus, the proposed solution preserves the privacy of the location of vehicle users using the Voronoi diagram.

A location-based on multi-party computation [17] with a public key infrastructure using fog computing has been proposed for a smart city. A machine learning approach is introduced in solution to achieve location privacy using multi-party computation. The learning model is trained for datasets, which predicts the results of datasets to preserve the context privacy of users. A solution also overcomes the centralized storage limitation to preserve the data sender position by guaranteeing the authentication prover location's privacy in a fog network [243].

- **Privacy Policy and Access control:** Fine-grained data access control solution with cloud defined policies have been proposed for fog-enabled IoT [67]. This solution is based on methods including Attribute-based Signature and Ciphertext Attribute-based Encryption. The methods provide data encryption with the data owner's access control and privacy policies. Furthermore, the solution provides a signature authentication in a fog node, thus alleviating the computational overhead from the cloud in computing digital signature at the edge of the IoT sensor devices. Due to the use of the fog node for processing the authentication, policies, and access control, the proposed solution also mitigates the transmission delays caused [244].

  Centralized access control solution reduces computational overhead in encryption and decryption phases, which caused high complexity in a centralized healthcare monitoring system [244]. The policies and access control regulation defined for cloud deployment to the fog-enabled IoT is unlikely to be suitable. For example, a cloud storage purpose is to provide users' access guarantees and privacy setting modification, whereas a fog node aims to authenticate all network nodes [137]. Therefore, different sets of policies and access controls should be applied to serve fog-enabled IoT purposes rather than deployed for cloud computing [244].

3.3. Privacy Preservation of IoT Application in Fog Computing

This section summarizes the main challenges based on our in-depth analysis discussed in the previous section. First of all, we discuss the limitations of IoT-based schemes and then present the mapping between the IoT-based schemes and the fog-enabled IoT schemes. The mapping provides information regarding improvements that the fog-enabled IoT schemes can provide over the IoT-based schemes. The demand for preserving content and context privacy grows sharply due to an individual's awareness of protecting personal information. Several promising solutions have been proposed for smart IoT devices integrated with cloud computing as discussed in previous subsections. However, some limitations restrict the full usability of IoT platforms to enhance end-user services and reliability. We review and discuss key challenges and limitations found in the existing IoT solutions for preserving content privacy and then

elaborate on the countermeasures provided in fog-enabled IoT in sub-section 3.3.1. Similarly, sub-section 3.3.2 discusses the limitations of the existing IoT models for preserving context privacy, and then the sub-section elaborates the countermeasures provided in fog-enabled IoT.

3.3.1. Content Privacy

For preserving the content privacy in application domains of IoT, the limitations of the IoT-based privacy models and countermeasures in fog computing, as indicated in Table 4, are summarized as follows:

- **Centralized and Third-Party Involvement:** A centralized unit (e.g., a server, control center, Cloud or single aggregator) to store or process data, which leads to a single point of failure threats and DOS attacks has been considered in most of the schemes [25, 226, 240, 245-254]. An attacker may gain access to a centralized unit to delete, insert or update its stored or processed data. Also, the attacker can disrupt network traffic making centralized unit's resources unavailable to the network. Data privacy and integrity may be compromised by involving a third party. The responsibilities of the third party in IoT applications are to generate security keys, authentications' agreement, and monitor and managing IoT devices/cloud. An eavesdropper may invade communication links between a third party and cloud/ IoT devices to capture transmitted security keys.

  On contrary, the schemes [129, 133, 231] in fog-enabled IoT overcomes this limitation of IoT-based privacy models. The crowdsensing concept for fog nodes is introduced to mitigate the need of the third party and reduces the transmission delays to cloud. Also, data computation and storage are carried out in a distributed manner at fog nodes, which provide resilience to single point of failure threat, DoS and man-in-the-middle attacks.
- **Computationally Expensive Cryptosystem:** Most of the existing work [9, 25, 56, 103, 204, 213, 236-238, 247, 255] relies on public key infrastructure. The public key infrastructure consists of generally heavyweight methods, for example, RSA, ElGamal, Paillier cryptosystem and homomorphic, attribute-based signature, and encryption. These methods require larger memory capacity and high computation consumption for heavyweight mathematical operations and storage. Thus, heavyweight methods result in communication overhead and degrade the data accuracy.

  Fog computing provides lightweight methods [56, 129, 133, 239] to compute IoT data, for example, ECC, AES, machine learning, bit-wise encryption, hashing and Chinese remainder theorem. These methods overcome the performance overhead issues and provide a high level of data accuracy.
- **Heterogenous Data:** Few of the IoT-based schemes [236-238] are not suitable for computing

heterogeneous data. These schemes considered the homogenous data sources, for example, schemes only consumed electricity readings data from smart meters. In the schemes, the diversity of IoT application domains with heterogeneous data, including blood pressure, heart rate, noise level, humidity, temperature, acceleration, voltage, and sound measures has not been adopted. The aggregation of hybrid data into one ciphertext due to the homogenous nature of data computation has not been provided in schemes [236-238]. Fog-enabled IoT considers the diversity of IoT applications and provides solutions [56, 256] suitable for heterogeneous data sources. These solutions also consider the aggregation of hybrid data into one ciphertext to improve the efficiency of data computation.

Table 4 Mapping of Content Privacy models between IoT and Fog-enabled IoT

Privacy Model	Basic IoT System Model	Limitations in basic IoT model	Fog System model	Countermeasures using Fog computing
Behavior and Action	Preserving data privacy using data duplication [245]	● Do not update outsourced data in an efficient manner ● Trusted authority is required for key update ● Incurs significant-high communication and computational cost	Fog used for Storage mechanism, while preserving the privacy of using deduplication a dynamic ownership management [133]	● Update mechanism and key management authority at the user level ● Fog Storage architecture introduced for alleviating the burden from the Cloud ● Computational and communication cost low
	Mobile sensors collect data and detect anomaly behavior of vehicles [240]	● Data storage on a Cloud server ● Transmission delays to centralized Cloud	Crowdsensing concept using Fog computing [231]	● Anonymity provided ● Less Computational and communication overhead
State of Body and Mind	Trustworthy big data processing with full access control to individual patients to set policies for health data [25]	● Centralized unit involved in storing data ● Data prediction accuracy low due to high computation bottleneck	Fog computing-based multiple-layer neural network for a clinical decision support system [129]	● Patient health status monitoring without personal health information leakage ● High accurate health prediction
Media	Cloud computing-based and cross-enterprise biometric identification system (CloudID) with privacy-preservation. [26]	● Cloud computing bandwidth overhead	Fog computing-based face and resolution identification [109]	● Data transmission overhead less ● Resilient against Man-in-the-middle and forgery attack ● Introduced authentication and session key agreement
Data Aggregation	Differential privacy-preserving data aggregation for smart grid [236-238]	● Not suitable for heterogeneous data, unable to aggregate hybrid data into one ciphertext	Chinese remainder theorem used to aggregate IoT data securely in Fog with lightweight	● Resist against false data injection ● Fault tolerance ●Lightweight privacy-preserving scheme

			differential privacy [56]	● Suitable for heterogeneous data aggregation
	Mobile sensors collect data and detect anomaly behavior of vehicles. Data is stored in the Cloud server [240]	● Transmission delays to centralized Cloud ● Extensive bandwidth required	Crowdsensing concept using Fog computing and encryption aggregation without the need for certificates [231]	● Resilient to key escrow attack ● Less Computational and communication overhead ● No need of a trusted third authority for aggregation certificates
Data Query	Attribute-based encryption for preserving the privacy of fine-grained search data [213]	● The burden on resource constraint IoT devices to perform ABE complex operation, creating a performance bottleneck	Fog-based secure index and keywords generation and transmission. Also, communication between four entities carried out; end-users, data owner, for nodes and Cloud [239]	● Perfect forward and backward secrecy ● Computational efficiency ● Alleviating performance overhead on IoT devices ● Chosen keyword and swapping attack resistance
	Preserve the differential privacy of IoT devices data [212]	● High energy consumption for IoT devices for differential privacy	Four renewable Fog nodes used to efficiently preserve differential privacy with the executable query function [120]	● Preserving the privacy of edge and Fog nodes ● Efficient data utility consumption ● Validity and Reliability of privacy protection ● Quantify the level of privacy protection

3.3.2. Context Privacy

For preserving the context privacy in application domains of IoT, the limitations of the IoT-based privacy models as indicated in Table 5 are similar to content privacy limitations including centralized and third-party involvement and heavyweight cryptosystem. Heavyweight cryptosystem also has an impact on data authorization for policy and access control. Several schemes have been proposed for providing data access control to a data owner [67, 97, 101, 135]. The access control and policy-driven techniques used in these schemes are heavyweight, for example, attribute-based encryption and ontologies incur high computational and memory costs. In [97, 135], multiple authoritative nodes with different resource availability, energy consumption, and geographical location are involved in regulating heavyweight attribute-based access control and data-owner-based policies. Due to the incompatibility of resources and heavyweight access control and policies, the performance overhead of these schemes increases. On the contrary, fog-enabled IoT delegates the computation of authentication and attribute signature to fog nodes in a distributive manner [67]. The distribution of access control and

policies creation at fog nodes optimizes the performance efficiency of computing heavyweight cryptosystems such as CP-ABE.

Table 5 Mapping of Context Privacy Models between IoT and Fog-enhanced IoT

Privacy Model	Basic IoT System Model	Limitations in basic IoT model	Fog System model	Countermeasures using Fog computing
Identity Privacy	Biometric identification with privacy preservation in Cloud servers [28-30]	●Bandwidth problem exists while ensuring privacy ●Response time high from face identification databases ●Vulnerable to identity forgery and man-in-the-middle attack	Introduced Fog computing for resolution applications and face identification privacy [109]	● AES mechanism used for identity privacy ● Resilient against identity forgery and man-in-the-middle attack ● Response time decreased from different face identification databases ● Communication bottleneck decreased
	Identity privacy considering Cloud centralized unit for IoT devices [31]	● The trusted third party introduced as an intermediate between end-users and Location-based servers.	Enhance privacy preservation using Fog computing [67]	● No need for trusted third party ● Important information preserves in the Fog to enhance better management and security
	Cloud-based fine-grained access control framework for healthcare monitoring smart system [244]	●ABE method high computational cost in encryption and decryption phases with the high complexity of policies ● Transmission delays to centralized Cloud	Anonymous user authentication in ciphertext and authentication signature updating delegated to Fog nodes [67]	● Secure against know attacks ● Signature authentication computational burden alleviating from Cloud to Fog nodes
	Cloud Radio Access Networks (C-RAN) and Heterogeneous Cloud Radio Access Networks (H-CRANS) for incorporating radio access networks with Cloud computing to improve energy efficiency and system capacity to store and process data [257, 258]	● Communication bottleneck Vulnerable to DoS and replay attack	Preserving privacy authentication and key agreement in Fog Radio Access Networks (F-RAN) [256]	● Distributed substantial amount of storage ● Adaptive to the dynamic traffic and radio environment with affordable scaling ● Overcomes DoS, man-in-the-middle and reply attacks

Location Privacy	Aggregate vehicle real-time speed and position information using vehicular ad hoc networks [27, 28]	● The malicious vehicle may control traffic lights. ● Pavement loop detectors used which are centrally controlled (Cloud or server). ● Increase in controllers communication resulted in high latency	Vehicular ad hoc networks in a Fog for a secure intelligent traffic light [259]	● Communication latency decreased by only broadcasting and performing lightweight operations ● Resists Denial of service attack ● Detectors not centrally controlled
	Preserve sender positioning privacy [243]	● Cannot guarantee the privacy of prover location, which is highly sensitive	Fog based location position key exchange and Fog nodes (multiparty) computation [260]	● Guarantee the privacy of prover positioning ● No additional computational overhead introduced
	Trajectory privacy considering Cloud centralized unit for IoT devices [31]	● The trusted third party introduced as an intermediate between end-users and Location-based servers.	Enhance privacy preservation using Fog computing [67]	● No need for trusted third party ● Important information preserves in the Fog to enhance better management and security
Privacy Policy and Access Control	Cloud-based fine-grained access control framework for healthcare monitoring smart system [244]	● High computational cost due to complexity of Attribute-Based Encryption (ABE)	Anonymous user authentication signature updating delegated to Fog nodes [67]	● Based on ABS and CP-ABE providing secure data access control in Fog computing

4. Open Issues and Challenges

Fog-enabled IoT opens opportunities to provide content and context privacy both for the IoT and cloud network. Extending existing privacy solutions to fog-enabled IoT remains challenging due to the dynamic nature of fog computing devices, distributive computation, certificate managements, privacy privilege escalation, and insider rogue nodes. This section aims to outline these open issues/ challenges and provides deep into promising future direction for effective privacy protection in fog computing. Table 6 presents an overview of the privacy solutions facing issues and challenges in fog-enabled IoT. Each issue and challenge is extensively explained in subsequent sub-sections.

4.1. Privacy Privilege Escalation

Most of the schemes proposed in fog-enabled IoT suffers from privilege escalation issue [16, 42, 67, 109, 120, 121, 129, 134, 140, 143, 144, 231-233, 239, 241, 242]. In these schemes, privilege

escalation arises when a malicious user exploits a configuration error, bug, or design flaw in a model [261]. This exploitation leads to a challenge, as follows:

Table 6 Issues/Challenges of Privacy solutions in Fog-enabled IoT

References	Issues/ Challenges					
	Privacy Privilege Escalation	Tracking data Accuracy	Usage Pattern Privacy	Rogue Fog node	Certificate Management	Communication Overhead
Behavior and Action						
[120]	✓	✓		✓		
[41]			✓			✓
[133]					✓	
[232]	✓	✓	✓	✓		✓
[231]	✓	✓	✓	✓	✓	✓
State of Body and Mind						
[129]	✓		✓	✓		
[233]	✓	✓	✓	✓		
[140]	✓	✓	✓	✓		
Media						
[109]	✓	✓	✓			
[143]	✓		✓	✓		
Social Interaction						
[234]		✓	✓	✓		
Data Aggregation						
[13]		✓	✓	✓		
[16]	✓	✓	✓	✓		
[42]	✓		✓			✓
Data Query						
[239]	✓		✓	✓		
[120]	✓		✓	✓		
[144]	✓		✓	✓		
[134]	✓		✓	✓		✓
Identity						
[109]	✓	✓	✓			
[142]			✓	✓		
Location						
[121]	✓			✓		
[241]	✓	✓		✓	✓	
[242]	✓	✓		✓		✓
Privacy Policy and Access Control						
[67]	✓	✓		✓	✓	✓

- **Fog Resources Unauthorized Gain**

  The malicious user gains access to preserved resources of fog nodes. The resources of fog nodes are preserved for computation and storage operations, typically the malicious user would be restricted to gain such fog resources. After gaining unauthorized privileges, the malicious user can also access the private data stored at that fog node or request other fog nodes to forward data. The malicious user would also be capable of deploying malware, running commands, and potentially damaging the fog node's cluster.

- **Mitigation for Privilege Escalation**

The proposed schemes do not consider the systematic vulnerabilities in the software design of their models [16, 42, 67, 109, 120, 121, 129, 134, 140, 143, 144, 231-233, 239, 241, 242]. Innovative approaches are required to preserve the resources' privacy in fog-enabled IoT fully. The approaches, including software-defined segmentation and pre-assessment, can mitigate the escalation issues. In software-defined segmentation, the segmenting of fog nodes' resources can decrease the attack surface, and pre-assessment of model vulnerabilities can reduce the design flaws.

4.2. Tracking Data Accuracy

Few of the schemes in fog-enabled IoT have employed the mechanisms of data accuracy in their models to track the reliability of data [41, 120, 133, 134, 143, 144, 239]. The mechanisms include keyword and fine-grained search, and blockchain are summarized as:

- **Keyword Search**

  A keyword and fine-grained search are carried out by the data-owner or end-user devices in fog-enabled IoT models to generate a query for data accuracy proof [120, 134, 144, 239]. The query contains a randomly sampled keyword or hash ID, which is generated from the original data. To track data accuracy, a query sample is matched with a query result provided by fog nodes. If a query result does not match with the query sample, then the data-owner or end-user device reports to the authentic cluster head about the inconsistency of data processed by the fog node. The keyword and fine-grained query mechanisms are vulnerable to jamming attacks. An internal or external adversary can prevent data-owner or end-user from communicating with fog nodes by occupying the communication channel.

- **Blockchain**

  In the blockchain mechanism, proof of work (PoW) is a consensus method, which provides the reliability of data blocks stored at each fog node [41]. Using PoW, end-user device, and data-owner tracks the contribution of each fog node in the processing data block is accurate. The PoW demands enormous power to execute a complex tracking transaction. Thus, most of the fog nodes' resources are allocated to execute the consensus of PoW.

A lightweight consensus method with resilience to jamming and DoS attacks is required to fully track the accuracy of data processing/ storing at fog nodes.

4.3. Data-owner's usage pattern privacy

Fog-enabled IoT comes with another critical issue: the pattern's privacy of data-owner's utility usage. The usage patterns in IoT applications and the schemes providing patterns privacy are discussed as:

- **Consumption Data Patterns**

In IoT applications such as smart cities, smart homes, and smart grids, the sensors collect the data of electricity consumed by data-owners on a daily basis. The consumption data refers to the habitual activities of the data-owners, such as at what time data-owner switched on/off certain electric appliances, available or unavailable at home, the electricity consumption patterns, etc. Due to the resource's limitation of sensors, the usage pattern of data generated at sensors cannot be preserved using heavyweight operations such as obfuscation and public-key cryptography. Distributed computational resources can be utilized to ensure the pattern's obfuscation at the edge of the IoT network.

- **Unlinkability**

  Most of the proposed schemes in fog-enabled IoT have not considered the usage pattern privacy [13, 16, 41, 42, 109, 120, 129, 134, 140, 142-144, 231-233, 239]. More focus of these schemes has been on preserving the privacy of data-owner's data generated at sensors. Few schemes in fog-enabled IoT have considered the unlinkability of data-owner's pattern to their identity and location [121, 241, 242]. These schemes used the mechanism of machine learning, Voronoi grid, and ECC to provide unlinkability and tracing back to the source node (i.e. sensor and data-owner). Due to unlinkability, the usage patterns cannot be traced back to the data-owner, which makes usage patterns not very useful for malicious users. Although the likelihood of a malicious user successfully sabotaging the IoT network for tracing usage patterns to data-owner is low, it is still essential to have adequate privacy for usage patterns in place.

4.4. Rogue Fog node

The presence of a rogue fog node is a potential threat to fog-enabled IoT network in the context of privacy. The rogue fog node is a type of fog node representing itself to end-users as a legitimate fog node [261]. It is challenging to identify rogue fog nodes due to several reasons. On the one hand, fog computing has a distributed computing, which brings about complex trust situations among fog nodes. On the other hand, fog computing considers numerous devices with creating, deleting, adding, and revoking devices concurrently. For these reasons, it is difficult to detect and manage rogue nodes. The approaches adopted in fog computing to avoid IoT data misuse are:

- **Deduplication Approach**

The deduplication approach is put forward to avoid data misuse by rogue fog nodes [133]. Deduplication is applied to encrypted data and then deduplicated data is forward to the cloud. The encrypted data is further processed at fog nodes. Assuming that the data is exposed at the rogue fog node. The rogue fog node decrypts the data to make changes to it. Due to the

deduplication of data, the changes made will be detected as soon as the rogue node connects with cloud or end-user device.

- **Divide-and-conquer Approach**

The divide-and-conquer approach also mitigates the misuse of data at the rogue fog node [42]. In this approach, the trusted miner encrypts data with a secret key and divides data into blocks. The secret key is hashed and forward to the end-user device. The data blocks are distributed among fog nodes for storage and aggregation processes. Only partial aggregation is performed at fog nodes, remaining aggregation and decryption of data are performed at end-user devices. The rogue fog node has access to the only block of data that is encrypted. The block of data does not provide much information to the rogue Fog node about the entire data packet. For the rogue fog node, the block is useless without a chain of blocks aggregated together. Secondly, the chain of blocks can only be aggregated and decrypted by the end-user device.

Although these approaches avoid data misuse at a rogue fog node, it is still challenging to detect a rogue fog node before data is accessed.

4.5. Certificate Management

Certificate management authority is required to support encryption standards and policies for preserving-privacy of fog-enabled IoT models. The responsibilities of the certificate authority and challenges of certificate management are discussed as follow:

- **Certificate Authority Responsibility**

The responsibility of the certificate authority should be to ensure that the standards and policies are correctly installed and utilized by fog nodes and application domains. Certificate authority should also be responsible for monitoring that the fog policies are correctly defined for the restriction of fog nodes from data access, which may regulate the release of the data owner's locations to third-party [262]. Due to fog computing providing cloud services at the edge of the IoT network, it has a high frequency of data throughput and relatively limited storage for the data backup and recovery process [263]. Therefore, certificate management authority to develop policies and standards for data selection, accessibility roles, mapping, and testing should be considered during backup and recovery processes.

- **Cloud-based Policies for Fog-enabled IoT Network**

Solutions based on fine-grained access control with cloud defined policies have been proposed for IoT applications in fog computing [67, 133, 231, 241]. The methods used in these solutions are based on Attribute-based Signature and Ciphertext Attribute-based Encryption. These methods considered the data owner’s access control and privacy policies defined for cloud for data processing. The mapping of cloud defined policies and standards to fog-enabled IoT is unlikely to be suitable. A cloud purpose is to provide users’ access guarantees and privacy setting

modification, whereas a fog node's purpose is to authenticate entire network nodes and provide lightweight processing. Due to fog nodes' dynamic and mobile nature, federated and distributive role and attribute-based access control architecture is needed. New fog nodes join the fog network after authentication and develop trustworthiness with other fog nodes in a network. To ensure trustworthiness, certificate authority handling policies are needed. Further, there must be a compatible standard for each data-owner's employing different fog devices.

4.6. Communication Overhead

Fog-enabled IoT comes with another important issue: communication overhead for privacy-preserving of data in a distributive manner. The distributive solutions [41, 42, 134, 231, 232, 242] in fog-enabled IoT are based on the following methods:

- **Distributed processing and Storage Methods**

In the blockchain, divide-and-conquer, and crowdsensing methods, the data processing, and storage is distributed among fog nodes. For data transmission, several fog nodes and clusters of fog nodes authenticate themselves to other fog nodes, clusters, sensors, and end-user devices. Instead of a single fog node to perform data processing and storage, several fog nodes are involved during the processing in these methods. Due to distributive interconnectivity and computation, these methods incur high communication overhead.

- **Voronoi Mapping and Ontologies Methods**

The solutions based on Voronoi mapping and ontologies methods also provide low-performance efficiency in terms of communication cost. The distributed communication includes the transmission of data maps and policies, generation and transmission of rule-based languages, and region mapping, data collection, and transmission to fog nodes. The fog devices also make a repeated launch to different fog nodes for communication, thus creating communication overhead and finding a vulnerable point of entry into the network [76].

Although the aforementioned methods provide strong privacy in a distributive manner, an increase in communication cost hinders the full adoption of the schemes in fog-enabled IoT network. Optimization methods based on meta-heuristics approaches such as Ant Colony Optimization (ACO) and Genetic Algorithm (GA) should be introduced to minimize the communication overhead.

Lastly, we believe that the awareness of privacy challenges goes along with the diversified use of a fog network in different IoT application domains. We summarize the challenges based on the analysis of solutions being provided in fog-enabled IoT together with possible future directions. Furthermore, we can see that smart healthcare and smart military system application domains, where information privacy is of utmost importance, need to envision possibilities of fog-enabled IoT to improve their privacy measures and policies.

5. Conclusion

In this paper, we review research work in the area of privacy perseveration in IoT and fog-enabled IoT. We classify and analyze the publications based on the privacy requirements for IoT-based applications followed by the identification and discussions of the state-of-the-art privacy-preserving solutions. Our classification consists of two majors categories of privacy: Content and Context. On one hand, content privacy is further divided into six subcategories: behavior and action, state of body and mind, media, social interaction, data aggregation, and data query. On the other hand, context privacy contains four subcategories: identity, location, temporal, and privacy policy and access control. Then, we discuss the mapping between the existing IoT applications and the fog-enabled IoT applications and identify the key benefits and improvements provided by fog computing. Based on our comprehensive analysis, we summarize key fog computing research challenges for privacy-based IoT designs and future research directions to motivate practitioners and researchers to effectively and efficiently develop privacy-preserving fog computing designs that support more sophisticated IoT applications in the future.

'Data is the pollution problem of the information age, and protecting privacy is the environmental challenge.' *---Bruce Schneier*

Chapter 3: Lightweight, Divide-and-Conquer Privacy-Preserving Data Aggregation in Fog Computing

Abstract

With the increasing popularity of the Internet of Things (IoT) and fog computing paradigm, aggregating IoT data considering privacy concerns over fog networks can be seen as one of the biggest security challenges. Numerous schemes address this problem. However, most of the existing schemes and their associated methods are heavyweight facing issues related to performance overhead. Furthermore, performing data aggregation at a single aggregator fog node causes an overly computational burden on the node, which results in high latency, degraded reliability, and scalability leading to a single point of failure risks. To fill these gaps, this paper presents a lightweight, Divide-and-Conquer privacy-preserving data aggregation scheme in fog computing to improve data privacy, data processing, and storage capabilities. Particularly, we design a data division strategy based on the Level of Privacy (LoP) defined by data owners. The data division strategy not only effectively divides data according to LoP and distributes it among participating fog nodes for aggregation and storage processing, but also reduces computational and memory overhead in the processing simultaneously. Moreover, we perform a privacy analysis of our scheme and perform comprehensive experiments to compare it with other traditional schemes to evaluate performance efficiency. The results demonstrate that our scheme can efficiently achieve data privacy in fog computing and outperforms the other schemes in computational and memory costs.

This Contribution has been published in Future Generation Computer Systems Journal (Elsevier)

1. Introduction

In recent years, the interconnectivity of smart things has significantly improved every day's life including home security, pervasive health care (smart hospitals), infrastructure support, household activities, smart supply chain, smart meters for balancing bills, air quality management, and so on [5]. By the year 2020, Smart devices and sensors connected to the Internet are predicted to reach 34 billion approximately [20]. This rapid development in the Internet of Things (IoT) has increased substantial overhead on data processing to the IoT system [7]. Intuitively, aggregating data to reduce the energy consumption of IoT sensors, data storage costs, data redundancy while improving data analysis speed and computing efficiency has been considered [13]. Utility data of energy companies are aggregated from installed smart meters at customer sites to improve the overall efficiency and reliability of their grid infrastructure [34]. Similarly, various kinds of wearable devices collect aggregated data of the health sector that is needed for medical research [35]. Also, aggregated data collected from vastly installed street and environment sensors analyze a road network to improve transportation services for drivers [36].

Despite the utilization of data aggregation in the IoT system for innovative services, several concerns undermine the full adoption of IoT applications. One of the main concerns is the users' identity and data privacy. Most data owners worry about the potential use of their sensitive or private data collected from different IoT devices and then forwarded to the Cloud to process or store [264]. Such kind of data not only contains general data fields of a user, for example (name, telephone, number, or address). But may also have very sensitive information, including medical health reports and readings of habitual patient behavior that can be accessed by an unauthorized person [18]. Furthermore, in a smart grid, collecting data from hundreds of smart meters and aggregating metering information can also raise issues of consumer privacy such as exposure of activities patterns of consumers and location tracking of consumers [22].

Numerous schemes [56, 265-270] have been adopted to aggregate data while preserving data privacy from entities inside a network, operators, and external eavesdroppers. However, most of the state-of-the-art schemes are computationally expensive and only suitable for homogeneous data [265-269]. Further, performing data aggregation at a Cloud server/single aggregator [131, 271] increases the computational burden on the Cloud server/single aggregator, which results in high latency, degraded reliability, and scalability. Also, the use of a single aggregator may lead to Denial of Service (DoS) and single point of failure risks [56, 131, 265-271].

CISCO's researchers proposed a Fog computing concept in 2012 as an alternative paradigm to solve the aforementioned issues found in many IoT applications [10]. The main idea is based on partly shifting Cloud computing and storage from Cloud data centres to the edge of terminal devices a.k.a. Edge nodes. Fog computing can be viewed as an extension of the Cloud computing paradigm at a network edge [10]. Fog computing is becoming popular as it provides computing, networking, and storage capabilities to IoT's end-users, where each fog node is located closer to IoT devices [12]. Furthermore, the architecture of fog computing can reduce the amount of data transfer and processing to the Cloud. Thus, alleviating much of the burden to fog servers itself and improving performance efficiency [13].

Recently, numerous work [9, 12, 13, 54-56] have proposed schemes for data aggregation in fog computing, and only aggregation results are forwarded to the Cloud. The performance results of these schemes [9, 12, 13, 54-56] show a significant improvement in computation, communication efficiency, and latency as compared to state-of-the-art schemes [56, 131, 265-271]. However, within a fog layer, the utilization of multiple nodes to distribute the workload of data aggregation has not been considered. Thus, making their schemes [9, 12, 13, 54-56] vulnerable to DoS attack and single point of failure risks. Also, there has been no other fog node integrated into a network to minimize a failure probability of a fog node during data aggregation.

Most of the schemes [9, 56] are based on heavyweight cryptosystems (for example, pairing-based cryptosystems), which increase computational and storage costs within the fog layer. Cryptosystems also involve a third-party authority to generate public/private keys, which may increase communication overhead and eavesdrop attacks on third-party authority or a communication link. Further, the fog layer has not provided authority to data owners for defining the level of privacy of their generated data.

Recently, Sharma *et al.* [12] proposed a scheme based on blockchain for secure data distribution among fog nodes and to mitigate a single point of failure risk, and optimize fog resources for data processing. Although the scheme can optimize performance efficiency during data distribution, however, aggregating data to reduce redundancy has not been considered in a scheme. Their scope is only limited to distribute data securely among fog nodes for data storage.

Despite the numerous benefits that fog computing provides regarding preserving privacy during data aggregation, the utilization of heavyweight cryptosystem, single processing node, third-party involvement, and no consideration of data owners for data authority still hinder the full utilization of fog computing. Therefore, we propose a lightweight, Divide-and-

Conquer framework to achieve efficient data aggregation while preserving a higher level of data privacy in fog computing. The main contributions of this paper are summarized as follows:

1. We propose a lightweight Divide-and-Conquer approach to preserve the data privacy for data aggregation in fog computing together with a scheme for the distribution of data processing among fog nodes to mitigate a single point of failure risks.
2. We propose a data division strategy based on the Level of Privacy (LoP) defined by a data owner. This strategy provides authority to the data owner to define LoP for the privacy of their sensor-generated data in fog computing.
3. We evaluate the performance efficiency, and the results show that our scheme can efficiently preserve the privacy of data in fog computing as compared with state-of-the-art schemes.

The remainder of the paper is organized as follows: In Section 2, we review related work, and we present our scheme with an in-depth analysis in Section 3. Section 4 provides the security analysis and performance evaluation of our proposed scheme compared with existing ones. Finally, Section 5 concludes the paper with future work.

2. Related Work

Recently, several research works [9, 12, 13, 54-56], have been carried out to perform data aggregation and storage processing in fog computing. With their approach, Fog nodes only forward the aggregation results to the Cloud, which has improved communication efficiency and latency delays. A secure and anonymous data aggregation scheme using fog computing has been proposed by Wang *et al.* [9]. In the proposed scheme, a fog node aggregates data from sensor devices and forwards the aggregated data to the cloud for long-term storage [9]. The scheme is based on Castagnos-Laguillaumie, short-signature, and bilinear pairing cryptosystem to provide secure aggregation and identity privacy at fog edges [9]. However, the scheme provides data aggregation using a single fog device, which can be vulnerable to DoS attack and a single point of failure risk. The adversary model of the scheme is also limited, considering only the possible internal attacks. Furthermore, in the case of fog device failure, there has been no other fog node integrated into the network to recover the aggregated data.

Camillo *et al.* have also proposed a lightweight data aggregation scheme using a fog device, which provides resistance to false data injection with efficient data aggregation and supporting fault tolerance [56]. However, data is kept at a centralized unit 'Control Centre' which may lead to a single point of failure risk. Also, the scheme considers the Paillier cryptosystem for data encryption, which is a computationally expensive public-key cryptosystem [255]. Therefore, there is a need for distributed data storage with a computationally inexpensive cryptosystem.

Also, the workload of a single fog device should be distributed to decrease the network bottleneck. Furthermore, third-party authority is used for the generation of public and private keys. Involving third authority to perform key generation may also increase communication overhead and eavesdrop attacks on third party authority or communication link.

To overcome DoS attacks, Sharma *et al.* proposed a technique based on blockchain for secure data distribution among fog nodes with optimization of the performance efficiency and fog resources [12]. The authors claim that their technique is better in reducing response time delays, increasing throughput, and detecting real-time attacks as compared to existing techniques. One of the aims of using blockchain technology in the Sharma *et al.* technique is to provide complete data privacy in fog computing [12]. However, the scope of this blockchain technique is only limited to distribute data among fog nodes for storage securely. Also, there is no data processing, for example, the aggregation process on data. Also, the proposed technique lacks to provide energy-efficient communication methods between fog nodes themselves, which results in limitation to workload balance and resource allocation in fog environment.

Basundan *et al.* adopted the concept of crowdsensing for data aggregation using fog computing and encryption without the need for data verification certificates [231]. The concept provides resilience to key escrow attacks with less computational and communication overhead. In this concept, third-party authority is not needed for generating aggregation certificates. Another scheme based on the Paillier cryptosystem and online/offline signature method for data aggregation using fog computing has been proposed [272]. The scheme achieved data privacy preservation, authentication, and confidentiality during data aggregation. However, the scheme is based on an asymmetric cryptosystem, which results in higher computational overhead. Guan *et al.* [7] also proposed a scheme based on the Paillier cryptosystem for preserving privacy during data aggregation in fog-enhanced IoT. Data aggregation in both the Paillier based schemes [7, 272] is carried out on a single aggregator node, and third-party authority is involved for key generation. Therefore, both schemes are vulnerable to a single point of failure attack.

In another scheme, Yang *et al.* [13] have applied a machine learning approach to achieve data privacy in fog computing, data is distributed among two fog nodes, and data training is carried out on raw data. Additive and non-additive aggregation of raw data is also provided on fog nodes. Then only the aggregated results from fog nodes are sent to the Cloud. Authors have also claimed that using a machine learning technique to preserve data privacy during aggregation improves performance efficiency as compared to cryptographic heavyweight technique, for example (Paillier homomorphic encryption). Whereas, we contradict their claim as lightweight cryptographic functions can provide the same level of performance efficiency with a high level of security as compared to machine learning training and testing techniques. Also, an enormous amount of data is produced by IoT devices, and performing machine learning

techniques on that data will be time-consuming. Furthermore, sensor data has been not kept private from fog nodes during the data training procedure in the proposed machine learning approach, which may lead to data exposure in the case of a malicious fog node in a network.

3. The Divide-and-Conquer Scheme

In this section, we provide an overview of our proposed system model and the adversary model. We also present the outline of our proposed system model in Figure 11. In this section, we also present our privacy-preserving scheme based on the system and the adversary model. The visual summary of the proposed scheme's phases is shown in Figure 12 and discussed in subsequent sections/sub-sections.

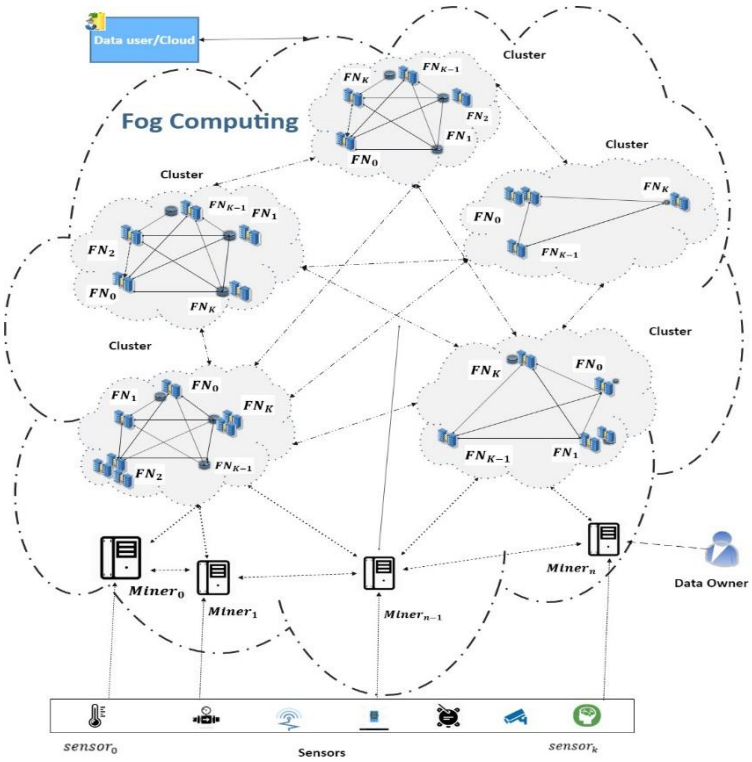


Figure 11 An overview system model of the proposed scheme

3.1. System and Adversary Model

The system model as shown in Figure 11, comprises three layers and a data owner. The first layer consists of sensor nodes to transmit generated data of data owners to a second layer. The second layer is the fog layer, which consists of two sub-layers (Miner layer and Cluster layer). The second layer is responsible for performing most of the processing and storage tasks. Only the aggregated result is forwarded to the third layer, which consists of the end-user/cloud device. All the entities in each layer of the system are discussed as follow:

- The sensor devices ($Sensor_0 - -, Sensor_k$, where k represents the total number of sensor devices in a network) record the raw data and sends data in established JSON format to the fog layer for processing.

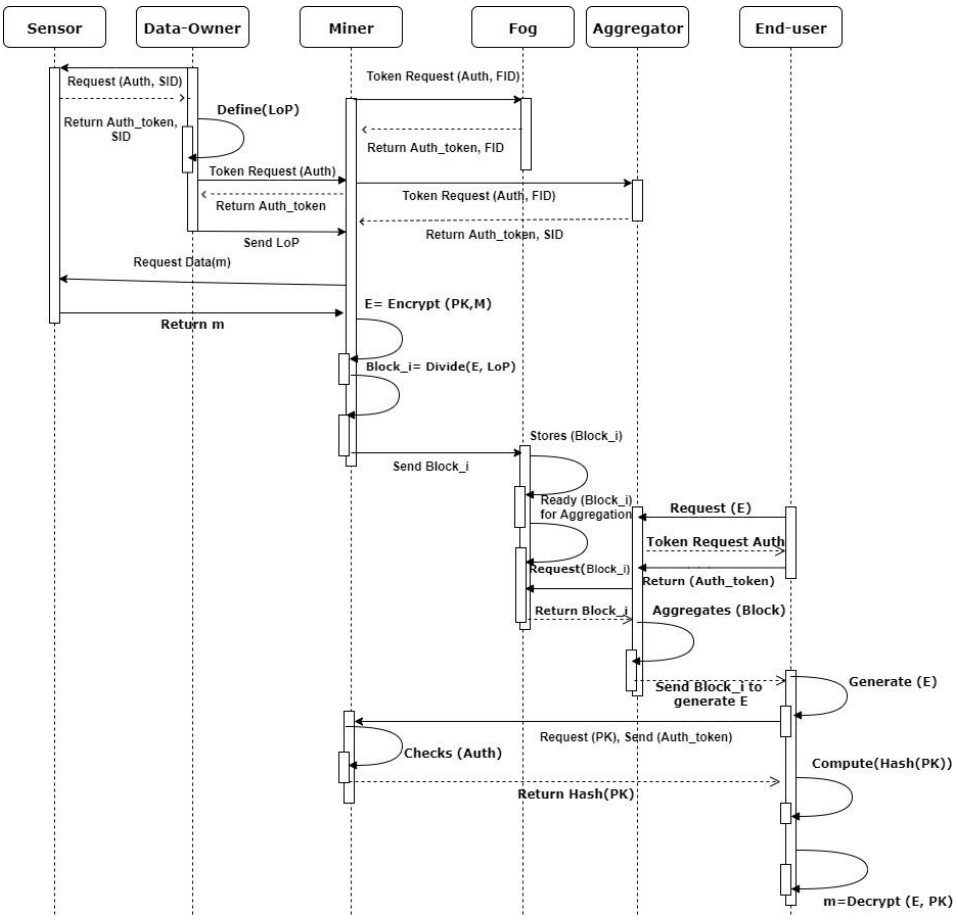


Figure 12 Sequence Diagram (SSD) for Divide-and-Conquer privacy-preserving data aggregation scheme

- In our model, the fog layer is a critical component for providing services, including fetching, analyzing, and processing sensor devices generated data. For providing these services, the fog layer is subdivided into two layers, the miner layer, and the cluster layer.
- The miner layer consists of miner nodes $(Miner_0 - -, Miner_n$, where n represents the total number of miner nodes in a network). The miner nodes are deployed at the sensor network edge. Miner nodes serve as an access point for sensor devices to fog nodes of the cluster layer. In particular, miner nodes $(Miner_n)$ fetch data from sensor devices and update the privacy table with data packet Id. Further, process the data of sensor devices according to the defined Level-of-Privacy (LoP) by a data owner.
- In our model, a data owner is considered as an individual or a device, which owns the data generated by sensor devices. The data owner has the authority to define the Level of Privacy (LoP) for each data sets generated on different sensors. According to the LoP settings, miner nodes can process data and forward it to the cluster layer.
- In a cluster layer, different clusters of fog nodes $(FN_0 - -, FN_k$, where k represents the total number of fog nodes in the network) receive processed data from miner nodes $(Miner_n)$. The fog nodes store the processed data coming from the miner nodes. Further on a request of end-user/cloud device $(User_{device})$ for the stored data, the fog nodes in a

cluster act as aggregators and perform the data aggregation on stored data. This aggregation process is distributed among the fog nodes. Then the end-user/cloud device ($User_{device}$) receives the aggregated data from the fog layer, which can be decrypted by the end-user/cloud device for generating results.

In the adversary model, we consider the cases which are important for preserving the privacy of data during aggregation in fog computing. Miners in the fog layer are considered fully trusted, as they only process sensor data at the network edge close to sensors and are not involved in the aggregation process. Also, each miner node is unaware of data packets from other miner nodes.

Fog nodes in the cluster layer are honest-but-curious because they may be affected by undetected malware which may compromise data privacy. The scope of our current work is limited to the privacy preservation of data within the fog layer only. Therefore, IoT and user devices are not compromised. Furthermore, any adversary can be strong enough to perform actions as an internal attack and external attack. In an internal attack, an adversary can compromise the fog node in a cluster layer and gain access to data kept at that node. Whereas in the external attack, an adversary impersonates as one or a group of fog nodes and sends false data on behalf of data coming from miners to the aggregator. Also, it can mount Denial of Service (DoS) attacks on fog nodes.

3.2. Network Design and Setup

Before defining our scheme, it is important to understand how the network can be designed. Both layers in the fog layer have their dedicated roles and tasks to perform. The layer division helps to balance the load and increases performance efficiency to provide privacy. Generally, all the fog and miner nodes are interconnected in the fog layer with each other, and the fog nodes are assigned with memory and computational power. The network formation between sensor nodes and miner nodes in the miner layer depends upon the following possibilities:

- The shortest distance between the sensor and the miner node.
- In the case, there exists more than one miner node at the same shortest distance from the sensor node then the miner node with the highest memory and computational resources available will be selected.
- In the case, there exists more than one miner node, and all the miner nodes have the same resources available. Then on a first come first serve basis, the miner node which authenticates itself first to the sensor node will be considered for data transmission.

In the cluster layer, clusters can be formed based on KHOPCA: k-hop clustering [273] concept and distance factors between nodes. Each fog node F_k in the cluster layer is continuously involved in cluster formation. Fog node in the cluster layer is assigned with dynamic weights,

which are randomly selected from the maximum (Max) and minimum (Min) weights range. In the start configuration, Max weight is the maximum weight, which is assigned to the assumed cluster fog head and Min weight is the smallest possible weight for a fog node in a network. We assume that the fog nodes awareness available to the fog node F_{k-1} is only the nearest located neighbor fog node F_k with weight W_k. Clusters are created by following the k-hop clustering state transition rules in [273]. The first rule deals with the top-to-down hierarchical structure by setting-up the fog node F_{k-1} weight to the closely located neighbour node's F_k with the highest weight subtracted by 1 (i.e. W_k -1). The second rule deals with declaring the isolated fog node as a fog-head. An isolated fog-node is a node which has minimum weights and is fog-head-less.

The third rule deals with the situation where a higher weighted fog node, which is not a fog-head attracts the surrounding less weighted fog node towards itself. According to the rule, the higher weighted fog node decreases its weight to join an existing nearby cluster. The fourth rule resolves the very close proximity fog-heads conflicting situations. In such a situation, one fog-head survives, while others must die [273].

The miner nodes authenticate each fog node in a cluster for nodes' awareness and communication with miner nodes. Each miner node creates a policy table of each fog node of a cluster with available resources and distance from the miner node. According to the policy table, the miner node sends data blocks to the closest available fog nodes.

3.3. Level of Privacy (LoP) and Distribution Setup

In our proposed scheme, data distribution is based on the Level of Privacy (LoP) settings defined by a data owner who is authorized to select the LoP for their sensors data. LoP consists of five levels. Level 5 represents the highest privacy level, whereas level 1 represents the lowest privacy level. The higher the privacy level, the requirement of privacy is greater, which indicates that during processing and storage in fog computing, the data should be kept more secure and private as compared to a low level of privacy settings. For each LoP, the data distribution (i.e. linear and tabular) type varies. The following Equations identify the type of distribution and division factor to be used for LoP.

$$\boldsymbol{Type1} = \boldsymbol{Linear}, \boldsymbol{LoP} \leq ((MaxLoP + MaxLoP\ mod 2)/2) \quad \textbf{eq 3.1}$$

$$\boldsymbol{Type2} = \boldsymbol{Tabular}, \boldsymbol{LoP} > (MaxLoP + MaxLoP\ mod 2)/2 \quad \textbf{eq 3.2}$$

$$\boldsymbol{For\ Type1}, \quad \boldsymbol{D.F} = 2 * LoP \quad \textbf{eq 3.3}$$

$$\boldsymbol{For\ Type2}, \quad \boldsymbol{D.F} = 2^{LoP} \quad \textbf{eq 3.4}$$

Where MaxLoP represents the maximum level of privacy that is considered as 5. Type1 represents linear distribution type, whereas Type2 represents tabular distribution type, and DF is for division factor. According to the Equations mentioned above, Table 7 shows the LoP and distribution setup for the proposed scheme.

Table 7 Distribution setup

LoP	Type	DF
1	Linear	2
2	Linear	4
3	Linear	6
4	Tabular	16
5	Tabular	32

In linear distribution, data is divided into a number of columns based on the division factor (DF). For example, from Table 1 above, if LoP is 2, then DF is 4 and type is linear, which means that basic privacy should be applied with linear data division by dividing a data packet into four numbers of data blocks. Let S represents the total size of the encrypted data. For linear distribution, the size of each block becomes:

$$\boldsymbol{For\ Linear}, \boldsymbol{B} = \lceil \mathbf{S}/\mathbf{D}.\mathbf{F} \rceil \qquad \textbf{eq 3.5}$$

For tabular distribution, data is first divided into rows and then into columns as discuss as follow:

Step 1: For dividing data into rows, first, we identify a total number of rows $\boldsymbol{D}$ as:

$$\boldsymbol{For\ Rows}, \boldsymbol{D} = \lceil \sqrt{\boldsymbol{D}.\boldsymbol{F}} \rceil \qquad \textbf{eq 3.6}$$

Now the length $\boldsymbol{r}$ of each row will be the division of the size of each block to a total number of rows:

$$\boldsymbol{r} = \lceil \mathrm{S}/\mathrm{D} \rceil \qquad \textbf{eq 3.7}$$

Step 2: After row division for column distribution, rows from Equation 3.7 will be divided into columns as:

$$\textbf{For Tabular, T} = \lceil \mathrm{r}/\mathrm{D}.\mathrm{F} \rceil \qquad \textbf{eq 3.8}$$

From Equations 3.6, 3.7, and 3.8, it can be deduced that for the tabular distribution, i.e. LoP levels 4 and 5, the data is first divided into a number of rows. Then further each row is divided into a number of columns.

3.4. Nodes Authentication

Before sensor nodes transmit data for processing and storage to the fog layer, each node in the network proves its legitimate identity and authorization using a token-based authentication mechanism. Each node generates the packet fields including secret, type of token, hashing algorithm, and node ID in the header and payload. Also, algorithmically signs the packet to produce a token. Nodes send token to other nodes with whom they wish to have data communication. Using header-claim-signature with Auth0, nodes can verify generated secrets for authorization and legitimacy of each other. In the case of an incorrect secret inside a packet transmitted, an invalid signature in a token will be computed and will differ from the original token. Thus, the data communication request will be denied with an 'unauthorized node' status.

Table 8 Symbols used in Algorithms

Symbols	Definition
$\boldsymbol{M}$	Data packet from the sensor node
$\boldsymbol{k}$	security key
$\boldsymbol{s}$	Randomly generated number
$\mathbf{pktid}$	Packet Identification bits
$\boldsymbol{LoP_{s^i}}$	Level of privacy defined by data owner for the sensor at i^{th} position.
$\mathbf{DF}$	Division factor
$\boldsymbol{S}$	The total size of the encrypted data
$\mathbf{B}$	Size of a block
$\boldsymbol{R_i}$	Unique number for a block of encrypted data
$\mathbf{D}$	Integral number randomly selected
$\boldsymbol{c_i}$	length of the ith column

3.5. Data Encryption

Miner nodes are responsible for performing sensor data encryption. After proving the authentication, miner nodes receive JSON format data M from sensor nodes and apply lightweight Advanced Encryption Standards (AES) symmetric technique to encrypt data M , as shown in Algorithm 1. The symbols used in Algorithms are in Table 8. First of all, at the miner node, each packet's id is hashed for future use and is kept in a packet Id Pkt_i table. Then AES, Rijndael block cipher technique is applied on data M. The first step is the derivation of a new round of keys k_i with 128, 192, and 256 bits using Rijndael's key schedule. Then data M is divided into column blocks, for example, if an AES block size is 128 bits then four by four-columns of 16

Algorithm 1 Data Encryption

Procedure $\boldsymbol{AES_k(M)}$

Input $(\boldsymbol{M}, \boldsymbol{k}, \boldsymbol{s},)$

For each data packet at Miner node from a sensor

Hash packet Id

1. $\boldsymbol{Pkt_i \leftarrow Hash_s(\mathbf{pktid})}$

For each $\boldsymbol{key\ k}$ length at Miner nodes

Initialize initial round

$\boldsymbol{Rounds \leftarrow\ 1}$

2. If $\boldsymbol{K_{len} \leftarrow 128}$ bits

Then $\boldsymbol{Rounds \leftarrow 10}$ times

Else if $\boldsymbol{K_{len} \leftarrow 192}$ bits

Then $\boldsymbol{Rounds \leftarrow 12}$ times

Else if $\boldsymbol{K_{len} \leftarrow 256}$ bits

Then $\boldsymbol{Rounds \leftarrow 14}$ times

For each M, perform Sub Bytes, Shift Rows, Mix columns, Add round key

Assign state for each $\boldsymbol{M}$

3. $\boldsymbol{state[i] \leftarrow M[i]}$
4. SubByte$\leftarrow (\mathbf{state[i]})$

Set Rijndael's S-Box for n= 128, 192 and 256

5. $\boldsymbol{S-box[n]\ \leftarrow\ (0X63, \ldots\ldots\ldots, 0X16)}$
6. $\boldsymbol{state[i] \leftarrow S-box\left[state[i]\right]}$
7. ShiftRows$\leftarrow (\mathbf{state[i]})$

For shifting rows, assign a temp number to the state

8. $\boldsymbol{temp[0] \leftarrow (state[0]),\ temp[1] \leftarrow (state[5]), - -\quad --, temp[15]\ \leftarrow\ state[11]}$
9. $\boldsymbol{state[i] \leftarrow tmp[i]}$
10. MixColumn$\leftarrow (\mathbf{state})$

Fetch multiples from Rijndael mix columns

11. $\boldsymbol{temp[0] \leftarrow (mul2[state[0]]\text{^}mul3[state[1]]\text{^}state[2]\text{^}state[3]), - - -, temp[15]\ \leftarrow}$ $\boldsymbol{(mu13[state[12]]\text{^}state[13]\text{^}state[14]\text{^}mu12[state[15]])}$
12. $\boldsymbol{state[i] \leftarrow tmp[i]}$
13. $\boldsymbol{AddRound\ \leftarrow\ (state, k)}$ ∵ repeat rounds
14. $\boldsymbol{state[i] \neq\ key[i]}$
15. $\boldsymbol{En_M \leftarrow SubByte, ShiftRows, MixColumns, AddRound(M)}$

Append hashed packet id with $\boldsymbol{En_M}$

16. E$\leftarrow\ \boldsymbol{En_M + Pkt_i}$

Applies Divide-and-Conquer method on E as shown in Algorithm 2

End Procedure (E)

bytes will be created. After division into column blocks, round keys are added to the blocks using an additive encryption Algorithm. The next step is to substitute each data byte $\boldsymbol{M[i]}$ into $\boldsymbol{state[i]}$ according to the Rijndael pre-determined table in S-Box.

After this step, each block $\boldsymbol{state[i]}$ row is shifted, and columns are mixed to further diffuse the block. Then the keys k_i which are derived by the miner node in the first step are added to the final $\boldsymbol{state[i]}$**.** The steps will be repeated for each round (i.e. for 128 bits 10 rounds, 192 bits 12 rounds, and 256 bits 14 rounds) and $\boldsymbol{state[i]}$ is altered at every stage. Finally, after the completion of rounds, encrypted data is derived. Encrypted data is appended with packet id Pkt_i**.** The appended result is then inserted into the data division and distribution method, as shown in Algorithm 2 for further processing.

3.6. Divide-and-Conquer Scheme

Based on the LoP defined for each sensor node by a data owner, the tabular and linear distribution table is created, as mentioned in sub-section 3.3. According to the Table, each encrypted data packet from Algorithm 1 is divided into blocks, as shown in Algorithm 2. At first, miner nodes check the $\boldsymbol{LoP_{s^i}}$ defined by the data owner for each sensor data. In the case of $\boldsymbol{LoP_{s^i} = 1, 2\ or\ 3}$, the miner nodes apply linear distribution to the encrypted data by dividing data into a number of blocks, as discussed in sub-section 3.3. Also, the miner node generates a unique number $\boldsymbol{R_i}$ and block hash $\boldsymbol{B_i}$ for each block and keeps generated $\boldsymbol{R_i}$ **and** $\boldsymbol{B_i}$ in $\boldsymbol{table_{policy}}$**.** Then the miner nodes append blocks with the previous block hash $\boldsymbol{B_{i-1}}$ as $\boldsymbol{E_{i_i} = E_i + B_{i-1}}$.

In the case of $\boldsymbol{LoP_{s^i} = 4\ or\ 5}$, the miner nodes apply tabular distribution as mentioned in sub-section 3.3. Based on Equations 3.6, 3.7, and 3.8, all blocks are first divided into rows and then divided into columns. The blocks are also appended with the previous block hash $\boldsymbol{B_{i-1}}$. Only the resultant blocks $\boldsymbol{E_{i_i}}$ are distributed among the fog nodes (FN_k) in the cluster layer.

In our proposed scheme, we perform data M encryption before data M tabular and linear division. The reason for data M encryption first is to reduce the complexity and generation of Rijndael's rounds for each block $(E_0 + \cdots + E_i)$ of data M separately. If we first divide data M into blocks according to the LoP. Then separately apply encryption on each block (i.e. *AES* $(E_0), \ldots, AES(E_i)$*)* requires higher key derivations and rounds of Rijndael's block ciphers simultaneously for each block. In contrast, encrypting data M requires a single Rijndael's block cipher process. After the blocks $\boldsymbol{E_{i_i}}$ generation, $\boldsymbol{E_{i_i}}$ are stored at fog nodes, and on end-user/cloud requests for aggregated data, the stored blocks at fog nodes are sent to aggregator nodes for aggregation.

Algorithm 2 Divide and Conquer

Procedure $\boldsymbol{D(E)}$

Input **(*E*)**

After data encryption, Miner nodes check the $\boldsymbol{LoP}$ for each sensor

1. $\boldsymbol{LoP_{s^i} \leftarrow \text{Data\_owner}\ (1,2,3,4,5)}$
2. if $\boldsymbol{LoP_{s^i} == 1,2\ or\ 3}$ $\because$ **From Table 2**
3. Then type $\leftarrow$ **Linear** &&
4. $\boldsymbol{D.F = 2 * LoP_{s^i}}$ $\because$ **From equation 3.3**
5. $\boldsymbol{B \leftarrow \lceil S/D.F \rceil}$ $\because$ **From equation 3.5**

According to block length $\boldsymbol{B}$, divide E $\because$ **Algorithm 1**

6. $\boldsymbol{(E_0 + \cdots + E_i) \xleftarrow{division} E_n}$

$\boldsymbol{R_i}$ is generated for each data block and kept in a table $\boldsymbol{table_{policy}}$

7. $\boldsymbol{table_{policy} \xleftarrow{contains} R_i \in E_i}$

Each block hash ($\boldsymbol{B_i}$) is generated and associated $\boldsymbol{with\ R_i}$ in $\boldsymbol{table_{policy}}$

8. $\boldsymbol{B_i = Hash(E_i)}$
9. $\boldsymbol{table_{policy} \xleftarrow{contains} B_i\ w.r.t\ R_i}$

Each block is appended with a hash of the previous block

10. $\boldsymbol{E_{i_i} = E_i + B_{i-1}}$
11. Else $\boldsymbol{LoP_{s^i} == 4\ or\ 5}$
12. Then type$\leftarrow$ **Tabular** &&
13. $\boldsymbol{D.F = 2^{LoP}}$ $\because$ **From equation 3.4**

Divide data into rows

14. $\boldsymbol{D = \lceil \sqrt{D.F} \rceil}$ $\because$ **From equation 3.6**
15. $\boldsymbol{r_i = \lceil S/D \rceil}$ $\because$ **equation 3.7**

According to $\boldsymbol{r_i}$, divide E $\because$ **E from Algorithm 1**

16. $\boldsymbol{(E_0 + E_1 + E_2 + \cdots + E_i) \xleftarrow{division} E_n}$ $\because$ $\boldsymbol{LoP_{s^i} = 4\ or\ 5}$
17. Repeat step 7, 8, 9 and 10

Divide rows into columns

18. $\boldsymbol{T = \lceil r/D.F \rceil}$ $\because$ **From equation 8**

According to the T divide $\boldsymbol{(E_0 + E_1 + E_2 + \cdots + E_i)}$

19. $\boldsymbol{(E_{0_0} + .. + E_{0_i}) + \ldots + (E_{k_0} + \cdots + E_{k_i}) \xleftarrow{division} (E_0 + .. + E_i)}$
20. Repeat step 7, 8, 9 and 10

$\boldsymbol{Miner_n}$ holds $\boldsymbol{table_{policy}}$ of $\boldsymbol{R_i}$ with a hash of each block $\boldsymbol{B_i}$

$\boldsymbol{Miner_n}$ sends data blocks to $\boldsymbol{FN_k}$

21. $\boldsymbol{FN_k \xleftarrow{send} (E_{0_0} + .. + E_{0_i}) + \ldots + (E_{k_0} + \cdots + E_{k_i})}$

End Procedure $\boldsymbol{(E_{0_0} + .. + E_{0_i}) + \ldots + (E_{k_0} + \cdots + E_{k_i})}$

3.7. Data Aggregation and Decryption

In our proposed scheme, we perform an additive aggregation process to compute the sum aggregate of all sensor node's data for end-user/cloud usage. Firstly, the data blocks stored at the fog nodes are aggregated together by aggregator nodes. For aggregation, the miner node selects the fog nodes, which has computing resources available for data aggregation to act as an aggregator node ($Fognode_{aggregator}$). The fog node selection as an aggregator node is only

done when the end-user/cloud requests a miner node for the aggregated data. After selection, the aggregator node requests the clusters to send the data blocks stored at fog nodes. The aggregation process is not performed at a single aggregator, and aggregation is distributed among aggregator nodes to speed up the process and minimize a single point of failure risk. The fog nodes ($\boldsymbol{FN_k}$) forward data blocks to the closet available $Fognode_{aggregator}$. After receiving data blocks, aggregators request miner nodes to send $table_{policy}$.

The purpose of requesting $table_{policy}$ is to check $\boldsymbol{B_i}$ and corresponding $\boldsymbol{R_i}$ for summing up blocks together. Once $Fognode_{aggregator}$ receivers table $table_{policy}$, then aggregator node checks $\boldsymbol{B_{i-1}}$, which is appended with blocks received from $\boldsymbol{FN_k}$. If the appended hash of the previous block $\boldsymbol{B_{i-1}}$ with a block match with the one in the $table_{policy}$ then $Fognode_{aggregator}$ aggregates blocks together. Else $Fognode_{aggregator}$ checks a previous hash $\boldsymbol{B_{i-1}}$ to be null, then $Fognode_{aggregator}$finds out the next hash value $\boldsymbol{B_{i+1}}$ of a block and compare it with $\boldsymbol{B_{i+1}}$ in $table_{policy}$, if both the hash values are equal then $Fognode_{aggregator}$ aggregates blocks.

Otherwise $Fognode_{aggregator}$ forwards data block to another $Fognode_{aggregator}$, as shown in Algorithm 3 to find out hashes of the block and aggregate blocks chain. After the summation of data blocks, all the aggregator nodes $Fognode_{aggregator}$ send blocks chain to end-user/cloud device to perform the additive aggregation on data.

First, the end-user/cloud U_i device aggregates all data blocks chain ($\boldsymbol{E_{f_0}} + \cdots + \boldsymbol{E_{f_i}}$) together to generate one file $\boldsymbol{E_l}$. Then U_i requests $\boldsymbol{Miner_n}$ nodes to send hashed Pkt_i and corresponding keys k for decrypting data packets in $\boldsymbol{E_l}$. Using AES decryption method for each data packet ($\boldsymbol{E_0} - - - \boldsymbol{E_n}$) kept in $\boldsymbol{E_l}$, the end-user/cloud device decrypts each data packet. Also, sums up the data packets ($\boldsymbol{E_0} + - - + \boldsymbol{E_n}$) in $\boldsymbol{E_l}$ to only provide the summation of data to end-user/cloud for further processing or usage.

Algorithm 3 Data Aggregation & Decryption

Procedure $\boldsymbol{Agg_{pk}(E)}$

Input $(\boldsymbol{E_n}, \boldsymbol{table_{policy}})$

$\boldsymbol{Fognode_{sender}}$ forwards data blocks to $\boldsymbol{Fognode_{aggregator}}$

1. $\boldsymbol{E_{i_i}} \xleftarrow{\boldsymbol{Sends}} \boldsymbol{Fognode_{sender}}$
2. $\boldsymbol{E_i + B_{i-1}} \xleftarrow{\boldsymbol{computes}} \boldsymbol{E_{i_i}}$

$\boldsymbol{Fognode_{aggregator}}$ requests $\boldsymbol{Miner_n}$ to send $\boldsymbol{table_{policy}}$

3. Checks $\boldsymbol{table_{policy}}$ for $\boldsymbol{R_i}$ and $\boldsymbol{B_i, B_{i-1}}$ $\boldsymbol{and}$ $\boldsymbol{B_{i+1}}$
4. If $\boldsymbol{B_{i-1}} \in \boldsymbol{E_{i_i}} == \boldsymbol{B_{i-1}} \in \boldsymbol{table_{policy}}$
5. Then $\boldsymbol{E_{f_i}} \xleftarrow{\boldsymbol{aggregate}} \boldsymbol{E_i + E_{i-1}}$
6. $\boldsymbol{tab}[\boldsymbol{B_i}] - -$ $\therefore$ keep finding previous hashes
7. $\boldsymbol{e}$lse $\boldsymbol{If}$ $\boldsymbol{B_{i-1}} \leftarrow$ Null

8. $\boldsymbol{then\ find\ B_{i\ and}\ B_{i+1}}$
9. If $\boldsymbol{B_{i+1} \in E_{i+2_{i+2}} == B_{i+1} \in table_{policy}}$
10. Then $\boldsymbol{E_{f_i} \overset{aggregate}{\longleftarrow} E_i + E_{i+1} + E_{i+2}}$
11. $\boldsymbol{tab[B_i]}$++ ∴ keep finding next hashes
12. $\boldsymbol{Else\ Fognode_{aggregator+1} \overset{send}{\longleftarrow} E_{i_i}}$
13. $\boldsymbol{Repeate\ steps\ (2-11)}$

Each Aggregator node sends $\boldsymbol{E_{f_i}}$ to an end-user device to compute final aggregation and decryption.

14. $\boldsymbol{E_l = E_0 - - - E_n \overset{aggregate}{\longleftarrow} E_{f_0} + \cdots + E_{f_i}}$
15. $\boldsymbol{Pkt_i, k \overset{requests}{\longleftarrow} U_i}$
16. $\boldsymbol{E_l = Decrypt_{Pkt_0,k}(E_0) + \cdots + Decrypt_{Pkt_n,k}(E_n)}$

End Procedure ($\boldsymbol{E_l}$)

4. Privacy and Performance Analysis

In this section, we perform the privacy and performance analysis of our scheme and evaluate our scheme as compared to other schemes.

4.1. Experiment Setup

A set of AES-based Algorithms with security parameters 128, 192, and 256 are implemented in C++ using Network Simulator based on a 500 MHz Linux based-system. Crypto++ library is used for the implementation of AES with hashing algorithms. Using the network simulator, we considered 10 KB to 1000 KB varying data sizes for analyzing and comparing schemes. The variation of data size is based on different case scenarios:

1. Simple Level: In the simple level of network complexity, we consider data sizes vary between 10 KB to 200 KB with 3-10 sensor nodes, 2-5 miner nodes, and 5 -15 fog nodes in 2-5 of clusters and an end-user device.
2. Medium Level: In the medium level of network complexity, we consider data sizes varying 200 KB – 500 KB with 10-40 sensor nodes, 5-15 miner nodes, 10-50 fog nodes in 5- 20 clusters, and an end-user device.
3. High Level: In the high level of network complexity, we consider data sizes varying 500 KB – 1000 KB with 40-100 sensor nodes, 15- 30 miner nodes, 50-80 fog nodes in 15- 40 clusters, and an end-user device.

   We present our results in the form of graphs. The graph illustrates the computational, memory, and communication overhead for our scheme compared with the ECBDA [22], Masker [131], APPA [7], and LVPDA [272] schemes.

4.2. Formal Security Analysis

We consider Ouafi *et al.* & Gope *et al.* privacy models [57, 58] to formally analyze the privacy of our scheme. In the privacy model of our scheme, an adversary A can eavesdrop fog nodes and

communication channels between miner nodes and end-user device to gain data block E_{k_i}, E_{f_i} or to target a fog node. An adversary can also try to decrypt $(E_{k_i} or\ E_{f_i})$, and forward amended $E_{i_{i'}}$ to fog nodes or E_{f_i}, to end-user device. An adversary A can also perform any active or passive attacks and be allowed to run the following queries.

1. **Execute** $(Miner_n, FN_k, Fognode_{aggregator}, m)$: This query depicts the passive attack for an adversary A. In this query, A can eavesdrop on the transmitted data block E_{k_i} between the $Miner_n$ and FN_k or between FN_k and $Fognode_{aggregator}$ in the mth session where (0 <m<Total (execute)). A can also eavesdrop on the aggregated data blocks E_{f_i}, which is transmitted from $Fognode_{aggregator}$ to U_i.
2. **Send** $(FN_k, Fognode_{aggregator}, E_{k_i}, E_{f_i}, m)$: For A, this query represents the active attack in the fog network. In this query, A has permission to impersonate a fog node FN_k or an aggregator node $Fognode_{aggregator}$ in the mth session. An adversary A also has permission to forward amended $E_{k_{i'}}$ to $Fognode_{aggregator}$ and $E_{f_{i'}}$ to U_i. Besides, A can block the exchanged E_{k_i} or E_{f_i} between FN_k and $Fognode_{aggregator}$.
3. **Corrupt**$(FN_k, Fognode_{aggregator}s, E_{k_i}, E_{f_i})$: A has permission to access a data block E_{k_i} stored at fog node FN_k or aggregated data blocks E_{f_i} at $Fognode_{aggregator}$. A can corrupt E_{k_i} or E_{f_i} as $E'_{k_i} or E_{f_i}{}'$.
4. **Test** (FN_0, FN_1, m): This query defines the indistinguishability-based notion of untraceable privacy. The test is the only query, which does not correspond to A's abilities to perform active or passive attacks as this query only defines the notion of untraceable privacy. An adversary A is given FN_b from the set $\{FN_0, FN_1\}$ depending on the randomly chosen bit b {0, 1} by a fog node in the mth session. Then A decrypts the block ${E_{k_i}}^{FN_b}$ or aggregates and decrypts ${E_{f_i}}^{FN_b}$. Informally, an adversary A succeed if it correctly guesses the bit b, and correctly aggregates and decrypts E_{k_i} or E_{f_i} in the mth session. For untraceable privacy notions to be meaningful, a Test session m must be fresh according to a freshness in Definition 2.

Definition 1 (Partnership and session completion of Fog nodes: FN_k, $Miner_n$ *and* $Fognode_{aggregator}$*):* A miner node instance $Miner_n$ and fog node instance FN_k in layer 2 are partners if, and only if, both have mutually authenticated each other with output Accept ($Miner_n$) and Accept (FN_k), respectively. Similarly, an instance of a fog node FN_k and aggregator node $Fognode_{aggregator}$ will do the same procedure for mutual authentication of partnership. The Instances of FN_k, $Miner_n$ *and* $Fognode_{aggregator}$ nodes then signify the completion of a partnership and mth session protocol to perform data block E_{k_i} transmission, storage, and aggregation.

Definition 2 (Session Freshness): An instance of a fog node FN_k, miner node $Miner_n$ and aggregator node $Fognode_{aggregator}$ is fresh at the end of mth session execution if, and only if (i) The FN_k or $Miner_n$ or $Fognode_{aggregator}$ node has output *Accept* with or without a partner node instance and (ii) both the instances including partner instance (if partner node exists) have not been sent a Corrupt query.

Definition 3 (Indistinguishability-based untraceable Privacy (INDPriv)): It is defined by a game G played between fog nodes instance (FN_k, $Fognode_{aggregator}$) and an adversary A. A runs the G with the setting as follows:

- Learning phase: For accessing E_{k_i} or E_{f_i}, an adversary A runs Execute and Send queries to interact with randomly chosen FN_0 and FN_1 from FN_k or $Fognode_{aggregator}$.
- Challenge phase: A selects two fog nodes (FN_0 and FN_1) and then forwards a Test query (FN_0, FN_1, m) to a challenger Cr. A Cr selects bit b {0, 1} for A and then using Execute and Send queries, A determines the fog node $FN_b \in$ (FN_0 and FN_1) which holds E_{k_i} or E_{f_i}. After determining FN_b, A runs reverse decryption to obtain blocks E_{f_i} or E_{k_i} and forward blocks using the execute query.
- Guess phase: An adversary A finishes the G and provides b' to be a guess of $b \in$ {0, 1} and $E_{k_i}{}'$ or $E'_{f_i} \in FN_{b'}$. For A, the security breach of INDPriv and success in a game is evaluated based on an A advantage to decrypt correctly E_{k_i} or decrypt and $aggregate\ E_{f_i}$ correctly from FN_0 or FN_1, which is denoted by $Advantage_A^{INDPriv}(d) = \left|\Pr\left[\mathrm{b} = b'\right] - \frac{1}{2}\right|$ or we can say the advantage is $Advantage_A^{INDPriv}(d) = \left|\Pr\left[E_{k_i} = E'_{k_i}\right] - \frac{1}{2}\right|$ $or\ Advantage_A^{INDPriv}(d) = \left|\Pr\left[E_{\mathrm{f}} = E'_{f_i}\right] - \frac{1}{2}\right|$, $where\ d\ represents\ a\ security\ parameter$.

Proposition: Divide and Conquer scheme satisfies INDPriv.

Proof: In the divide and conquer scheme, data M received from sensor nodes at $Miner_n$ is encrypted using AES, and then encrypted data is divided into blocks using tabular and linear distribution. The blocks are distributed to FN_k and the knowledge of previous and next block hashes is not known to FN_k. Further, the keys and unique identification numbers for M decryption are also not provided to FN_k and $Fognode_{aggregator}$. Besides this, tokenID for FN_k node authentication is updated in each session. Therefore, performing a traceability attack for an A using the following phases is difficult:

- Learning phase: A runs an Execute query ($Miner_n, FN_0, s$) in s-round during mth session and

obtains data block $\{E_{k_i,s}{}^{FN_0}\}$ at $FN_{o,s}$ node. Or A runs an Execute query $(FN_0, Fognode_{aggregator}, s)$ in s-round and obtains data block $\{E_{f_i,s}{}^{FN_0}\}$ at $\{Fognode_{aggregator}, s\}$ node.

- Challenge phase: A selects two fog nodes (FN_0 and FN_1) and send a Test query (FN_0, FN_1, $s+1$). Then according to a randomly chosen bit b {0, 1}, A is given a fog node $FN_b \in$ (FN_0 and FN_1). Next, $either\ A$ sends an Execute query ($Miner_n$, $FN_b, s+1$) and obtains data block $\{E_{k_i,s+1}{}^{FN_b}\}$. Or A sends an Execute query ($FN_k, Fognode_{aggregator(b)}, s+1$) and obtains data block $\{E_{f_{i,s+1}}{}^{FN_b}\}$. Then A runs reverse hash decryption to identify the message inside block $\{E_{k_i,s+1}{}^{FN_b}\}$. Similarly, for aggregation and decryption of block $\{E_{f_{i,s+1}}{}^{FN_b}\}$, A executes hash decryption to identify aggregated previous block B_{i-1}. Based on the pseudorandom key generator, A applies decryption of AES to $Decrypt\{E_{f_{i,s+1}}\}$.
- Guess phase: In a learning phase, the fog node (FN_0) does not know the hash of the previous block B_{i-1} as $B_{i-1} = hash(E_k)$ and the security keys (k) to decipher $\{E_{k_i}$ or $E_{f_i}\}$. As the private values including security keys (k), a unique ID of a data packet E_n and previous block hash B_{i-1} are kept at $Miner_n$ and not provided to fog node (FN_0). Only the hash of these private values $Hash_p(k, ID, H)$, where p is the security key of the end-user device, is forward to the end-user device. In the case of fog node (FN_0) acting as an aggregator node holding $\{E_{f_i}\}$ then $Miner_n$ only sends (B_{i-1}) to FN_0, where l is a security key of $Fognode_{aggregator}$ Therefore, in two subsequent rounds of learning and challenge phase $s\ and\ s+1$ of mth session, B_{i-1} and k are calculated as $b_{i-1,s}^{FN_0}$= $hash(E_{k_{i-1},s}{}^{FN_0})$ and $b_{i-1,s+1}^{FN_b}$= $hash(E_{k_{i-1},s+1}{}^{FN_b})$, $k_s^{FN_0}\ and\ k_{s+1}^{FN_b}$ using the Rijndael key schedule. Then $E_{k_i,s}{}^{FN_0}$ is computed as $E_{k_i,s}{}^{FN_0} = (E_{i,s}) + b_{i-1,s}^{FN_0}$, $E_{k_i,s+1}{}^{FN_b} = (E_{i,s+1}) + b_{i-1,s+1}^{FN_b}$. $E_{f_i,s}{}^{FN_0}$ is computed as $E_{f_i,s}{}^{FN_0} = Decrypt_{k_s^{FN_0}}(AES(E_{f_i,s}{}^{FN_0}))$. And $E_{f_i,s+1}{}^{FN_b}$=$Decrypt_{k_{s+1}^{FN_b}}(AES(E_{f_i,s+1}{}^{FN_b}))\ respectively$. Since $b_{i-1,s}^{FN_0} \neq b_{i-1,s+1}^{FN_b}$,$k_s^{FN_0} \neq k_{s+1}^{FN_b}$, $E_{k_i,s}{}^{FN_0} \neq E_{k_i,s+1}{}^{FN_b}$, $E_{f_i,s}{}^{FN_0} \neq E_{f_i,s+1}{}^{FN_b}$ and $hash(.)$ is an ε secure pseudorandom function, and AES is a secure block cipher. Thus A needs to make a random guess. Therefore, the advantage of A at correctly guessing E_{k_i} or E_{f_i} at $FN_0\ or Fognode_{aggregator}$ can be represented by:

$$Advantage_A^{INDPriv}(d) = \left| Pr\left[E_{k_i} = E'_{k_i}\right] - \frac{1}{2} \leq \varepsilon \right|$$

$$or\ Advantage_A^{INDPriv}(d) = \left| Pr\left[E_{f_i} = E'_{f_i}\right] - \frac{1}{2} \leq \varepsilon \right|$$

Also, an adversary A has to eavesdrop on maximum fog nodes in networks to find out maximum subsets of E_n as $(E_{0_0} + \cdots E_{k_i} + \cdots + E_{i_i}) = E_n$ and aggregate them in the right manner to acquire E_n. Further, without the knowledge of previous block hashes B_{i-1} kept in $table_{policy}$ at $Miner_n$, A cannot aggregate data blocks correctly together. Thus, A cannot process, aggregate, or decrypt E_{k_i} without acquiring the rest of the blocks kept at different fog nodes FN_k. Also, the probability of compromised block E_{k_i} leading to prefix and suffix of the whole encrypted message E_n blocks are close to zero as block E_{k_i} only contains a subset data of E_n and previous block hash value B_{i-1}, which cannot reveal the whole E_n. Furthermore, the higher the LoP settings, as shown in Table 7, the chances of data privacy violations are low. Moreover, aggregating blocks E_{f_i} is not possible without the knowledge of the previous block hash value $\boldsymbol{B_{i-1}}$, the hash keys, and unique identification number ID.

Additionally, all the blocks are only aggregated $(E_{0_0} + \cdots E_{k_i} + \cdots + E_{i_i}) = E_n$ and decrypted at the end-user/cloud device. No single fog node aggregator $Fognode_{aggregator}$ is aggregating the whole sensor data alone in a fog layer. Distributive aggregation processing is carried out to minimize the single point of failure risks. In the case of a single aggregator node being compromised or being exposed to a Denial of Service (DoS) attack, then another fog node in a k-hop cluster with computational resources availability will act as an aggregator and requests fog nodes to forward data blocks for aggregation.

4.3. Performance Analysis

We evaluate the performance of our scheme in terms of computational, memory, and communication overhead in different security parameters with the ECBDA [22], Masker [131], APPA [7], and LVPDA [272] schemes. ECBDA, Masker, APPA, and LVPDA are implemented based on the proposed architectures to analyze the computational, memory, and communication costs during data processing and storage. In the ECBDA [22] scheme, the sensor data is secured at miner nodes (acting as a gateway for the scheme) using elliptic curve-based ElGamal encryption with bilinear homomorphic mapping. Trusted Third Party (TTP) is also involved in the generation and distribution of keying material to miner nodes for encryption and end-user device for decryption. TTP is online during key initialization and set offline during the aggregation process. Encrypted data from different miner nodes are aggregated on a single aggregator and forwarded to the end-user device.

For the Masker [131] scheme, masking values are generated at the miner node for each data set received from sensor nodes. Masking values are implemented using a symmetric encryption algorithm with a sequence number for each data set. Further, the sequence number is incremented after an iteration of each masking value. In the data packet, masking values are

appended with sensor data reading and a digital signature of the miner node. A resultant packet is forwarded to aggregators which sum-up packets and then end-user device sum-up reading by removing masked values from data packets.

In the APPA scheme [7], sensor nodes authenticate and register themselves to fog nodes using Local Certification Authority (LCA) and Trusted Certification Authority (TCA). Each fog cluster owns LCA for authenticating new sensor nodes using pseudonyms certificates. TCA is involved in the generation and distribution of security keys and registration management. The identity of the sensor node is anonymized using Rivest, Shamir & Adleman's (RSA) zero-knowledge signature. Then sensor nodes are authenticated to miner nodes in the fog layer using pseudonym certificates. Data received from an authentic sensor node is encrypted at the miner node using Paillier public-key cryptosystem. Encrypted data from different miner nodes are aggregated at a single fog node with pseudonym certificates. The fog node computes the final aggregation using its own pseudonym certificates, and then end-user/cloud device decrypts aggregated data using a private key with Paillier decryption.

The LVPDA scheme [272] is based on online/offline signature, bilinear pairing, and Paillier homomorphic cryptosystem. TTP generates and distributes private and public keys using homomorphic cryptosystem and bilinear settings. Sensor nodes authenticate themselves to miner nodes by performing an offline-signature authentication process. Miner node encrypts the sensor node's data with the Paillier encryption. Furthermore, transmit the encrypted data to the aggregator fog node using online-signature verification. The aggregator node aggregates data and generates an aggregation signature for verification of authenticity to the end-user device. Aggregated data with aggregation signature is forward to the end-user device. Aggregated data is verified and decrypted with a key provided by TTP.

4.3.1. Computational Cost

Figure 13 and Table 9 show the computational overhead comparison of the five schemes. From Figure and Table, we can learn the facts as follows.

Table 9 Average Computational overhead (in milliseconds) based on security parameters

Security Parameter	Masker [19]	ECBDA [10]	APPA [7]	LVPDA [272]	Divide-and-Conquer scheme
128	2.16	3.2	4.4	4.48	1.5
192	5.0	7.8	8.56	8.68	3.2
256	9.5	11	12.5	13	6.2

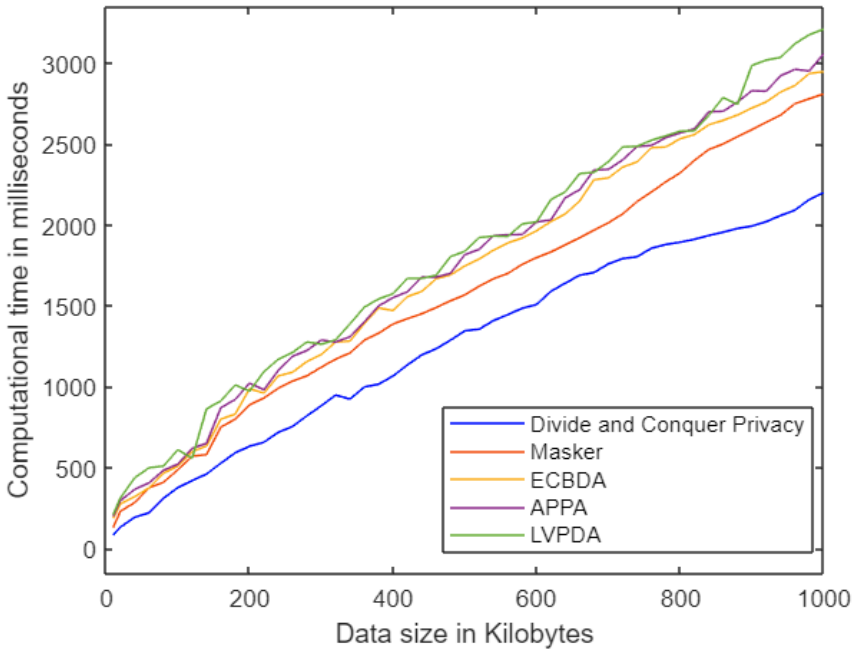


Figure 13 Computational overhead of scheme vs. Masker, ECBDA, APPA & LVPDA schemes

1. First, computation overhead for processing data during encryption and aggregation is reasonably low in our scheme as compared to the ECBDA, APPA, and LVPDA schemes. Our scheme encryption mechanism is based on the symmetric cipher AES Algorithm, which requires less time and is faster than the ECBDA, APPA, and LVPDA schemes. In the ECBDA, Elliptic Curve Cryptography (ECC) asymmetric cipher is used, which is based on shared secret and signature generation protocols (private and public keys). The use of asymmetric cipher slower the processing speeds and introduces the network complexity. Similarly, the APPA and LVPDA schemes are also based on the public-key cryptosystem. The schemes require addition and multiplication of asymmetric ciphertext with great common, least common divisors and modular multiplicative inverse. The use of the asymmetric Paillier cryptosystem slows encryption processing and increases network complexity. In the LVPDA scheme, online signature verification and generation at the aggregator node increase the overall computational cost compared to the APPA scheme.
2. Further, aggregation in the ECBDA, APPA, and LVPDA schemes are performed at a single aggregator node, which increases the computational burden of aggregation on a single aggregator node as compared to our scheme. In the case of the Masker scheme, the AES Algorithm is also used in the Counter mode Deterministic random byte generator (CTR_DRGB) with a sequence number and non-secret value. Further, masking is applied for data generation, which increases more computational overhead as compared to the basic AES Algorithm. With the data size increase, the computational overhead is becoming notably high as compared to our scheme, as shown in Figure 13.
3. Varying security parameters, i.e. 128 bits, 192 bits, and 256 bits to encrypt data also impact

the computational overhead, as shown in Table 9. The Table also clearly indicates that our scheme outperforms ECBDA, APPA, LVPDA, and Masker scheme. Based on security parameter 128, the miner nodes of the ECBDA scheme spend approximately 3.28 milliseconds on average for performing encryption, about 4.4 milliseconds for APPA, 4.43 milliseconds for LVPDA, and 2.16 milliseconds for Masker. However, the miner nodes in our scheme perform data encryption using less than 1.5 milliseconds.

Consequently, for the security parameter 192, APPA and LVPDA utilize more than 8.5 milliseconds for data encryption as compared to ECBDA using more than 7 milliseconds and 5 milliseconds for Masker. Whereas our scheme is utilizing approximately 3 milliseconds for encryption and the rest of the time is utilized for aggregation processing. Similarly, for security parameter 256, our scheme requires less than 7 milliseconds to perform encryption. However, APPA, LVPDA, ECBDA, and Masker require 12.5 milliseconds, 13 milliseconds, 11 milliseconds, and 9.5 milliseconds remotely. Due to increasing computation overhead in the APPA and LVPDA schemes, all the incoming data packets from the sensor nodes have to wait in a queue to get allocated to processing in a fog layer for approximately 3 seconds every 5 minutes. For ECBDA, a delay is around 2.5 seconds every 5 minutes. And about 1.5 seconds delay every 5 minutes for the Masker scheme, whereas our scheme delay is less than one second for every 5 minutes.

4.3.2. Memory Cost

A memory overhead comparison of our scheme with ECBDA, Masker, APPA, and LVPDA, as shown in Figure 14. Memory cost is the overhead due to the increase in the data size after data processing. From Figure 14, it can be deduced that LVPDA incurs high memory overhead followed closely by APPA. Both the schemes are based on Paillier Cryptosystems and require signature/ certificate verification for data aggregation, which incurs high memory space as compared to ECBDA, Masker, and our schemes.

The variation in the data size of our scheme, Masker, ECBDA, APPA, and LVPDA schemes based on Figure 14 is shown in Table 10. In our scheme, the data size increase is moderate throughout the processing due to lightweight encryption, distributed storage, and aggregation. An increase in total data size after the processing of original data ranging from 100 KBs to 500 KBs is approximately between 20 KBs to 100 KBs. Whereas, the increase in the ECBDA, Masker, APPA, and LVPDA schemes is around 40 KBs to 500KBs, respectively. For data ranging from 600 KBs to 1000 KBs, the increase in data size after processing in our scheme is approximately between 100 KBs to 300 KBs. However, for the ECBDA scheme increase is roughly between 210 KBs to 1000 KBs, for the Masker scheme around 170 KBs to 700 KBs, data size increase in the APPA scheme is around 600 KBs to 2400 KBs and for LVPDA it is between 700 KBs and 2600 KBs.

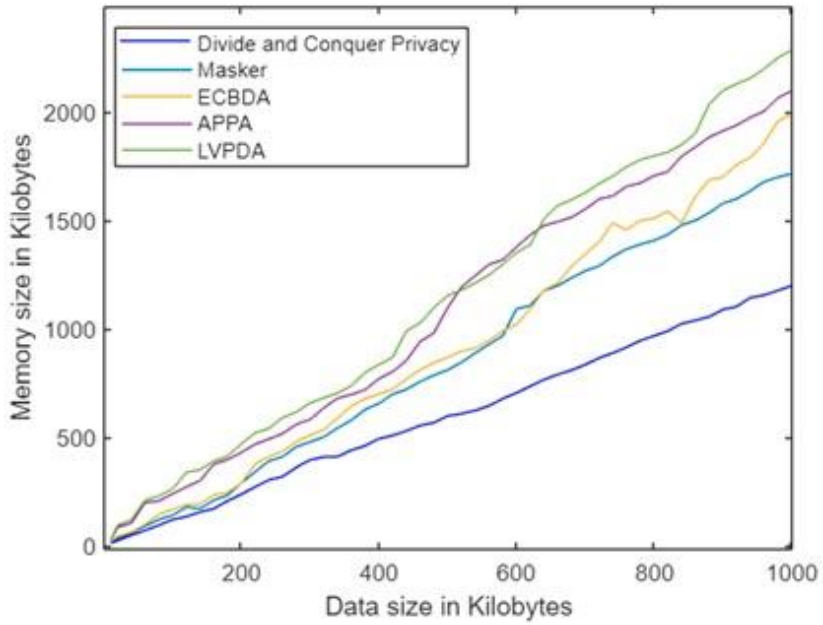


Figure 14 Memory size of scheme vs the ECBDA, Masker, APPA & LVPDA Schemes

Table 10 Data size variation results from Figure. 14

Original Packet data size (KBs)	Data size (K.B.s) in Masker scheme	Data size (KBs) in the ECBDA scheme	Data size (KBs) in the Divide-and-Conquer scheme	Data size (KBs) in the APPA scheme	Data size (KBs) in the LVPDA scheme
100	140	167	120	208.5	230
200	269	288	237	329.5	369
300	478	510	395	551.8	558
400	655	698	493	739.5	755
500	810	865	598	1017	1033
700	1267	1345	834	1571	1627
1000	1715	1988	1198	2417	2689

Overall, the LVPDA scheme incurs a larger memory overhead as compared with the APPA, ECBDA, Masker, and our scheme. An increase in data size is not only because of the heavyweight cryptosystem for encryption, but the increase is also due to the generation of aggregated data signature for verification and then appending signature with the aggregated data for end-user devices to verify and decrypt aggregated data. Thus, signature addition to aggregated data increases the overall data size. Further, memory overhead introduced by asymmetric APPA and ECBDA is also larger than the proposed scheme. Also, the notable memory overhead introduced by the Masker scheme as compared to our scheme is due to the masking generation and signature bits. In our scheme, no single aggregator is used to aggregate data, a minimum of two aggregator nodes are utilized for 1KB to 5 KBs of data.

With an increasing number of sensor nodes and data size, aggregator nodes in fog layer 2 vary as discussed in section 3. All the aggregator nodes send the resultant data to the end-user/cloud device which sum up all the nodes' data together. Whereas, the data summation is carried out at a single aggregator node in the LVPDA, APPA, ECBDA, and Masker schemes. Due to a single aggregator node performing aggregation processing, memory overhead increases, and the probability of being exposed to DoS attacks is higher as compared to our scheme.

It can be deduced that the LVPDA, APPA, ECBDA, and Masker schemes overhead during data processing at fog node and aggregation at a single node is significantly higher than the distributed aggregation in our scheme. Because of the distributed aggregation workload, all the aggregator nodes aggregate data simultaneously, which minimizes the overall processing and memory overhead on a single node.

Consider 100 KBs of data from three sensor devices are transmitted after processing to an aggregator. In the case of the LVPDA scheme, data is encrypted using Paillier encryption and appended with a signature, so the data size becomes 230 KBs, as shown in Figure 14. Therefore approximately 230 KBs of data aggregated from three different sensor devices at a single aggregator will result in 690 KBs memory consumption. In the APPA scheme, data is also encrypted using the Paillier encryption, and the total memory consumption at an aggregator node becomes 624 KBs. Data is encrypted using ECC in ECBDA, and data size becomes approximately 500 KBs at a single aggregator node. The Masker scheme consumes memory of 400 KBs at an aggregator node. In contrast, our scheme consumes about 180 KBs of memory at a minimum of two aggregator nodes. All aggregator nodes separately send sum-up data to the end-user device for summation of all aggregator nodes data and then decryption.

4.3.3. Communication Cost

The communication overhead of our scheme is significantly higher than the LVPDA, APPA, ECBDA, and Masker schemes, as shown in Figure 15. It is due to the distributive data processing, storage, and aggregation, which involves communication between a miner and fog nodes. Distribution of data between fog nodes and then aggregating data at multiple aggregator nodes. Before applying the proposed encryption, Divide-and-Conquer, and aggregation Algorithms, network decomposition also adds up communication overhead. However, the Masker, ECBDA, APPA, and LVPDA schemes only add up communication overhead during authentication and key exchange from a TTP, data aggregation on single nodes, and data transmission process. Due to the addition of local certified authority for authentication along with TTP, the communication overhead of the APPA scheme is slightly higher than Masker, ECBDA, and LVPDA.

Figure 15 shows communication overhead for all schemes. The communication overhead increases with an increase in data size. In our scheme, miner nodes and end-user devices communicate with each other to request, process, store, or distribute data. Whereas for the LVPDA, ECBDA, and Masker schemes, sensor nodes communicate with only a miner node, a trusted authority, and a single aggregator node. Our scheme avoids a single point of failure at the cost of a slight increase in communication overhead but reduces the computation and memory cost.

Although the communication overhead of our scheme is greater as compared to the APPA, LVPDA, ECBDA, and Masker schemes, still our scheme markedly reduces the performance overhead in terms of computational and memory cost. The reduction in computational and memory cost is due to the distributive processing of data at miner nodes, storing at fog nodes, and aggregating at multiple aggregator nodes. Also, our scheme provides high data privacy with distributive computation and aggregation as compared to single node computation and aggregation in the APPA, LVPDA, ECBDA, and Masker schemes.

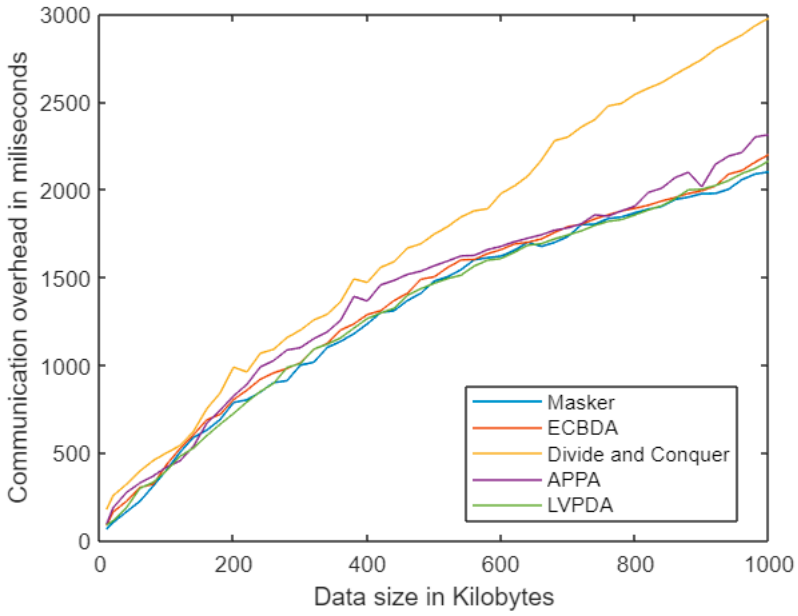


Figure 15 Communication overhead of our scheme vs. the ECBDA, Masker, APPA, LVPDA schemes

5. Conclusion

This paper presents a lightweight privacy-preserving aggregation scheme in fog computing. The scheme is based on a Divide-and-Conquer approach to improve data privacy during data aggregation as compared to the traditional schemes. First, we presented a system model and an adversary model for preserving privacy in the fog layer. Then we proposed a Divide-and-Conquer privacy-preserving data aggregation scheme with data-owner defined level of privacy settings.

The proposed privacy scheme requires data encryption, which mainly is lightweight symmetric cryptographic operations, and encrypted data division is based on the data-owner predefined LoPs' settings. Regarding our privacy analysis, we proved that for an attacker, it is not possible to be successful in internal and external attacks. Also, the performance evaluation showed that our scheme minimizes the computational and memory overhead in comparison with traditional schemes.

For future work, we will improve our scheme's performance efficiency in terms of communication overhead. Further, enhance the data distribution Algorithm to make it suitable for the larger data size. The main purpose of our scheme is to preserve the privacy within the fog layer, the privacy preservation at sensors and end-user/cloud is out of the scope. Therefore, in the future, we also aim to improve our scheme to preserve end-to-end privacy.

'True optimization is the revolutionary contribution of modern research to decision processes'.

--- George Dantzig

Chapter 4: Joint Optimization of Time and Energy Consumption for Data Aggregation in Fog-enabled IoT Networks

Abstract

Fog computing is an emerging concept for providing networking, computing, and storage capabilities that can support the Internet of Things (IoT). IoT devices can offload computational tasks to fog nodes within their proximity instead of a remote cloud. Offloading tasks including data aggregation can reduce data redundancy while improving data analysis's speed and data storage at the edge of an IoT network and data aggregation tasks can be performed in a distributive manner. Although by offloading aggregation tasks to fog, network overhead and energy consumption of IoT devices can be reduced, it may incur a large time consumption including execution, transmission, and waiting time to aggregate data at fog nodes. Therefore, fog computing poses a challenge to optimize the time consumption with the energy consumption of data aggregation. To address this challenge, first, we formulate a multi-objective optimization problem with a joint objective to optimize time consumption and energy consumption for data aggregation in fog computing. Second, we define the multi-objective optimization method based on the NSGA-III (non-dominated sorting genetic algorithm III) to find optimal solutions concerning both time consumption and energy consumption. Finally, we conduct comprehensive simulations and systematic experiments to demonstrate and evaluate the efficiency of our method compared with the state-of-the-art methods.

This contribution has been submitted to IEEE Internet of Things Journal

1. Introduction

The Internet of Things (IoT) has merged the interconnectivity of smart things and objects, such as wearable devices and mobile devices [7, 274]. Mobile devices can be connected to IoT using wireless networks, which integrate IoT with mobile sensing and computing capabilities [59] and IoT devices can sense mobile users in the surroundings and generate real-time data for processing [275]. Further, IoT devices may take some responsibility in the aggregation and analysis on generated data. This responsibility includes the coordination of aggregated data and analysis actions to make a decision on the utilization of data during these actions [276].

Due to the limitation of IoT devices' battery and computational resources, cloud computing provides enormous data processing, including aggregation, analysis, and storage capabilities [277]. The data generated by IoT devices can be transmitted to the cloud for processing, which is known as task offloading [278].

Task offloading to the cloud can increase IoT network bandwidth and latency overhead, therefore, the concept of fog computing has been introduced to provide processing and storage capabilities closer to IoT devices [279]. Instead of using cloud, task offloading on fog nodes can avoid an increase in latency time and network congestion.

In addition, the process of data aggregation in fog computing reduces the data redundancy while improving data analysis speed and data storage [280]. The offloading tasks including data aggregation and storage can be performed in a distributive manner in fog computing [42, 184, 281] which requires extensive time and energy consumption. Hence, arbitrarily offloading tasks to fog nodes hardly reduce computation time used by the nodes and it is essential to efficiently compute distributed tasks while minimizing the time consumption and energy consumption required. However, this problem is an NP-hard problem as the difficulty of reducing time consumption and energy consumption increases exponentially with the increase in the number of offloading tasks, sensor nodes, and fog nodes.

It remains a challenge to optimize the multi-objective including time and energy consumption in fog computing for IoT applications. To address this challenge, we propose a multi-objective optimization method based on a non-dominated sorting genetic algorithm (NSGA-III). The main contributions of this paper are summarized as follows.

1. We analyze and formulate time consumption and energy consumption for data aggregation in fog-enabled IoT.
2. We propose a multi-objective optimization method (MUOM) based on NSGA-III (non-dominated sorting genetic algorithm III) to reduce time consumption and energy consumption on each fog node.
3. We comprehensively evaluate the efficiency of our method compared with state-of-the-art optimization methods.

The remainder of the paper is organized as follows. We discuss related work in Section 2. Section 3 outlines our proposed system model and problem formulation. In Section 4, we present our multi-objective optimization method for fog-enabled IoT followed by the evaluation of the proposed method in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Computation tasks in IoT devices can be offloaded to remote cloud/servers for processing due to low computational and power capacity of the devices [282, 283], however by doing so there are challenges in managing network overhead, increased communication and computational energy and cost. Compared with the cloud, fog computing can reduce the network overhead and congestion by alleviating the workload from a remote cloud to the edge of a network close to IoT devices [66].

Several works have proposed different approaches to offloading tasks from IoT devices to fog nodes [284-292]. In [284], Yousefpour *et al.* proposed a framework for task offloading to reduce the service delay of IoT-cloud applications in fog computing. The framework provides a minimizing policy for the service delay and the policy considers the service delay based on the load of each fog node. If the service delay is greater than the threshold value, then the offloading task is transmitted to the best neighboring fog node. Otherwise, the task is accepted at the same fog node for processing. In their framework, fog nodes are interconnected in a distributive manner, which leads to the data transmission overhead. Therefore, the minimizing policy is not able to optimize transmission overhead that may incur. Also, the optimization of energy consumption for service delay is not considered in the proposed framework. Similarly, Yousefpour *et al.* in [285, 286] do not consider the transmission and energy overhead of fog nodes for service delay optimization in fog computing.

Another study focused on the computation of the IoT tasks partially at fog computing [287]. For a further computation of tasks, IoT data can be forwarded from fog nodes to the cloud and offloading tasks to the cloud can reduce the workload and power consumption of the fog nodes that are needed to compute heavyweight tasks. However, the transmission of tasks to the cloud increases network overhead and computational and communication costs. Further, the study assumed that the cloud is connected to fog nodes using a single communication point. In this context, the communication network between fog nodes and the cloud are vulnerable to a single point of failure threat.

In [289], Jiang *et al.* proposed a meta-heuristic method, which investigates the placement of tasks offloaded to the fog nodes. Based on the meta-heuristic method's cost function, the study considered communication cost, computation cost, and power consumption

of fog nodes. In this method, a technique for priority mapping is used to schedule the placement of tasks to help reduce the power consumption, communication, and computational cost of fog nodes during the placement. However, the optimal solutions for optimizing the energy and power consumption of fog nodes for the tasks, including data processing and data storage are not provided.

Liu *et al.* proposed a method [288] for optimizing energy consumption, execution time delay, and payment cost in fog computing. Their method transforms a multi-objective problem into a single-objective problem using scalarization and interior point techniques. These techniques are intuitively not satisfying as they do not visit vertices of a problem but only cover the interior region of the problem. The techniques can find an optimal solution from an interior region without considering a problem's vertices.

Naqvi *et al.* [290] proposed a meta-heuristic method based on the ant colony optimization (ACO) method to optimize response times of smart grid applications in fog computing. This study did not consider the optimization of transmission time and transmission energy. Further, the ACO method depends on profiling offloaded tasks, which incurs high transmission overhead. Hussein *et al.* in [292] further enhanced the ACO method to optimize the transmission time of offloaded tasks at fog nodes. Still, the new ACO method is based on a single-objective optimization on the transmission time. Also, the proposed method offloads the aggregation tasks to the cloud for further processing and storage, which results in more network overhead.

In [291], Binh *et al.* proposed a method based on a genetic algorithm (GA) for offloading tasks and scheduling tasks at fog nodes. The main objective of the method is to achieve a trade-off between offloading tasks, scheduling tasks and monetary cost to efficiently complete tasks in fog and cloud system. The proposed GA achieved high cost and performance efficiency for offloading tasks to fog nodes.

However, none of these methods [284-292] considered multi-objective problems concerning time consumption and energy consumption for computing tasks such as data aggregation in fog computing. In conclusion, there is still key challenge in finding optimal solutions concerning time consumption (including transmission, execution, and waiting) and energy consumption for efficient task offloading in fog computing. To address this challenge in this paper, we propose a new multi-objective optimization method (MUOM) in a fog computing for IoT applications.

3. System Model and Problem Formulation

In this section, we discuss our proposed system model and problem formulation of the optimization time consumption and energy consumption for data aggregation in fog computing. Data aggregation formulation regarding time and energy consumption are based on the divide-and-conquer scheme for data aggregation proposed in [42]. Our notations used in this paper with their descriptions are listed in Table 11.

Table 11 Key notations and description

Notation	Description
$\mathbf{fn_{l,j}}$	The computing task of l^{th} fog node
$\mathbf{T(fn_{l,j})}$	The total time consumption of $\mathrm{l^{th}}$fog node
$\mathbf{SN_k}$	Kth sensor node
$\mathbf{FN}$	Fog Node
$\boldsymbol{t_i}$	Ith data-owner
$\mathbf{T_{exe}}$	The execution time
$\mathbf{T_{trans}}$	The transmission time
$\mathbf{T_{wait}}$	The waiting time
$\mathbf{w_{l,j}}$	Workload of $\mathrm{l^{th}}$fog node
$\mathbf{T(fn^1_{l,j})}$	The total time consumption of the $\mathrm{l^{th}}$ miner node in fog layer 1
$\mathbf{T(fn^2_{l,j})}$	The total time consumption of the $\mathrm{l^{th}}$ fog node in fog layer 2
$\mathbf{E(fn_{l,j})}$	The total energy consumption of $\mathrm{l^{th}}$fog node
$\mathbf{E_{trans}}$	The energy consumption of transmission
$\mathbf{E_{exe}}$	The energy consumption of execution
$\mathbf{E_{wait}}$	The energy consumption of waiting for execution of the precursor executing task
$\mathbf{E(fn^1_{l,j})}$	The total energy consumption of the $\mathrm{l^{th}}$ miner node in fog layer 1
$\mathbf{E(fn^2_{l,j})}$	The total energy consumption of the $\mathrm{l^{th}}$ fog node in fog layer 2

3.1. System Model

Our system model is illustrated in Figure 16, which is based on the model presented in [42]. Figure 16 consists of fog nodes, data-owner, sensor nodes, and end-user devices. In this model, a fog computing layer is divided into two sub-layers: Fog layer 1 and Fog layer 2. Fog layer 1 comprises fog nodes for computation and analysis of IoT tasks. In fog layer 2, fog nodes are organized in a cluster and they are responsible for aggregation and storage tasks.

IoT devices, i.e. sensor nodes, can be connected to fog layer 1 through a local area network (LAN) connection and both layers (fog layer 1 and fog layer 2) are interconnected through a LAN connection. End-user devices and the data-owner are connected to fog layer 2 and fog layer 1, respectively, through a wide area network (WAN) connection.

Let $SN_k = \{sensor_1, \dots, sensor_k\}\ (1 \le k \le max(sensor))$ represents a set of sensor nodes for generating IoT data and a data-owner can be denoted as $DT = \{t_1, \dots, t_i\}(1 \le i \le max(dataowner))$. A data-owner defines the data utilization and privacy policies for their generated data at sensor nodes.

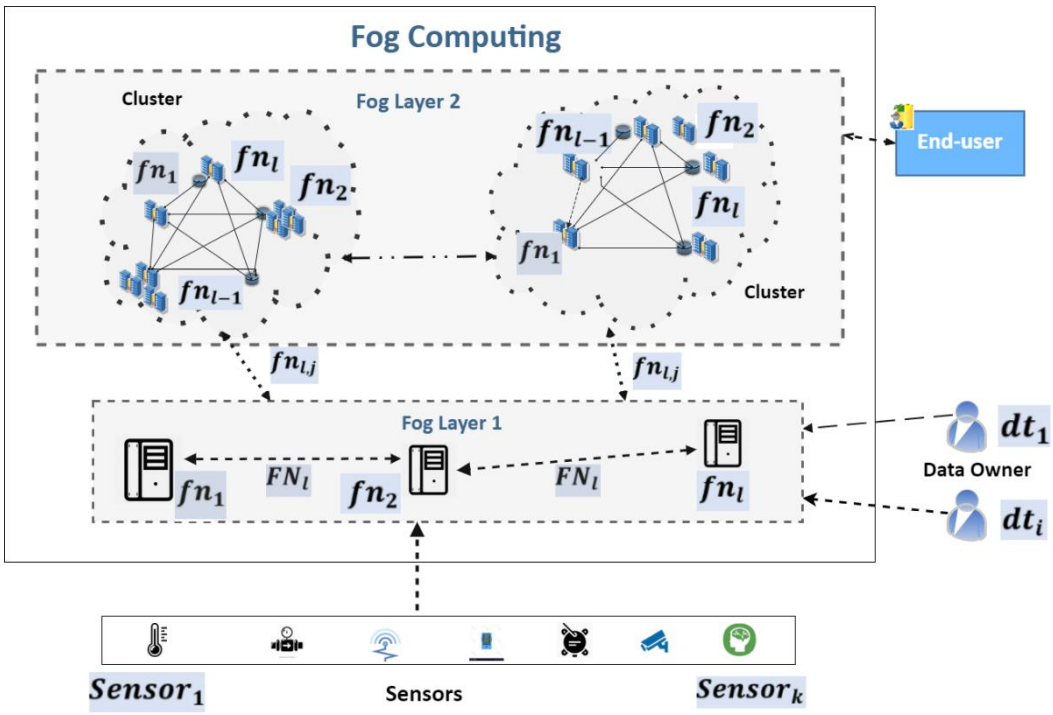


Figure 16 System Model

We also use FN to represent a set of for fog nodes in both layers and it can be denoted as $FN = \{fn_1, fn_2, \ldots., fn_l\}(1 \leq l \leq N)$, where $FN_l = \{fn_{l,j} | 1 \leq j \leq FN_l|\}$ represents a set of computing tasks in the l^{th} fog node. Let $w_{l,j}$ be a workload of the l^{th} fog node. Also, $pre\left(fn_{l,j}\right)$ indicates the precursory computing task, which is waiting in a queue for $fn_{l,j}$.

Based on the network design and setup presented in [42], we consider a scenario where SN_k sends data to FN in fog layer 1. FN has to perform FN_l which is a set of computation tasks including data authentication, data analysis, data encryption, scheduling transmission of channel and $fn_{l,j}$ allocation to FN in fog layer 2. According to the t_i policies, FN_l is processed, such as encrypted, analyzed, and authenticated at fog layer 1. Fog layer 2 receives processed $fn_{l,j}$ from Fog layer 1. Then FN in fog layer 2, stores and aggregates processed $fn_{l,j}$. On a request of the end-user device, FN in fog layer 2 sends aggregated data in $fn_{l,j}$ to the end-user device.

3.2. Time Consumption Model

In our proposed model, time consumption $T(fn_{l,j})$ is an amount of time consumed by fog nodes to perform computing tasks $fn_{l,j}$ for data aggregation. $T(fn_{l,j})$ consists of the transmission time T_{trans}, the execution time T_{exe} and the waiting time of the precursor executing task T_{wait}. The transmission time T_{trans} is an amount of time that each fog node in both fog layers takes to communicate with the nodes in fog layers, sensors, data-owner, and end-user devices. T_{trans} involves the communication time for $fn_{l,j}$ according to the tasks defined in [42], including data

request, data transmission, cluster formation, data authentication, public and private key, and table distribution.

The execution time T_{exe} is time taken by FN to executes the $fn_{l,j}$. T_{exe} involves $fn_{l,j}$ for executing data encryption, data authentication, creation of the public and private key, creation of the policy table, aggregation, and cluster formation. The precursor executing task T_{wait} is the waiting time that each computing task $fn_{l,j}$ must wait in a queue to be executed by FN. The total $T(fn_{l,j})$ based on [59, 293] becomes:

$$T(fn_{l,j}) = \sum_{FN_l \in FN} (T_{trans} + T_{exe} + T_{wait}) \tag{4.1}$$

The time consumption of both fog layers depends on the computing tasks $fn_{l,j}$, which are performed uniquely by each fog layer. Therefore, first, we compute the $T_{trans}, T_{exe}\ and\ T_{wait}$ for both layers separately and then combine them to get the total time consumption $T(fn_{l,j})$, respectively.

For fog layer 1, the transmission time T_{trans} can be calculated by

$$T^1_{trans} = \left(\sum_{fn_j \in fn_l} (fn_j) \right), Bandwidth_l, Num\ (FN) \tag{4.2}$$

In fog layer 1, the transmission time T^1_{trans} of our proposed model depends on the transmission tasks fn_j, the total number of fog nodes and network bandwidth of fog computing. The transmission tasks fn_j are the tasks for fog nodes' authentication, request for data policies, sending hash keys, requesting data, allocation of tasks, and scheduling channel. Each of these fn_j is discussed as follows.

In an authentication task $Auth_i$, the fog nodes send an authentication token to the other fog nodes in both layers with whom they wish to communicate. Authentication token is for checking the fog node's authenticity within a network. For policy fn_j, the fog nodes request the data-owner t_i for data policies P_{LoP}.

Also, fog nodes processing data tasks such as data encryption and division in fog layer 1 send a hash of a private keys P_k to the end-user device for decryption of processed data. In the requested fn_j, fog node requests data M from sensor nodes SN_k in close proximity and fog node receive data M from a sensor node SN_k. The fn_j for allocation, allocates the computing tasks to chosen fog nodes in fog layer 1. Another fn_j is the scheduling of transmission channels for fog computing network.

The transmission time T^1_{trans} also depends on the network bandwidth $Bandwidth_l$. The bandwidth between fog nodes in both layers, sensor nodes, and data-owner can be computed by $Bandwidth_l = \begin{cases} \infty, & FN = 0 \\ Bandwidth_{l,L}, & FN_l = 1, 2, \dots, N \\ Bandwidth_L, & SN_k = 1,2, \dots \max(sensor) \\ Bandwidth_{l,W}, & DT = 1, 2, \dots \max(data - owner) \end{cases}$

Let $Bandwidth_{l,L}$ be the bandwidth of a LAN network for the l^{th} fog node to the other fog nodes in both fog layers. $Bandwidth_L$ represents the bandwidth of a LAN network for the l^{th} fog nodes and sensor nodes SN_k, and $Bandwidth_{l,W}$ is WAN for the l^{th} fog node and the data-owner DT.

In the execution of a computing task in fog layer 1, the execution time T^1_{exe} of the l^{th} fog node is determined by the workload of the fog node and the computational capacity of the l^{th} fog node. Based on the formula [59], T^1_{exe} can be computed by

$$T^1_{exe} = \frac{\sum_{w_j \in fn_l}(w_j)}{C_{cap_l}} \tag{4.3}$$

The workload w_j at the l^{th} fog node consists of the following workloads for the computing task fn_j.

The w_j for encrypting data M at the fog node for processing data. Then the w_j for a division of encrypted M at the fog node. Also, the w_j is for generating and checking token-based authentication $auth_i$.

Further, the workload w_j for creating a hash of a private key p_k. The execution time T^1_{exe} also depends on the workload for creating a table with data-owner defined policies.

For fog layer 1, the precursor T^1_{wait} is a waiting time in a queue for the execution of fn_j at l^{th} fog node. The l^{th} fog node is represented as a tuple $\left(total\left(w_j\right), Num(FN_l)\right)$, where $total\left(w_j\right)$ represents the total workload at the l^{th} fog node and $Num(FN_l)$ represents the number of computing tasks that are scheduled to the l^{th} fog node.

$$T^1_{wait} = \sum_{j=1}^{total\,(w_j)} pre(T^1_{exe,j}) \tag{4.4}$$

By combining (4.2), (4.3), and (4.4), (4.1) becomes:

$$T(fn^1_{l,j}) = \left(\begin{array}{c} \left(\sum_{fn_j \in fn_l} (fn_j) \right) + \left(\frac{\sum_{w_j \in fn_l}(w_j)}{C_{cap_l}} \right) \\ + \left(\sum_{j=1}^{total\,(w_j)} pre(T^1_{exe,j}) \right) \end{array} \right), Bandwidth_l, Num\,(FN) \tag{4.5}$$

Now for fog layer 2, the transmission time T^2_{trans} taken by the l^{th} fog node is defined as:

$$T^2_{trans} = \left(\sum_{fn_j \in fn_l}(fn_j)\right), Bandwidth_l \;\; Num\;(FN), Num\;(C(FN)) \tag{4.6}$$

Similar to T^1_{trans}, the T^2_{trans} also depends on the transmission tasks fn_j, the network bandwidth $Bandwidth_l$, the total number of fog nodes $Num\;(FN)$ in fog layer 2, and the number of fog nodes in a cluster $Num\;(C(FN))$.

The fn_j for T^2_{trans} includes authentication, sending aggregated data, receiving data blocks, cluster formation, and channel scheduling. Each of the fn_j is discussed in detail below.

For the token-based authentication fn_j, fog nodes send a token $Auth_i$ to other fog nodes in both fog layers and end-user devices, with whom they wish to have data communication. Afterward, fog nodes communicate with neighboring fog nodes for cluster formation and scheduling of the transmission channel. Then fog nodes request and receive data blocks $block_i$ from fog layer 1. Fog nodes also request the table policy. Another fn_j is for the communication with the end-user device for sending aggregated data E. Further, the fog node requests the fog nodes in the same layer to send blocks $block_i$ for aggregation. Besides fn_j, the transmission time T^2_{trans} also depends on $Bandwidth_l$. The bandwidth between fog nodes in both layers and end-user devices. The $Bandwidth_l$ is measured as

$$Bandwidth_l = \begin{cases} \infty, & FN = 0 \\ Bandwidth_{l,L}, & FN_l = 1, 2, \dots, N \\ Bandwidth_{l,W}, user_i = 1, 2, \dots \max\,(end - user\;device) & \end{cases}$$

where $Bandwidth_{l,L}$ represents the bandwidth of a LAN for the l^{th} fog node in both fog layers. $Bandwidth_{l,W}$ represents the bandwidth of WAN for the l^{th} fog node and the end-user device.

Similar to fog layer 1, the execution time T^2_{exe} of the l^{th} fog node in fog layer 2 is determined by the workload of the l^{th} fog node and the computational capacity of the l^{th} fog node.

$$T^2_{exe} = \frac{\sum_{w_j \in fn_l}(w_j)}{C_{cap_l}} \tag{4.7}$$

The workload w_j at the l^{th} fog node consists of the following workloads for computing tasks fn_j in fog layer 2.

- One of the w_j is for generating and checking token-based authentication, like fog layer 1.
- Also, w_j is for processing the request of the data $block_i$.

- Further w_j involves the aggregation of encrypted M.

Similar to T^1_{wait}, the precursor waiting time T^2_{wait} is computed for fog layer 2. Thus, the total time consumption $T(fn^2_{l,j})$ for fog layer 2 becomes:

$$T(fn^2_{l,j}) = \begin{pmatrix} \left(\sum_{fn_j \in fn_l} (fn_j)\right) + \left(\frac{\sum_{w_j \in fn_l}(w_j)}{C_{cap_l}}\right) \\ + \left(\sum_{j=1}^{total\,(w_j)} pre(T^2_{exe,j})\right) \end{pmatrix}, Bandwidth_l Num\,(C(FN)), Num\,(FN) \quad (4.8)$$

From (5) and (8), the total time consumption $T(fn_{l,j})$ becomes:

$$T(fn_{l,j}) = \sum_{FN_l \in FN} (T(fn^1_{l,j}) + T(fn^2_{l,j})) \quad (4.9)$$

3.3. Energy Consumption Model

Energy consumption $E(fn_{l,j})$ of FN is the consumption of energy in the transmission E_{trans}, the execution E_{exe} and the waiting for execution E_{wait} of the precursor computing task $fn_{l,j}$ based on [59, 293].

Let E_{trans} be an energy consumption for transmission, which is a power consumed by each fog node in both fog layers to transmit data to the nodes in fog layers, sensors, data-owner, and end-user devices.

The energy consumption for execution E_{exe} represents a power consumed by each fog node to execute the computing tasks $fn_{l,j}$. The energy consumption for precursor execution E_{wait} is the energy consumption that each computing task $fn_{l,j}$ requires to wait in a queue to be executed by FN. Based on the time consumption in subsection 3.2, energy for fog layer 1 and fog layer 2 can be computed as

$$E(fn^1_{l,j}) = \begin{pmatrix} (T^1_{trans} * p_{trans}) + (T^1_{exe} * (p_a + p_i)) \\ +(pre(T^1_{exe} * (p_a + p_i))) \end{pmatrix} \quad (4.10)$$

$$E(fn^2_{l,j}) = \begin{pmatrix} (T^2_{trans} * p_{trans}) + (T^2_{exe} * (p_a + p_i)) \\ +(pre(T^2_{exe} * (p_a + p_i))) \end{pmatrix} \quad (4.11)$$

where p_{trans} represents the power consumption during $fn_{l,j}$ transmission in both fog layers, and $p_a\ and\ p_i$ represents the active and idle power consumption of a fn_l as represented in [61]. By combining (4.10) and (4.11), the total energy consumption $E(fn_{l,j})$ for both fog layers becomes

$$E(fn_{l,j}) = \sum_{FN_l \in FN} (E(fn_{l,j}^1) + E(fn_{l,j}^2)) \quad (4.12)$$

3.4. Problem formulation and Constraints

In this paper, we focus on the two-objective (2M) fitness function to optimize the time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ of fog nodes FN in both fog layers. The problem can be formally defined in (4.13), and the constraint of problem is shown in (4.14). The constraint guarantees that the gene values that are greater than the number of fog nodes FN will not be created by a chromosome.

$$\min T(fn_{l,j}), E(fn_{l,j}), \forall\, l \in \{1, 2, \dots, N\} \; and \; j \in \{1, 2, .., FN_l\} \quad (4.13)$$

$$s.t. \;\; \sum_{l=1}^{n} fn_l \leq N \text{ and } \sum_{(l,j)=1}^{n} fn_{l,j} \leq FN_l \quad (4.14)$$

where N represents the maximum number of fog nodes, FN_l represents the maximum set of computing task $fn_{l,j}$ and fn_l represents the fog node participating in computing task $fn_{l,j}$.

4. Multi-objective optimization method (MUOM) in fog computing

In this section, we propose MUOM based on NSGA-III, which is an accurate and efficient method for solving optimization problems with multiple objectives. In our proposed MUOM, the NSGA-III is used to optimize the time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ for data aggregation in fog computing as represented in (4.9) and (4.12), respectively. NSGA-III has high performance in search of an optimal solution and faster solution convergence. The NSGA-III introduces a selection method based on reference-point than the traditional NSGA-II method [294]. In the selection generation for NSGA-III, the reference-point guarantees the distribution's diversity for efficiently searching optimal solutions. First, we encode a strategy for optimal solutions, and then we initialize the fitness functions, constraints, and the first-generation of population. Afterward, we utilize the crossover and mutation operations for the new generation of solutions. The selection operations based on reference-point SAW [60], and MCDM [61] are chosen to select an optimal solution.

4.1. Encoding

We encode the optimal strategy for the time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ problem. In GA, chromosomes are composed of several genes, which represents an optimal strategy for FN. Figure 17 illustrates an example of an optimal strategy. In this example, a chromosome is an instance of the optimal strategy. The chromosome is encoded in an array of (0, 2, 3) integers.

$fn_{1,1}$	$fn_{1,2}$	$fn_{1,4}$	……	$fn_{l,j}$
3	2	0	……	3

Figure 17 Example of Encoding Chromosomes

Algorithm 4 Time Consumption Evaluation

Input: Optimal strategy **O** ($\boldsymbol{fn_{l,j}}$)

Output: Time consumption $\boldsymbol{T(fn_{l,j})}$

1. **for** $\boldsymbol{l = 1\ to\ N}$ **do**
2. **for** $\boldsymbol{j = 1\ to\ |FN_l|}$ **do**
3. Calculate $\boldsymbol{T^1_{trans}}$ by (4.2)
4. Calculate $\boldsymbol{T^2_{trans}}$ by (4.5)
5. Calculate $\boldsymbol{T^1_{exe}}$ by (4.3)
6. Calculate $\boldsymbol{T^2_{exe}}$ by (4.6)
7. Calculate $\boldsymbol{T^1_{wait}}$ by (4.4)
8. Calculate $\boldsymbol{T^2_{wait}}$ by (4.7)
9. $\boldsymbol{T(fn^1_{l,j}) = T^1_{trans} + T^1_{exe} + T^1_{wait}}$
10. $\boldsymbol{T(fn^2_{l,j}) = T^2_{trans} + T^2_{exe} + T^2_{wait}}$
11. **end for**
12. $\boldsymbol{T(fn_{l,j})}$= $\boldsymbol{T(fn^1_{l,j}) + T(fn^2_{l,j})}$
13. **end for**
14. return $\boldsymbol{T(fn_{l,j})}$

4.2. Fitness functions and Constraint

In GA, fitness functions predict whether a possible strategy is optimal or not. The fitness functions include two categories: the time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ for fog nodes FN, as represented in (4.9) and (4.12). The goal of the proposed MUOM is to find an optimal strategy for minimizing the fitness function's two categories, as shown in (4.13). The constraint associated with the fitness function is given in (4.14). We use NSGA-III for a hybrid optimization of the time consumption and energy consumption in fog computing. Also, NSGA-III addresses the multi-objective optimization problem (2M) with associated constraints.

Time consumption $T(fn_{l,j})$ is one of the fitness functions. Algorithm 4 presents the evaluation of time consumption $T(fn_{l,j})$. In this Algorithm, we input an optimal strategy denoted as O($fn_{l,j}$)**.** First, we calculate transmission time T_{trans}, execution time T_{exe}, and waiting time T_{wait} for data aggregation (lines 3 to 8). Then, we compute the time consumption $T(fn_{l,j})$ by fog nodes (lines 9 and 10). The time consumption of both fog layers together is the total time consumption in fog computing (line 12). Finally, the total time consumption is output in each task schedule.

Another fitness function is energy consumption $E(fn_{l,j})$. The evaluation of the energy consumption is elaborated in Algorithm 5. We first calculate transmission energy E_{trans}, execution energy E_{exe}, and waiting energy E_{wait} for both fog layers (lines 3 to 10). Then the total energy consumption $E(fn_{l,j})$ is an output.

Algorithm 5 Energy Consumption Evaluation

Input: Optimal strategy $\boldsymbol{O(fn_{l,j})}$

Output: Energy consumption E($\boldsymbol{fn_{l,j}}$)

1. **for** $\boldsymbol{l = 1\ to\ N}$ **do**
2. **for** $\boldsymbol{j = 1\ to\ |FN_l|}$ **do**
3. Calculate $\boldsymbol{E^1_{trans}}$ by (4.10)
4. Calculate $\boldsymbol{E^2_{trans}}$ by (4.11)
5. Calculate $\boldsymbol{E^1_{exe}}$ by (4.10)
6. Calculate $\boldsymbol{E^2_{exe}}$ by (4.11)
7. Calculate $\boldsymbol{E^1_{wait}}$ by (4.10)
8. Calculate $\boldsymbol{E^2_{wait}}$ by (4.11)
9. $\boldsymbol{E(fn^1_{l,j}) = E^1_{trans} + E^1_{exe} + E^1_{wait}}$
10. $\boldsymbol{E(fn^2_{l,j}) = E^2_{trans} + E^2_{exe} + E^2_{wait}}$
11. **end for**
12. E($\boldsymbol{fn_{l,j}}$)= $\boldsymbol{E(fn^1_{l,j}) + E(fn^2_{l,j})}$
13. **end for**
14. return $\boldsymbol{E(fn_{l,j})}$

4.3. Initialization

During the initial stages of GA, the parameters of GA needs to be determined and initialized. The parameters include the possibility of crossover POP_c, mutation POP_m, population size N_{pop} and maximum iterations Gen. In GA, the optimal strategy of the computing task $fn_{i,j}$ is indicated by each chromosome, $Crm_{l,i} = \{ g_{s,1}, g_{s,2}, \dots, g_{s,G}\}(i = 1, 2, \dots, N_{pop}, G = Gen)$, which is denoted as an array of integers. The chromosome $Crm_{l,i}$ consists of gene $g_{s,l}$ and the gene $g_{s,l}$ be an optimal strategy of $fn_{l,j}$ in the s^{th} schedule.

4.4. Crossover and mutation

In the crossover operation, two new chromosomes are generated from the combination of two-parent chromosomes. This operation is performed to acquire the better chromosomes while exchanging part of the gene's fragments from parent chromosomes. Figure 18 shows an example of a crossover operation. In this example, the crossover points for two chromosomes in a first schedule are determined. Then genes are swapped around the crossover point to generate two new chromosomes.

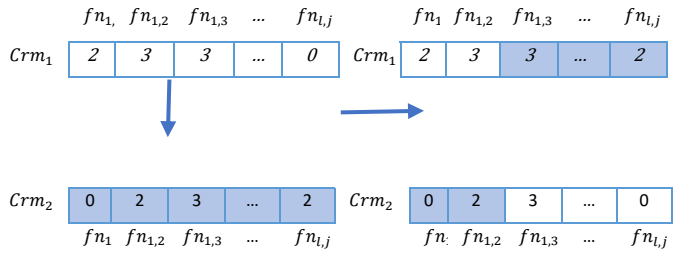


Figure 18 Example of crossover operation

After the crossover operation, the mutation operation is performed to generate better chromosomes. In mutation operation, some part of the chromosome genes is modified with higher fitness value, as illustrated in Figure 19.

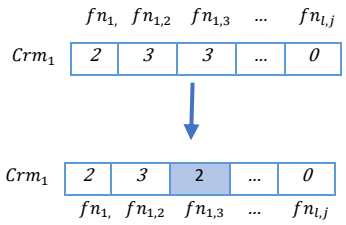


Figure 19 Example of mutation operation

4.5. Selection for the next generation

We aim to select chromosomes to generate individuals with higher fitness values for the next population. As discussed above each chromosome represents an optimal strategy for 2M objective. After crossover and mutation operations on the chromosomes, the population size increases to $2N_{pop}$. Algorithms 4 and 5 are used to evaluate the values of the 2M fitness function. For the next-generation population, optimal solutions' evaluating 2M fitness functions are sorted using a fast-non-dominated method. This procedure is carried out to generate non-dominated fronts $\{NF_1 + NF_2 + \cdots + NF_l\}$ with higher fitness values.

Then the generated fronts are randomly chosen to generate the next-generation population until the size of selected solutions are N_{pop}. By adding up the generated fronts, if the size becomes N_{pop}, then the procedure for selecting optimal solutions is finished, and the next generation is generated. Otherwise, optimal solutions need to be selected from the last l^{th} non-dominated front NF_l until the population size becomes N_{pop}.

After selecting optimal solutions, we adopt a normalization operation to normalize the 2M fitness function for all chromosomes in the population. In $2N_{pop}$ population, we search for the minimum time consumption and energy consumption, denoted as $T^{min}(fn_{l,j})$ and $E^{min}(fn_{l,j})$. The 2M objective values are computed as

$$T'(fn_{l,j})=T(fn_{l,j}) - T^{min}(fn_{l,j}) \quad (4.15)$$

$$E'(fn_{l,j})=E(fn_{l,j}) - E^{min}(fn_{l,j}) \quad (4.16)$$

Let φ^T, φ^E be a maximum value of time consumption and energy consumption in each dimension, which can be calculated by

$$\varphi^T = \max(T'(fn_{l,j})/wt^t) \quad (4.17)$$

$$\varphi^E = \max(E'(fn_{l,j})/wt^e) \quad (4.18)$$

where wt^t and wt^e represent a weight vector of the 2M fitness function.

Also, optimal solutions are sorted and selected in a non-dominated the l^{th} front NF_l. This process is repeated until all the solutions are selected. The selection steps are elaborated in Algorithm 6. In this Algorithm, the u^{th} generation (parent) represented as Gen_u is the input and the output is $(u+1)^{th}$ generation (child) denoted as $CGen_{u+1}$. Before sorting and selecting a solution for the next generation, we first compute each fog node's time consumption and energy consumption using Algorithm 4 and 5 (lines 2 and 3).

Then the non-dominated sorting for individual chromosomes with the size population of Gen_u is carried out (line 5). This sorting results in non-dominated fronts. Besides this, the population is selected primarily. The selected population S_u is constituted from fronts $\{NF_1 + NF_2 + \cdots + NF_l\}$ until the size of S_u becomes or exceeds N_{pop} (lines 6 and 7). Otherwise, a further selection is carried out (line 9). After selection, normalization is carried out, and the remaining optimal solutions are determined (lines 10-12). Finally, the next generation (child) population ($CGen_{u+1}$) is generated entirely.

4.6. Optimal selection using SAW and MCDM

In each population, chromosome represents an optimal solution to minimize time consumption T($fn_{l,j}$) and energy consumption E($fn_{l,j}$). To select the optimal chromosome from the population size N_{pop}, the reference-point-based SAW [60] and MCDM [61] are employed.

T($fn_{l,j}$) and E$\left(fn_{l,j}\right)$ are negative criteria as the higher the values are, the worse the solution becomes. Therefore, we normalize T($fn_{l,j}$) and E$\left(fn_{l,j}\right)$ in the i^{th} optimal strategy, as represented in (4.19) and (4.20).

$$N\left(T\left(fn_{l,j}\right)\right)=\begin{cases}\left(T_{max}(fn_{l,j})-T\left(fn_{l,j}\right)\right)/T_{max}(fn_{l,j})-T_{min}\left(fn_{l,j}\right), \\ \qquad T_{max}(fn_{l,j})-T_{min}(fn_{l,j})\neq 0 \\ I, \qquad T_{max}\left(fn_{l,j}\right)-T_{min}\left(fn_{l,j}\right)=0\end{cases} \tag{4.19}$$

$$N\left(E\left(fn_{l,j}\right)\right)=\begin{cases}\left(E_{max}(fn_{l,j})-E\left(fn_{l,j}\right)\right)/E_{max}(fn_{l,j})-E_{min}\left(fn_{l,j}\right), \\ \qquad E_{max}(fn_{l,j})-E_{min}(fn_{l,j})\neq 0 \\ I, \qquad E_{max}\left(fn_{l,j}\right)-E_{min}\left(fn_{l,j}\right)=0\end{cases} \tag{4.20}$$

where $T_{max}\left(fn_{l,j}\right),\ T_{min}\left(fn_{l,j}\right)\ and\ E_{max}\left(fn_{l,j}\right),\ E_{min}\left(fn_{l,j}\right)$ represents the maximum and minimum time consumption and energy consumption. To calculate the maximum values, $N\left(T\left(fn_{l,j}\right)\right)$ and $N\left(E\left(fn_{l,j}\right)\right)$ need to be combined with the associated weights $\frac{1}{2N}$ as shown in (4.21).

$$N\left(Crm_{l,i}\right)=\sum_{l=1}^{N}\frac{1}{2N}.N\left(T\left(fn_{l,j}\right)\right)+\sum_{l=1}^{N}\frac{1}{2N}.N\left(E\left(fn_{l,j}\right)\right)(1\leq i\leq N_{pop}) \tag{4.21}$$

where $N\left(Crm_{s,i}\right)$ represents the value of the i^{th} chromosome. The optimal solution represented by chromosome $N\left(Crm_{l,i}\right)$ can be computed as

$$N(C_l)=max_{i=1}^{N_{pop}}\,N\left(Crm_{l,i}\right)(1\leq l\leq N) \tag{4.22}$$

Algorithm 6 Selection for the next generation

Input: Parent Generation $\boldsymbol{Gen_u}$

Output: Child Generation $\boldsymbol{CGen_{u+1}}$

1. **for** $\boldsymbol{l=1\ to\ N}$ **do**
2. Calculate $\boldsymbol{T\left(fn_{l,j}\right)}$ by Algorithm 4
3. Calculate $\boldsymbol{E\left(fn_{l,j}\right)}$ by Algorithm 5
4. **end for**
5. Non-dominant sorting ($\boldsymbol{Gen_u}$) the POP solutions
6. Constitute $\boldsymbol{S_u}$ from fronts $\{\boldsymbol{NF_1+NF_2+\cdots+NF_l}\}$
7. Conduct Primary selection
8. **if** Size $(\boldsymbol{S_u})<\boldsymbol{N_{pop}}$ **then**
9. Conduct further selection
10. Normalize solutions by (15- 18)
11. Select remaining $\boldsymbol{z}$ solutions

12. $\boldsymbol{CGen_{u+1} = S_u \cup NF_l}$

13. **else**

14. $\boldsymbol{CGen_{u+1} = S_u}$

15. **end if**

16. **return** $\boldsymbol{CGen_{u+1}}$

4.7. Proposed MUOM overview

We aim at minimizing T($fn_{l,j}$) and E$\left(fn_{l,j}\right)$ for data aggregation in fog computing based on NSGA-III to obtain an optimal strategy for reducing T($fn_{l,j}$) and E$\left(fn_{l,j}\right)$. Algorithm 7 elaborates the overview of MUOM. In this Algorithm, we input the initialized population **N** and a maximum number of iterations Gen. The Algorithm outputs the optimal strategy for $T\left(fn_{l,j}\right)$ and $E\left(fn_{l,j}\right)$ in each schedule $(1 \le l \le N)$.

Firstly, the first-generation population is initialized. Then the child population is generated using crossover and mutation operations (lines 2-5). The child population size becomes $2N_{pop}$. This Algorithm also calculates the 2M fitness functions of $2N_{pop}$ solutions (lines 6 and 7), then the Algorithm selects the optimal individuals for the next generation. Next, the Algorithm evaluates the fitness function to select an optimal strategy using SAW and MCDM methods (lines 12 and 13). Finally, the optimal strategies are output (line 15).

Algorithm 7 Proposed MUOM in Fog computing

Input: The population size **N,** Max Iteration $\boldsymbol{Gen}$

Output: The optimal method $\boldsymbol{O}(\boldsymbol{fn_{l,j}})$

The optimal time consumption $\boldsymbol{T\left(fn_{l,j}\right)}$

The optimal energy consumption $\boldsymbol{E\left(fn_{l,j}\right)}$

1**. for** $\boldsymbol{l = 1\ to\ N}$ **do**

2. $\boldsymbol{i = 1}$

3. **while** $\boldsymbol{i \le Gen}$ **do**

4. **for** individuals: current population **do**

5. Crossover and Mutation operation

6. Calculate time consumption by Algo (4)

7. Calculate energy consumption by Algo (5)

8. **end for**

9. Selection for the next generation by Algo (6)

10. i++

11. **end while**
12. Evaluate objective function by (4.19-4.21)
13. Select an optimal strategy by (4.22)
14. **end for**
15. **return** $\boldsymbol{O(fn_{l,j})}$, $\boldsymbol{T(fn_{l,j})}$, $\boldsymbol{E(fn_{l,j})}$

5. Experimental Evaluation

This section presents our comprehensive simulation and experiments conducted to evaluate the performance of the proposed MUOM. A simulation-based evaluation is adapted because it controls environmental parameters and considers different constraints and scenarios for experiments. First, we introduce the test case scenarios for simulation, followed by a simulation setup including simulation parameters. Then, we discuss the performance evaluation of the proposed MUOM and comparative analysis with state-of-the-art methods.

5.1. Fog computing Test-case Architecture

We design a test-case scenario to test the effectiveness of the proposed MUOM and our scenario consists of three layers. In the first layer, we have 5-1000 sensor nodes to sense the heterogeneous data and generates a range of 1-1000 Kbs of data. Sensor nodes are randomly distributed within a range of 50-400 meters of fog nodes. The second layer is divided into two layers: Fog layer 1 and Fog layer 2. The number of fog nodes in each fog layer is between 10-1000. Fog layer 1 performs computing tasks $fn_{l,j}$ to process and analyze the data generated by sensor nodes. Whereas fog layer 2 performs computing tasks $fn_{l,j}$ for data storage and aggregation. Last, the third layer consists of data-owner and end-user devices, which are connected to the second layer through the internet.

5.2. Simulation Setup

A simulation is carried out via Network Simulator based on a Linux system with Intel (R) Core (i7), RAM 16.0 GB, and CPU 3.40 GHz. In the simulation, our present parameter values for the experiments conducted on the test-case scenario are presented in Table 12. We define a range of values of the corresponding parameters based on [294, 295].

Table 12 Parameters Settings

Parameter	Value
The Bandwidth of LAN	250 MB/s
The Bandwidth of WAN	20 MB/s
The Latency of LAN	(0.2-20) ms
The Latency of WAN	20 ms
Number of Fog nodes	10-1000
Number of Sensor nodes	5-1000

The Idle Power of Fog nodes	50 mW
The Active Power of Fog nodes	(100-500) mW
The Transmission Power	(100-500) mW
Computing capacity of Fog nodes	(1-50) GHz

The time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ of the fog nodes in fog layers are used to evaluate the performance of MUOM. For the comparative analysis of the proposed MUOM, the comparative methods are elaborated as follows.

1) Non-optimization Method (N-OPT)

In this method, computing tasks $fn_{l,j}$ for data aggregation including data encryption, data distribution, data storage, and additive aggregation are carried out on fog layers without utilizing any optimization method to minimize $T(fn_{l,j})$ and $E(fn_{l,j})$ [42].

2) Fully Cloud Method (FCM)

All the computing tasks $fn_{l,j}$ are fully offloaded from fog nodes to cloud to process, store, and aggregate data. NSGA-III method is considered for optimization of $T(fn_{l,j})$ and $E(fn_{l,j})$.

3) Partial Cloud Method (PFCM)

This method is a partial offloading of $fn_{l,j}$ from fog nodes to cloud using Ant-Colony Optimization (ACO) method [292]. The method based on ACO aims to find an optimal solution for $T(fn_{l,j})$ for data aggregation at fog nodes, and cloud for partial processing and aggregation of data.

These comparative methods are implemented under the same simulation setup of fog layers and sensor nodes, as discussed above.

5.3. Evaluation Criteria

We evaluate the performance of the proposed MUOM in terms of evaluation metrics including the number of fog nodes, the execution and transmission power, the computing capacity, the data size, the degree of workload imbalance, and the standard deviation of the workload imbalance.

The degree of workload imbalance shows the imbalance of workload among fog nodes. The imbalance can be calculated by considering the formula from [292] as shown in (4.23).

$$WL = \frac{\left(Max(R_l) - Min(R_l)\right)}{R_{average}}, \qquad l = 1, 2, \ldots . N \tag{4.23}$$

where $R_l = T(fn_{l,j})$. The workload imbalance WL is the difference of $T(fn_{l,j})$ of fog nodes to the average $T(fn_{l,j})$ of fog nodes.

The standard deviation evaluates the workload distribution among the fog nodes. The smaller the value of deviation, the higher the workload balanced between the fog nodes. The standard deviation can be calculated from [292] as

$$S.D = \sqrt{\frac{\sum_l (R_l - R_{average})^2}{N}} \tag{4.24}$$

5.4. Performance Evaluation of Proposed MUOM

This section evaluates MUOM to analyze the impact of the number of fog nodes, the execution and transmission power, and the computing capacity of fog nodes.

5.4.1. Impact of the Number of Fog nodes

We consider the impact of the number of fog nodes on $T(fn_{l,j})$ and $E(fn_{l,j})$ of our MUOM method. In the simulated network, fog nodes are responsible for performing computing task $fn_{l,j}$ including data encryption, data division, distribution, storage, and additive aggregation. The number of fog nodes that can perform computing tasks is ranging between 10-1000, as listed in Table 12. The results are illustrated in Figure 20 (a), (b), and (c).

According to the Pareto front chart in Figure 20 (a), we can observe that the MUOM method based on NSGA-III can get the optimal solutions $O(fn_{l,j})$ balance between extreme values of V1 and V2 for time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$. With the number of fog nodes increasing, the range of Pareto also increases, corresponding to the higher number of optimal solutions. For 1000 fog nodes (FN-1000), the optimal solutions are notably higher than the 500 fog nodes or fewer fog nodes. The higher number of optimal solutions is because of the higher computation tasks $fn_{l,j}$ with the increase of fog network size. We can also notice that the optimal solutions have a high degree of overlaps for the number of fog nodes less than 250 at reaching certain time and energy levels.

From our analysis, it can be concluded that the dimension of decision-making for optimal solutions becomes more extensive with a higher number of fog nodes rather than a fewer number of fog nodes. Figure 20 (b) and (c) depict the impact of the number of fog nodes on the time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$, respectively. The average time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ show a positive correlation with the increase in the number of fog nodes.

Although the increase in the number of fog nodes minimizes the execution time T_{exe} by a division of computing tasks $fn_{l,j}$ among fog nodes. Still the transmission time T_{trans} for requesting, transmitting, and authenticating $fn_{l,j}$ requires a larger amount of time than T_{exe}. Therefore, the overall time consumption $T(fn_{l,j})$ increases with an increase in the number of fog nodes, as shown in Figure 20 (b).

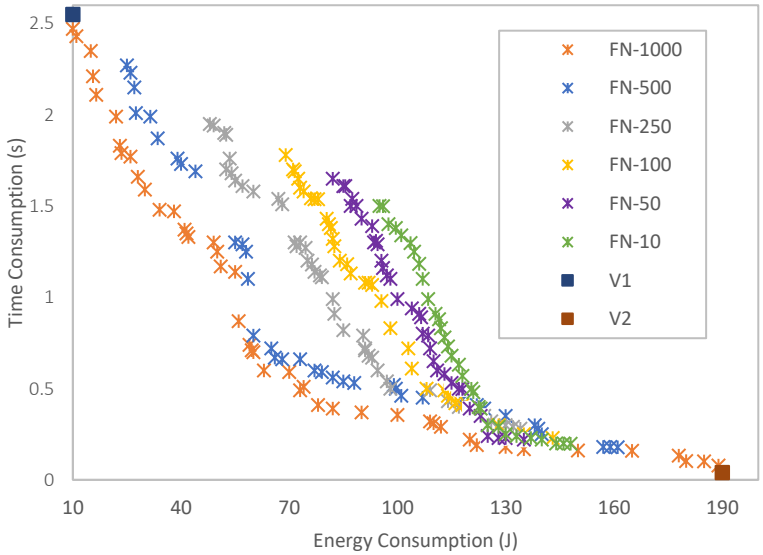
The relationship between energy consumption $E(fn_{l,j})$ and the number of fog nodes is shown in Figure 20 (c). Similar to T_{trans}, the transmission energy E_{trans} consumed by fog nodes for $fn_{l,j}$ including data transmission, data authentication, and data request becomes higher than the execution energy E_{exe}. This increase impacts the overall increase in energy consumption $E(fn_{l,j})$ for a higher number of fog nodes.

5.4.2. Impact of the Execution and Transmission Power

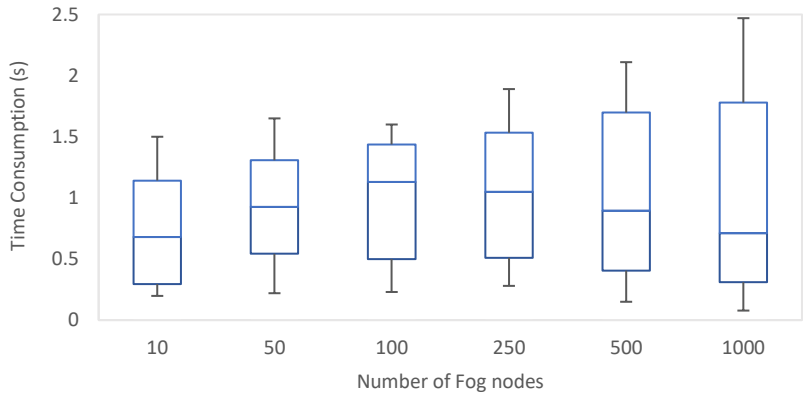
We focus on the influence of execution and transmission power on the time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ in this section. The range of power, including transmission p_{trans} and execution power p_a, p_i (idle and active power), varies from 100 to 1000 mW as listed in Table 12. Figure 21(a) depicts multiple optimal solutions in a Pareto chart with varying power values. From the Figure, we can observe that the proposed MUOM method based on NSGA-III can always find multiple optimal solutions between extreme V1 and V2 values. In addition, the optimal solutions for transmission and execution power greater than 500 have a high degree of overlapping. It can be concluded from the overlapping that the dimension of decision-making for optimal solutions becomes very small when the transmission and execution power reaches 500 mW.

The relationship between time consumption $T(fn_{l,j})$ and power is negatively correlated, as shown in Figure 21 (b). The processing and transmission speed of computing tasks $fn_{l,j}$ becomes faster with the more considerable power, which results in smaller T_{trans} and T_{exe} at fog nodes. In contrast, energy consumption $E(fn_{l,j})$ shows a positive correlation with power, as illustrated in Figure 21 (c). With an increase in p_{trans}, p_a and p_i, the higher energy consumption $E(fn_{l,j})$ is required to process $fn_{l,j}$ within a network.

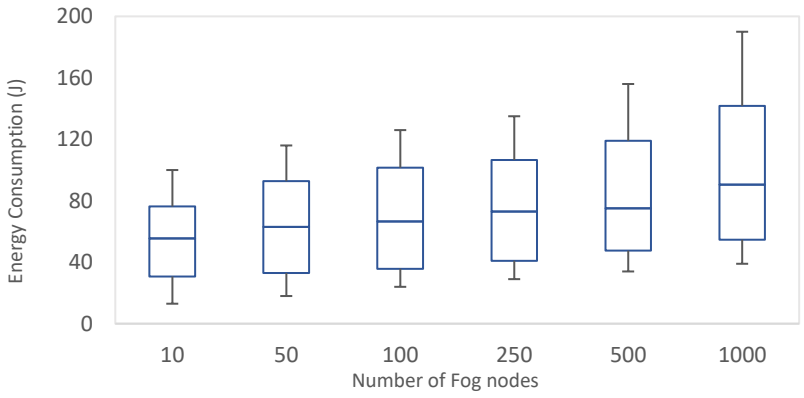
Overall, we have noticed a slowdown in trend for $E(fn_{l,j})$ increase and $T(fn_{l,j})$ decrease. Although the power is increasing evenly, still the degree of $T(fn_{l,j})$ and $E(fn_{l,j})$ is shrinking



(a)



(b)



(c)

Figure 20 Impact of the number of fog nodes. (a) Pareto front for optimal solutions. (b) Box plots of the time consumption in a varying number of fog nodes. (c) Box plots of the energy consumption in a varying number of fog nodes.

gradually. Therefore, the effect of transmission and execution power on the network is not higher than the other variables' impact, including the number and computing capacity of fog nodes.

5.4.3. Impact of Computing Capacity of Fog nodes

This section discusses the relationship between the computing capacity C_{cap} of fog nodes and the network performance. In our experiment, the range of computing capacity C_{cap} varies from 1 to 50 GHz as listed in Table 12 and the results are depicted in Figure 22.

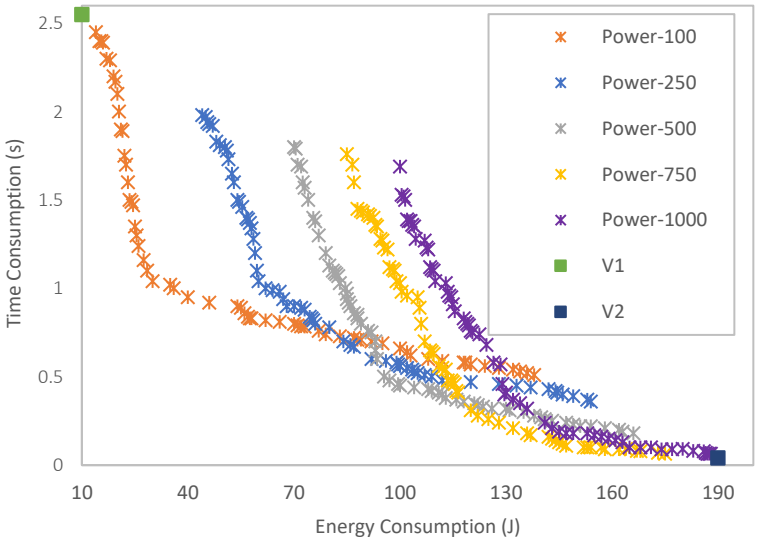
The optimal solutions in the Pareto chart with a variation of fog nodes' computing capacity are shown in Figure 22 (a). We can observe that the multiple optimal solutions fall between extreme V1 and V2 values. The extreme value at the top left of Figure 22 (a) has the maximum time consumption $T(fn_{l,j})$ of 2.55s and the minimum energy consumption $E(fn_{l,j})$ of 10J. For the maximum time consumption $T(fn_{l,j})$ and the minimum energy consumption $E(fn_{l,j})$, each optimal set of Pareto has almost the same value as the extreme value at the top left. In addition to the obvious relationship, we also notice that the lower computing capacity C_{cap} values, i.e. C-1 and C-20, make the scope of the optimal solutions denser and smaller as compared to values greater than 20 GHz.

Figure 22 (b) and (c) show the box plot relationship of time consumption $T(fn_{l,j})$ and energy consumption $E(fn_{l,j})$ with computing capacity C_{cap}. The time consumption $T(fn_{l,j})$ shows a negative correlation with computing capacity C_{cap}. As the computing capacity C_{cap} of fog nodes increase the execution time T_{exe}, and the transmission time T_{trans} for $T(fn_{l,j})$ becomes smaller. In contrast, the energy consumption $E(fn_{l,j})$ is positively correlated with computing capacity C_{cap}. The higher the computing capacity C_{cap}, the more the energy is consumed by fog nodes for processing data.

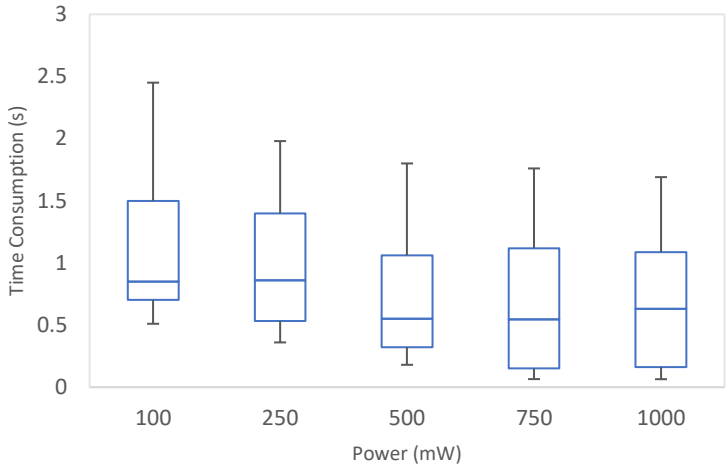
From Figure 22 (b) and (c), it can be concluded that the time consumption $T(fn_{l,j})$ decreases moderately with the computing capacity C_{cap} increasing evenly. Similarly, the energy consumption $E(fn_{l,j})$ increases moderately with the computing capacity increase.

5.5. Comparison Analysis

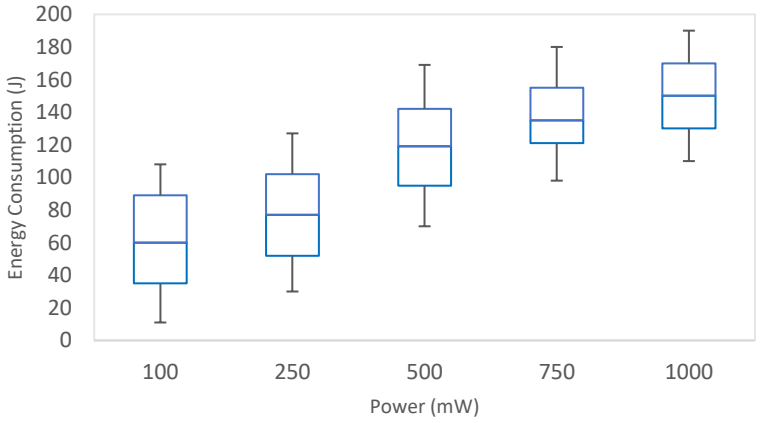
In this section, we evaluate the performance of the MUOM method with the N-opt, FCM, and PFCM methods. The time consumption and energy consumption with the data size and power consumption are metrics to assess MUOM and the other comparative methods' performance. Also, we evaluate the standard deviation and the degree of imbalance of MUOM is compared with PFCM.



(a)

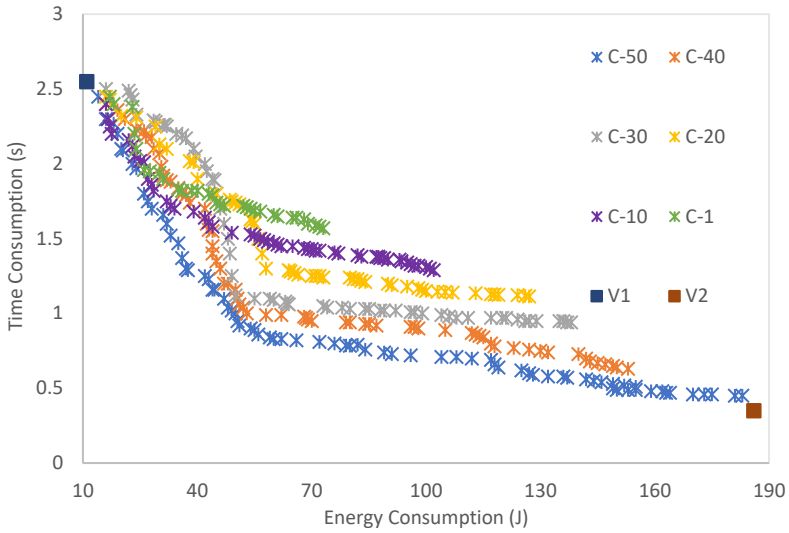


(b)

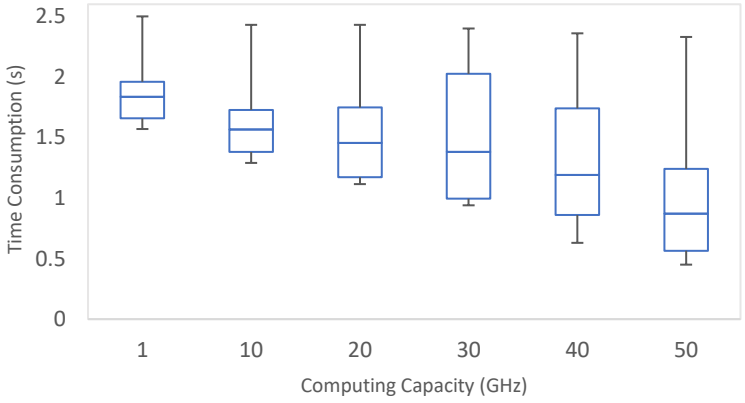


(c)

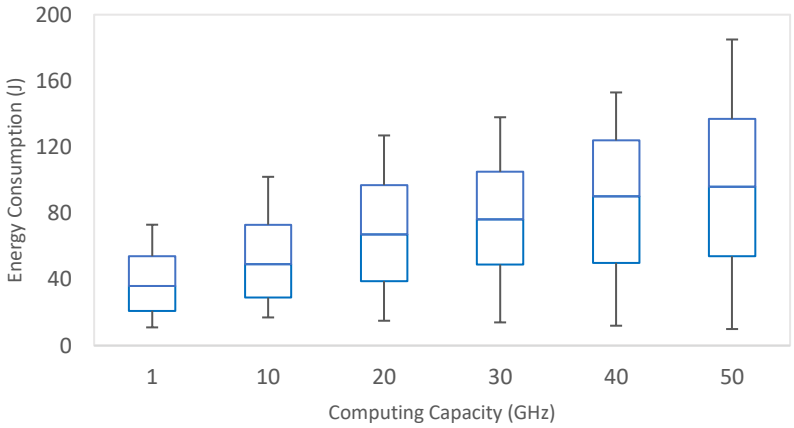
Figure 21 Impact of execution and transmission power. (a) Pareto front for optimal solutions. (b) Box plots of the time consumption in varying power values. (c) Box plots of the energy consumption in varying power values.



(a)



(b)



(c)

Figure 22 Impact of the computing capacity of fog nodes. (a) Pareto front for optimal solutions. (b) Box plots of the time consumption in varying computing capacity values. (c) Box plots of the energy consumption in varying computing capacity values.

5.5.1. Comparison of data sizes for time consumption and energy consumption

Figure 23 and Figure 24 present the time consumption and energy consumption in terms of data size from 1 to 1000 Kbs in processing the MUOM, N-opt, FCM, and PCFM methods. As discussed in Section 3, all the computing tasks $fn_{l,j}$ in the N-opt methods are executed and transmitted without applying the fog layers' optimization algorithm. Due to the lack of time consumption and energy consumption optimization, the transmission time and energy incurs high overhead for transmission of data including encryption, authentication, and key distribution. With a larger data size, the transmission time and energy increase abruptly, which results in an overall increase in time consumption and energy consumption.

The reasons for the FCM and PCFM results variation with data sizes for time and energy consumption are summarized in detail as follows.

On the one hand, the time consumption of FCM is a little higher than in PFCM, as shown in Figure 23. The fog nodes in FCM are connected to the cloud through WAN with lower bandwidth and higher latency than the fog nodes interconnected in PFCM through LAN. In FCM, the execution of computing tasks and data storage is carried out on the cloud. In contrast, PFCM executes computing tasks on fog layers and partially performs further execution on a cloud. Hence less time is consumed when the computing task is executed on fog layers in PFCM than entirely on the cloud in FCM.

Further, resources available for higher data size execution are limited and finite in fog layers. In the case of all fog nodes instantiated for computing tasks, then the execution requests for the remaining tasks in the queue have to wait until the fog node's resources become available. In PFCM, only partial computing tasks wait for resource availability in fog layers is required. Moreover, partial computing tasks are offloaded to the cloud. Also, the ACO algorithm is used to optimize the time consumption for tasks offloading in fog layers.

On the other hand, the energy consumption in PFCM for computing tasks without optimizing energy in fog layers is a bit higher than FCM, as shown in Figure 24. In PFCM, most of the computing tasks are executed at fog nodes, which requires higher energy consumption than all the computing tasks offloaded to the cloud in FCM. Further, ACO optimizes only a single-objective, i.e. time consumption in PCFM. Thus, energy consumption in PFCM is not optimized, and higher energy is consumed when the computing tasks are executed on fog layers than on the cloud as in FCM.

Compared to MUOM, partial tasks offloading to the cloud increases the time consumption and energy consumption of PCFM, as shown in Figure 23 and Figure 24. In MUOM, computing tasks are performed at fog layers, and no task is offloaded to the cloud. Further, optimal

solutions provided in PCFM are based on an ACO algorithm with single-objective optimization, i.e. time consumption, which incurs higher computational time and energy than MUOM. MUOM is based on NSGA-III, which provides a hybrid strategy for multi-objective optimization, i.e. time consumption and energy consumption.

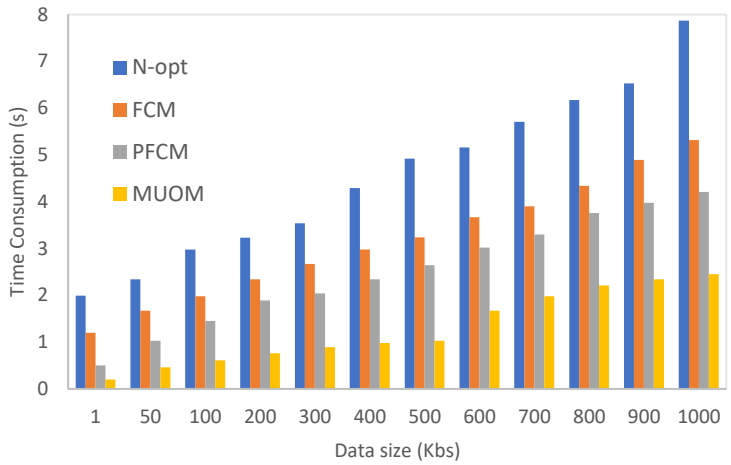


Figure 23 Data size comparison for time consumption

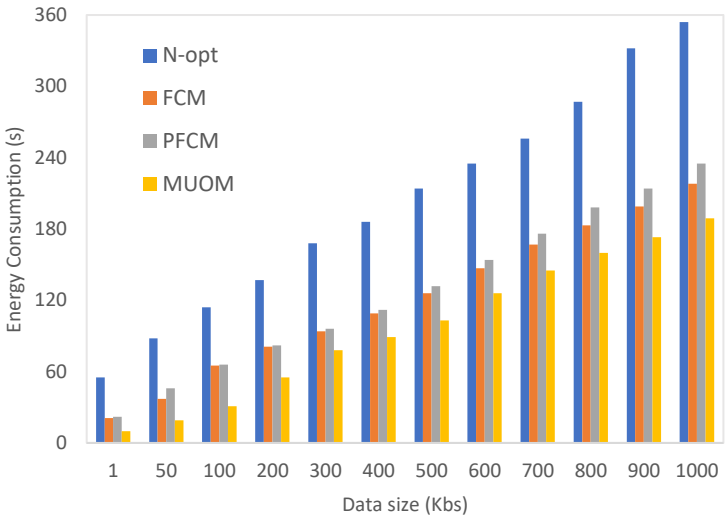


Figure 24 Data size comparison for energy consumption

5.5.2. Comparison of power consumption

Figure 25 and Figure 26 illustrate the time consumption and energy consumption in terms of power consumed by each of the four methods. Figure 25 depicts that the increase in power consumption decreases the overall time consumed by each method to execute and transmit the computing tasks. The proposed MUOM method incurs significantly less time as compared to the N-opt, FCM, and PFCM methods. Due to the lack of an optimization algorithm, the time consumed by fog nodes with varying power consumption in the N-opt method is remarkably high than the other three methods. In FCM, the tasks offloading to the cloud require a little

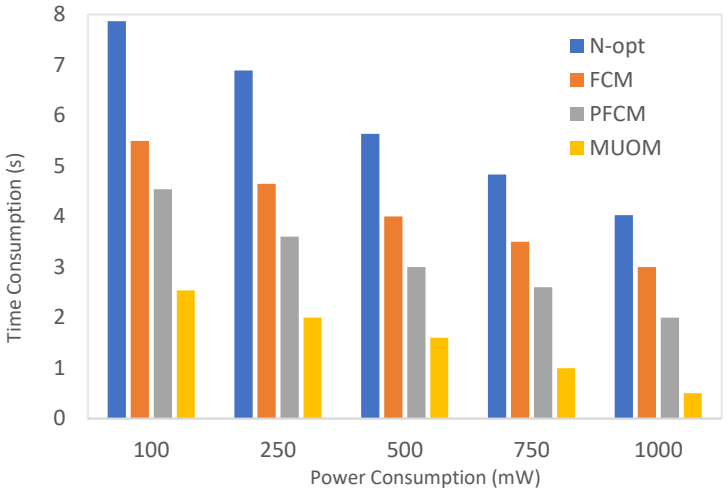


Figure 25 Power comparison for Time Consumption

higher power and time consumption than PFCM. The power consumption has a high impact on the energy consumed by methods to execute and transmit the computing tasks, as shown in Figure 26. We consider power as one of the parameters to measure the energy consumption of fog nodes.

We can conclude from Figure 26 that the increase in power consumption results in higher energy consumption for executing and transmitting tasks. Besides, the average power consumption of MUOM is a little lower than FCM and PFCM. At the same time, the power consumption of MUOM is remarkably lower than the N-opt method. The reason for the lower power consumption of MUOM is the optimization of energy consumption and performance of computing tasks within a fog layer.

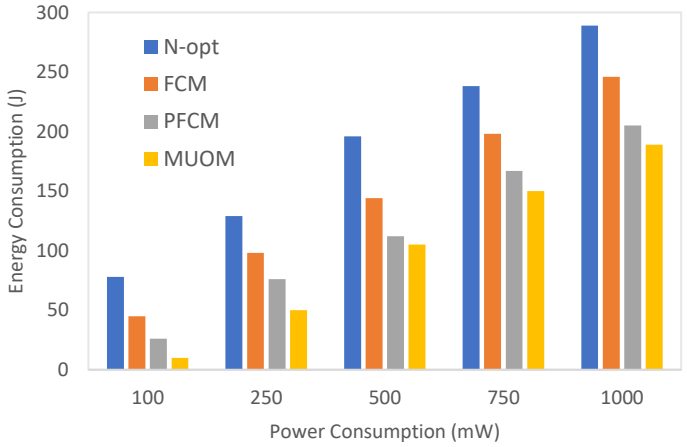


Figure 26 Power comparison for Energy Consumption

5.5.3. Comparison of workload imbalance

Figure 27 depicts the degree of workload imbalance at the fog layers for the MUOM and PFCM methods with increasing fog nodes. The workload imbalance is evaluated according to (4.23). Figure 27 shows that the degree of imbalance for MUOM is significantly lesser than PCFM, which means that MUOM balances computing workload at fog nodes effectively. With the optimal

solutions, the workload is balanced among fog nodes to execute and transmit computing tasks. In contrast, PFCM utilizes the ACO Algorithm for only optimizing time, not considering energy optimization. Due to high execution and transmission energy, the workload imbalance in PFCM is remarkably high than MUOM.

Similarly, Figure 28 shows the standard deviation of the workload distribution among fog nodes with an increasing number of fog nodes. The Figure depicts that the MUOM enhanced the standard deviation compared to PFCM, which means smaller standard deviation values, the higher the workload balance and distribution among fog nodes.

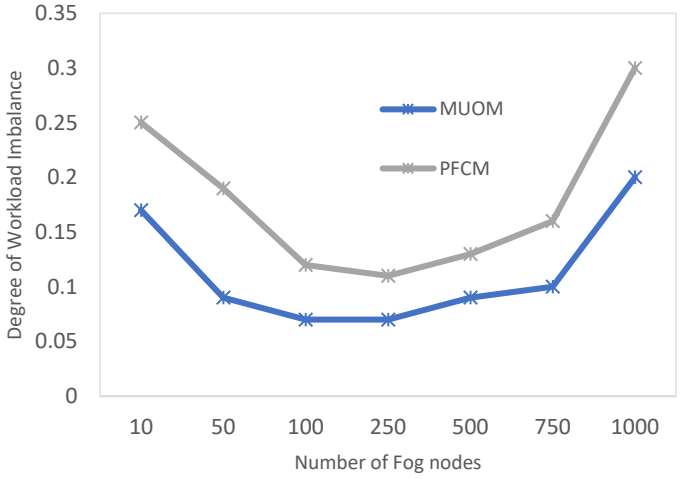


Figure 27 The degree of workload imbalance.

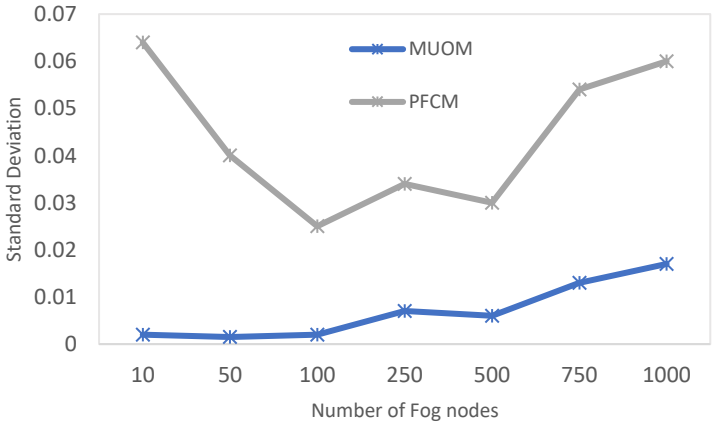


Figure 28 The standard deviation of workload distribution

6. Conclusion

In this paper, we investigated the problem of time consumption and energy consumption for data aggregation in fog computing. We provide two models for time consumption and energy consumption in fog computing and to optimize both models, we proposed MUOM based on the

NSGA-III Algorithm. Furthermore, comprehensive experiments and evaluations are carried out to analyse and compare the performance of the proposed MUOM with the other methods. Our experiment results showed that our MUOM can always obtain the Pareto optimal solutions within the extreme values and it outperforms the state-of-the-art methods in solving the optimization problem.

For future work, we will extend the proposed strategies in real-world scenarios of IoT. In addition, we will apply the proposed strategies for optimization of data replication in Fog-enabled IoT.

'Teach one girl how to code, she'll teach four. The replication effect is so powerful'.

--- Reshma Savjani

Chapter 5: Efficient Privacy-Preserving Data Replication for Fog-enabled IoT

Abstract

Internet of Things (IoT) devices continuously generate a high volume of data that is processed and stored in traditional cloud computing. The processing including data replication in traditional cloud computing often results in excessive resource utilization, performance overhead, and long response time. Fog computing has been proposed to overcome the shortcomings of cloud computing. Fog computing alleviates the processing and storage burden during data replication from cloud to the network edge closer to sensor devices. Numerous data replica schemes in fog computing have been proposed to improve the performance efficiency of the data, reduce the turnaround delays of data access, and minimize network latency. However, these schemes do not consider data replication privacy, which is essential for data protection, reliability, and authentication. Therefore, this paper proposes a data replica creation scheme and a data replica placement scheme for preserving the privacy of data in fog computing. Our proposed replica creation scheme is based on a Level of Privacy (LoP) defined by data-owners and service capacity of fog nodes. Our proposed replica placement scheme is based on the priority level of fog nodes. We have conducted a comprehensive experimental analysis to compare the performance of our scheme and the existing schemes. Our results demonstrate that the proposed scheme can achieve significant efficiency for replicas privacy, prediction accuracy, as well as outperform the existing state-of-the-art schemes in terms of computational and memory costs.

This contribution has been submitted to Future Generation Computer Systems Journal (Elsevier)

1. Introduction

According to research, 41.6 billion Internet of Things (IoT) devices will be generating 79.4 zettabytes of data by the year 2025 [6]. A high volume of data generated by IoT devices is processed and stored in cloud computing. The cloud computing has a strong dependency on network performance, bandwidth, and response time for IoT devices' data processing and storage. Data access and processing can be a bottleneck due to long turnaround delays and high demand for network bandwidth in remote cloud systems [52]. Fog computing brings cloud services to the edge of the IoT network, so it has many advantages, such as low-performance overhead, faster response time, and high network bandwidth for local computation [13, 62, 296]. In this context, fog computing can be referred to fog-enabled IoT.

The fog-enabled IoT network provides a promising way for data replication management of IoT devices generated data. Data replication is one of the most efficient and commonly used methods for data reliability in wireless sensor networks [297]. Data replication significantly improves data processing efficiency, data access, and reduces data transmission delays. Generally, data replication technique [47] consist of static and dynamic methods for replica creation, placement, and selection. Static methods create several data replicas at the time of processing system setup [45]. The main drawback of static methods is that a data replica creation and placement process is unaffected by the changes in storing/deleting replicas and user's data access patterns. Therefore, the method does not provide an accurate replicas creation which affects the replicas selection and placement [48]. Dynamic methods consider the dynamic nature of user's access patterns and store/delete replicas for data availability [45]. Also, dynamic methods take advantage of data mining approaches for data relationship management.

The dynamic and static methods for data replication consist of replica creation, placement, selection, and replacement processes [45]. A replica creation process determines the number of replicas of a data object. A replica placement process first determines the best possible location for replica creation and then decides replica placement based on network protocols. A replica selection process determines an appropriate replica location for job execution. The job execution includes replica creation, replica processing, and storage. In the case of storage limitations, a replica replacement process can change the replica locality with the new replica [46].

In recent years, several data replication schemes in fog computing have been proposed to reduce network latency, improve the performance efficiency of the data, and reduce the turnaround delays of data access [50-53]. However, these schemes do not consider the privacy of data replicas, which is essential for data protection, reliability, and authentication during

different data replication processes in an insecure computing environment. Data replication processing on various fog nodes in a network makes data replicas vulnerable to attacks like Denial of Service (DoS), authentication, and MAC spoofing attacks [48, 298]. An adversary can either modify/delete data replicas to make data unavailable to end-users or acquire replicas to monitor data-owner patterns and sensitive locations. Therefore, preserving the privacy of data replicas at fog nodes is vital to guarantee data protection, reliability, authentication, and survivability. Although there are existing works [50-53] for data replication in fog computing, the preservation of data replication privacy is not considered.

To address the shortcoming mentioned above, our research aims to provide efficient privacy-preserving data replica creation and placement schemes. Our proposed data replica creation schemes consider important factors, including service capacity and data owner's defined Level of Privacy (LoP) of fog nodes. The replica placement scheme utilizes privacy-preserving priority levels and service capacity of fog nodes. We design these two schemes to ensure adequate data replica privacy with low computational and storage costs. The main contributions of this paper are summarized as follows.

1. We propose data replication schemes in fog-enabled IoT to efficiently process replicas and preserve the privacy of replicas.
2. We propose a data replica creation scheme that can efficiently select fog nodes with the highest service capacity for replica creation. Also, the scheme can efficiently generate replicas based on the LoP defined by data-owners.
3. We consider a priority level based on the LoP and service capacity for a data replica placement scheme.
4. We provide an experimental and comparative analysis of both schemes. The analysis shows that the performance efficiency in terms of computational and memory cost of the proposed schemes is better than the state-of-the-art schemes.

The remainder of the paper is organized as follows: In Section 2, we review and discuss related work. In section 3, we present our proposed schemes along with our system and adversary models and time complexity analysis. Section 4 provides experimental results, privacy, and performance analysis. Finally, Section 5 concludes the paper with future work.

2. Related Work

2.1. Data Replica Creation and Placement

One of the most challenging issues in a real-time networking environment is data management. Recently, researchers have investigated this issue and provided data replication-based solutions

to mitigate and enhance data performance and reliability [299]. In this section, we present a review of data replication research for data management in cloud computing.

In hybrid cloud computing, Zhao *et al.* proposed a dynamic replica creation scheme based on file-access heat and node load methods [300]. In the scheme, the Markov chain model is used for file-access heat and node load methods to adjust the number of data replicas per node. The performance analysis results proved that the dynamic replica creation scheme reduces the response time and improves load balancing during the adjustment of several replicas in comparison to the HDFS (Hadoop Distributed File System). Also, due to a hybrid cloud, storage usage is reduced while achieving increased data reliability as compared to the HDFS. Although the replica creation scheme can minimize the response time and storage usage in hybrid cloud, the use of the replica replacement method increases processing overhead and data response delay. The replacement method is based on a hierarchical tree structure for the replica's deletion and update. The replica replacement method incurs high storage and computing overhead in a hybrid cloud.

Another replica placement policy (RPP) scheme based on HDFS for the data replica placement has been proposed by Dai *et al.* in [301]. The proposed RPP provides two advantages, firstly the scheme eliminates the need for load balancing utility for managing the data replica load. Secondly, RPP considers the heterogeneous environment as compared to HDFS for each data processing site. The comparative analysis presented in the scheme is based on the requirements of HDFS, which can balance and store replicas at appropriate network locations. The utilization of HDFS balancing requirements incurs high network and processing overhead for balancing and storing replicas.

For the HDFS, the concept of a Markov chain model for dynamic data replication has also been proposed in the scheme [302]. In this scheme, the number of replicas varies with the change in data due to cold and hot temperatures. For data replicas creation, the maximum availability/accessibility of data in the cloud data-center based on fuzzy logic has been considered in another scheme [303]. In both schemes [302, 303], data file popularity and computing node load are considered crucial factors for data replicas creation. Instead of distributed cloud resources, these schemes [302, 303] rely on centralized resources which significantly increase the average response time delay of nodes for replica processing and placement. A scheme based on multi-objective optimization for improving data response time, availability and load balancing has been proposed in [304]. In the scheme, the data availability has been monitored to balance the increase in data replication cost using load balancing method. The concept of knapsack and multi-tier hierarchical methods to provide efficient data-sharing management has been presented in a prefetching-aware data replication scheme [47].

A service provider is responsible for determining the total replication computational and storage cost required at the data center [48]. The scheme balances different objectives, such as load balancing, availability, and replication cost. An extensive experimental analysis using cloudsim proved that the proposed replica management scheme is energy efficient. However, the file popularity/ heat factor was not used in computing the replica management process. File popularity/ heat provides file usage and access patterns, which significantly improves response time in replica management.

Mansouri *et al.* presented a dynamic file popularity-aware replica scheme [305] to determine data replica creation. In the scheme, first, data file popularity is computed based on the user's access behavior. Then the most frequently access data is stored at the best possible location, which is identified based on the centrality, service capacity, and the number of file requests [46, 47]. Also, the scheme applies a parallel downloading approach for assembling replication of data files. Although the scheme can enhance replica creation using the proposed file popularity mechanism, it neglects the impact of file access patterns in decision making for replica selection.

A data replica placement scheme based on a bidding concept is proposed in [306]. In this scheme, factors of bidding are combined with characteristics of self-replication to initiate an activity for replica creation. When the availability of a file does not meet the given requirements for replica creation activity, then the bidding activity of replica placement takes place. In the scheme, replica placement considers service capacity and access probability criteria of each node for file availability. To enhance the data replica placement in cloud, Lizhen *et al.* constructed an approach based on a genetic algorithm using a three-layer graph [307]. In this approach, a genetic algorithm reduces the data transmission overhead in the cloud. Permutation in a genetic algorithm requires a high computational cost. Therefore, the computational overhead of the scheme for replica placement is high as compared to [306].

Most of the schemes [47, 300-307] discussed above depend on cloud or centralized/ database systems for computation and storage. Performing data replication on cloud/centralized system increases the computational and storage burden at the computing end, which results in degraded data reliability, scalability, and high bandwidth overhead. Overall response time of end-user requests to access data is also delayed due to the server's far away locality [300, 302, 303]. Synchronization issue during data replication is also one of the problems w.r.t. the response time delay. Further, most of the existing schemes are only suitable for homogenous data replication [47, 300, 302, 303].

2.2. Data Replica Creation and Placement in Fog-enabled IoT

Recently, researchers have begun to pay attention to fog computing for data replication [50-53]. Huang *et al.* proposed a latency-aware data replica placement scheme based on a greedy algorithm [50]. The scheme reduces the overall latency of the network using a pruning method. The performance analysis of the scheme proved that the pruning method can efficiently cope with real-time data scheduling. For decentralized replica placement, Aral *et al.* proposed a dynamic replica scheme [51]. The scheme relies on data request monitoring to create, replace, and delete replicas dynamically.

In [52] Shao *et al.* Introduced a collaborative computing environment for replica placement in the scheme. The scheme is based on integer programming and a swarm optimization algorithm to optimize the number of replica's placement in a decentralized network Naas *et al.* also investigated integer programming with a heuristic approach using geographical zoning in iFogStor for replica's placement in fog computing [53]. However, all of the above-mentioned schemes [50-53] did not take into account the data protection for replicated sensitive data to various locations. Sensitive data is replicated in plain format across fog nodes, which may expose data to unauthorized fog nodes/end-users for misuse.

2.3. Privacy in Data Replication

Data privacy is a deep concern for any wireless sensor network, including IoT and fog/ cloud computing due to invulnerability from different kinds of attacks such as DoS, authentication, MAC spoofing attacks [48]. Data-owner's sensitive data replication on various locations in the network is vulnerable to such types of attacks. For example, an adversary can modify or delete replicas to make data unavailable to end-users. Therefore, preserving replicated data privacy at various locations is essential for data reliability, authentication, and survivability.

Mansouri & Sharma *et al.* highlighted data protection concerns in a cloud environment [48, 49]. Sharma *et al.* proposed a scheme for data protection using data division into small fragments that can then be replicated to different locations in cloud using a fragment placement algorithm [49]. The scheme did not rely on cryptographic measures to encrypt data. Sharma *et al.* claimed that the non-cryptographic nature of the scheme makes it faster to perform replica placement operations in the cloud.

Similarly, Jayasaree and Saravanan considered a data security scheme [48] for data replicas. In the scheme, a particle swarm division algorithm has been adopted for optimizing the placement of the replicas in cloud computing. The scheme divides replicas into fragments and then distributes and stores them using the T-colouring concept [48]. Since the data fragments are not encoded, and they are distributed to different locations in the cloud, an adversary

exposing few fragments would be able to analyze and discover data patterns and their meaning. Also, the privacy and reliability of the data are not guaranteed. The schemes [48, 49] are also vulnerable to DoS and authentication attacks.

In the above-mentioned schemes [47-53, 300-307] fog/edge computing to optimize performance efficiency for data replica privacy has not been considered. Therefore, our study aims to design a data replication scheme that can preserve the privacy of data replicas while satisfying data reliability and performance efficiency in fog-enabled IoT.

3. Data Replica Creation and Placement Schemes: Model and Solution

In this section, we describe the data replica creation and placement schemes based on the system and adversary models. First, we discuss our proposed system model. Then we present our adversary model. Based on the system and the adversary model, we then discuss our schemes with Algorithms and formulas. We also present a time complexity analysis of the Algorithms. Different symbols used in this paper are given in Table 13.

Table 13 Summary of Symbols and Abbreviations

Symbol	Explanation
$\boldsymbol{m_{Sc_i}}$	Service capacity of the i^{th} node
$\boldsymbol{LoP_l}$	Level of Privacy defined for l^{th} file
D.F	Division Factor
$\boldsymbol{B_i}$	Total number of blocks at fog node
$\boldsymbol{f_{pri}}$	Priority level of a fog node
$\boldsymbol{file_l}$	l^{th} data file
$\boldsymbol{b_k}$	k^{th} data block of l^{th}data file
$\boldsymbol{m_{l_i}}$	Load condition of the i^{th} node
$\boldsymbol{m_{t_{ij}}}$	Response time required by the i^{th} node to transfer or receive the j^{th} node
$\boldsymbol{tl_i}$	Total load of the i^{th} miner node
$\boldsymbol{l_{avg}}$	Average load of miners in fog layer 1

3.1. System Model

Figure 29 shows an overview of the fog-enabled IoT network architecture, which is based on the system model presented in [42]. Fog nodes in the architecture are critical components for fetching, analyzing, and processing the data coming from sensor devices. In Figure 29, the fog-enabled IoT architecture consists of two fog layers:

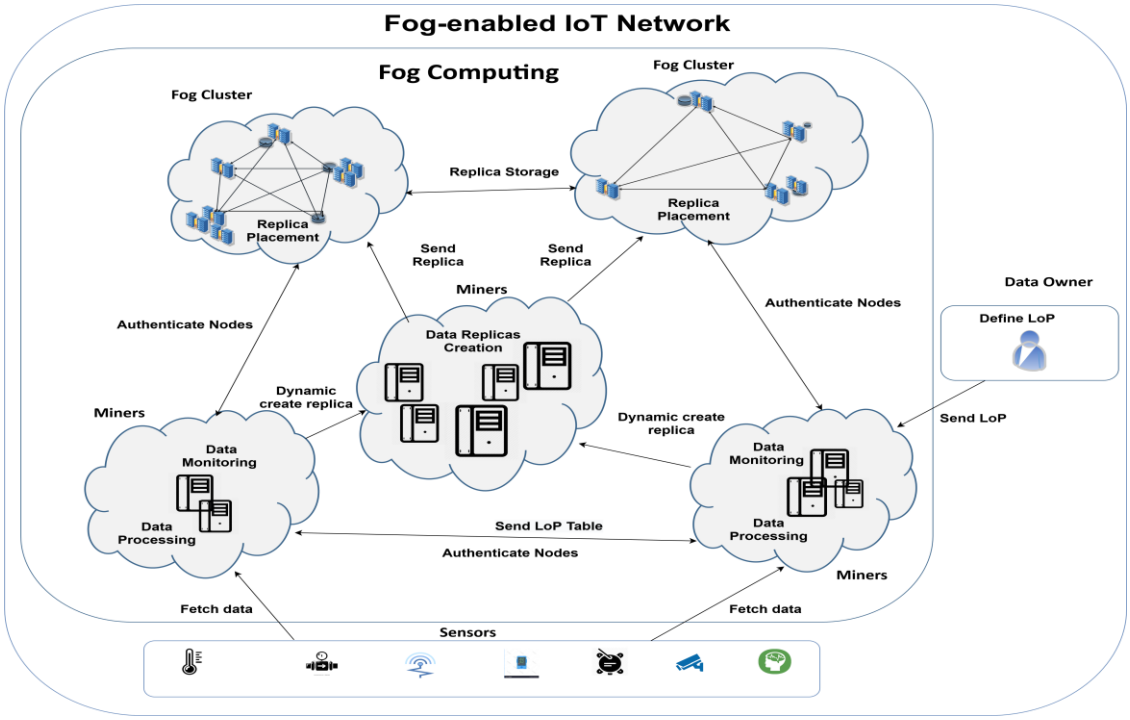


Figure 29 Fog-enabled IoT Network

- Fog layer 1 is responsible for processing and monitoring the computation services in fog layers architecture. The computation services include Level of Privacy (LoP) table for data, encryption, division, and replica creation. The fog layer 1 consists of multiple fog nodes represented as miner nodes, which are deployed at the edge of the IoT sensors. Miner nodes perform the computation services as well as authenticates and communicates with data-owner and fog nodes in fog layer 2.

  A data-owner communicates with a miner node in fog layer 1 to define LoP. The data-owner can be an individual or a device, which owns sensors' generated data to be transmitted to miner nodes. The data owner has the authority to define the LoP for data generated on different sensors. LoP is considered in creating data replicas at miner nodes.
- The data replicas created at miner nodes are transmitted to fog nodes in fog layer 2. The fog layer 2 represents the cluster of fog nodes that provides the services including storage and availability of data replicas to end-user/ cloud devices.

3.2. Adversary Model

Following previous work [42], we consider the miner nodes in the fog layer 1 as Services Provider (SP) and fully trusted entities, similar to Liu *et al.* [144] provided threat model for SP. We also assume that miner nodes will not collude with fog nodes in fog layer 2. We consider fog nodes in fog layer 2 as honest-but-curious nodes, which means that they will store the data replicas and strictly conform to the replica placement protocol. Still, fog nodes may try to infer the privacy of replica based on the information of the data block that fog node is holding. Security

threats that occur in two aspects to compromise the privacy of data are internal and external threats.

Specifically, internal threats could be from any fog node that is curious to know the information inside a block of data. The information may include personal data or the location of a sensor node (e.g, sensor id, health status, source, destination, data-owner id). We also assume that in fog layers 1 and 2, there is no node collision among fog nodes, which is an assumption used in cryptographic threat models [308]. On the other hand, external threats may occur when an adversary eavesdrop on a communication link between miner and fog nodes to intercept data. Furthermore, an adversary may disguise as an authorized or legitimate fog node to modify the data.

3.3. Data Replica Creation Scheme based on Level of Privacy and Service Capacity

In this subsection, we describe a replica creation scheme for data blocks in fog layer 1. The scheme is based on the data owner's defined LoP and service capacity of miner nodes. Figure 30 shows the architecture of the proposed scheme.

Based on the divide-and-conquer Algorithm [42], each data packet coming from a sensor is processed at miners before data replica creation. The processing at miners includes data encryption and data division. Data is encrypted using AES (Advanced Encryption Standards)

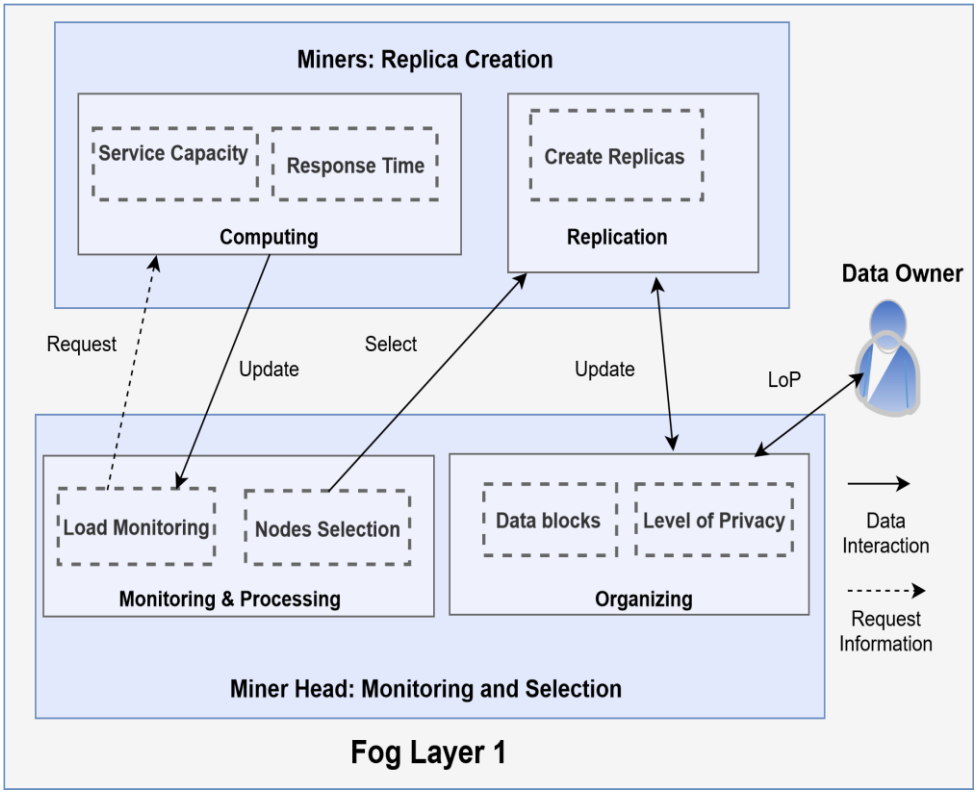


Figure 30 The Architecture of Replica Creation Scheme

technique. After the encryption, the data file is divided into blocks based on the Division Factor (D.F) derived from LoP in [42]. Then the replica creation scheme for each data block is carried out as shown in Figure 30.

In replica creation architecture, the miner head node is responsible to monitor, authenticate and select miner nodes for replica creation. The selection of miner nodes for replica creation is based on a service capacity factor. For decision making, the miner head checks the service capacity of each miner node and then selects an appropriate miner node for replica creation.

The selection process is discussed in detail in sub-section 1. The selected miner node then computes the total number of replicas for a data block using a formula that includes LoP and other factors. The replica creation process is discussed in subsection 2. In short, the replica creation scheme selects an appropriate miner node for replica creation and creates replicas dynamically considering the computing factors. An Algorithm for replica node selection and creation is represented in Algorithm 8.

Algorithm 8: Replica Creation

Input: $\boldsymbol{LoP_{max}}, \boldsymbol{LoP}, \boldsymbol{bl}, \boldsymbol{NumMiner}$ // maximum privacy value, data blocks' Level of privacy, data blocks and number of total miner nodes

Output: $\boldsymbol{Nr}, \boldsymbol{bl_i}$ // number of replicas for data blocks and replica blocks

Initialize $\boldsymbol{m_{rate}}$: response request rate of each miner node

Initialize $\boldsymbol{SB_{ij}}$: **Size of block transmitted btw miners or miner and sensor node**

$\boldsymbol{//Sensor_{node}}$ Represents a sensor node

// $\boldsymbol{Miner_{node}}$ Represents a miner node

1. $\boldsymbol{Compute}(\boldsymbol{Miner_{node}}, \boldsymbol{m_{Sc}})$; //compute the service capacity for each miner node
2. for $\boldsymbol{i \leftarrow 1\ to\ len\ (Miner_{node})\ do}$
3. for j← $\boldsymbol{1\ to\ len\ (Sensor_{node})\ do}$
4. if $\boldsymbol{i \neq j}$ then
5. $\boldsymbol{m_{Sc_i} = \frac{1}{m_{t_{ij}} \times m_{l_i}}}$; // from Equation 3.1
6. $\boldsymbol{m_{t_{ij}} = \frac{SB_{ij}}{NB_{ij}}}$ & $\boldsymbol{m_{l_i} = \frac{NR_i}{C_i}}$;
7. $\boldsymbol{m_{Sc_i} = \frac{C_i \times NB_{ij}}{SB_{ij} \times NR_i}}$;
8. $\boldsymbol{m_{Sc}[i] = m_{Sc_i}}$;
9. else $\boldsymbol{m_{sc_i} = 0}$;
10. end for
11. end for
12. Return $\boldsymbol{m_{Sc}}$;
13. *Sort* ($\boldsymbol{Miner_{node}}, \boldsymbol{m_{Sc}}$); //sort miner nodes by service capacity
14. Select the miner node with the highest $\boldsymbol{m_{Sc}}$
15. *Access* ($\boldsymbol{LoP}, \boldsymbol{b}$); // Level of privacy and data blocks request by selected miner node

16. Compute (**b**, $\boldsymbol{Nr}$); //Compute the number of replicas for data blocks
17. for k← $\boldsymbol{1\ to\ len\ (b)\ do}$
18. $\boldsymbol{Nr_k = (\alpha\ \times LoP_k/2\ \times LoP_k)/LoP_{max} + \frac{\beta \times \frac{d_i}{td_i} * L_i}{\frac{\sum_{i=1}^{n} L_i}{NumMiner}}}$;
19. for $\boldsymbol{l \leftarrow 1\ to\ (Nr_k)\ do}$
20. Create $((\boldsymbol{b_{k0}}, \ldots, \boldsymbol{b_{kl-1}}), \boldsymbol{Nr_k})$; // create block's replicas based on the number of replicas $\boldsymbol{Nr_k}$
21. end for
22. end for
23. Return $(\boldsymbol{b_{k0}}, \ldots, \boldsymbol{b_{k_{p-1}}})$;

3.3.1. Miner Nodes Selection for Data Replica Creation

In this part, we discuss a preliminary selection of a miner node for the replica creation scheme. Firstly, the miner head is selected for monitoring the service capacity of miner nodes. Then the service capacity of a miner node is computed by the miner head. The service capacity is based on the processing load and response time. Then, the miner head considers a target miner node with the highest service capacity for a higher number of replica creation.

We assume that the miner head selection is based on the previous data processing load on the miner node. The processing includes the encryption of data files and the division of data files into blocks. The miner node with fewer data processing loads as compared to other miner nodes is considered as a miner head node.

The miner head checks the service capacity of miner nodes. The Service capacity $\boldsymbol{m_{Sc_i}}$ indicates the serviceability of the i^{th} miner node ($1 \leq i \leq n$), where n is the total number of miner nodes in a network. The serviceability includes the load $\boldsymbol{m_{l_i}}$ of the node and response time $\boldsymbol{m_{t_{ij}}}$. The load $\boldsymbol{m_{l_i}}$represents the load condition of the i^{th} miner node and the response time $\boldsymbol{m_{t_{ij}}}$ represents the time required by the i^{th} miner node to transfer or receive the j^{th} sensor/ miner node ($1 \leq j \leq m$) data, where m is the total number of sensor nodes or another miner j^{th} node ($1 \leq j \leq n$) for $i \neq j$. Then the service capacity $\boldsymbol{m_{Sc_i}}$ of the i^{th} miner node ($1 \leq i \leq n$) based on formulas defined in [62, 309] becomes

$$\boldsymbol{m_{Sc_i} = \frac{1}{m_{l_i} \times m_{t_{ij}}}} \qquad (5.1)$$

The load condition $\boldsymbol{m_{l_i}}$ of miner nodes is measured by the number of node requests $\boldsymbol{NR_i}$ processed by the miner node at a current time divided by the number of cells $\boldsymbol{C_i}$ present in the miner node for processing requests.

The transmission time $\boldsymbol{m_{t_{ij}}}$ of a miner node is the transmission time consumed for data and authentication process between the $i^{th}\ and\ j^{th}$ nodes. The $\boldsymbol{m_{t_{ij}}}$ is lower if the distance between

the miner nodes or the miner and sensor nodes is less as compared to other nodes' distance. Then the $\boldsymbol{m_{Sc_i}}$ becomes

$$\boldsymbol{m_{Sc_i}} = \frac{\boldsymbol{C_i} \times \boldsymbol{NB_{ij}}}{\boldsymbol{NR_i} \times \boldsymbol{SB_{ij}}} \quad \text{where } i \neq j \qquad (5.2)$$

The miner head node checks the $\boldsymbol{m_{Sc_i}}$ of each miner node and selects a target miner node. The target node with the highest $\boldsymbol{m_{Sc_i}}$value is selected for a higher number of replica creations. For each iteration of replica creation, the $\boldsymbol{m_{Sc_i}}$ of each miner node is computed to check the current service capacity of a miner node to participate in the next round of replica creation. The computation of the $\boldsymbol{m_{Sc_i}}$ is completed once the total number of replicas of a block is created at the target miner node.

3.3.2. Data Replica Creation at Target Miner Node

In this part, we discuss the creation of data block replicas at the target miner node. First, the minimum number of replicas based on the LoP is computed. Then, considering a minimum number of replicas, LoP, load, and influencing factors, the total number of replicas is calculated. Finally, data replica creation is performed based on the computed total number of replicas.

We assume that $\boldsymbol{LoP_l}$ represents the level of privacy defined by a data-owner for a sensor-generated data $\boldsymbol{file_l}$ ($1 \leq l \leq$ d), where d is the total number of data generated at the sensor. Then, the level of privacy of the encrypted data block $\boldsymbol{b_k}$ belonging to $\boldsymbol{file_l}$ becomes $\boldsymbol{LoP_k}$. For replica creation, the miner head node forwards the encrypted data block $\boldsymbol{b_k}$ and $\boldsymbol{LoP_k}$ to targeted miner node.

Using $\boldsymbol{LoP_k}$, the targeted miner node calculates the minimum number of replicas as $\boldsymbol{nr_{min_k}} = \frac{\boldsymbol{LoP_k}}{\boldsymbol{2}}$. The computed number of replicas $\boldsymbol{nr_{min_k}}$ of data block $\boldsymbol{b_k}$ are the minimum replicas required for the reliability of data block $\boldsymbol{b_k}$ belonging to file $\boldsymbol{file_l}$

The minimum number of replicas is then further used in calculating the total number of replicas for the data block $\boldsymbol{b_k}$, as shown in Equation 5.3. $\boldsymbol{Nr_k}$ represents the maximum number of replicas required for the reliability of data block $\boldsymbol{b_k}$.

$$\boldsymbol{Nr_k} = \left\lceil \frac{\boldsymbol{\alpha} * \boldsymbol{nr_{min_k}} * \boldsymbol{LoP_k}}{\boldsymbol{LoP_{max}}} + \frac{\boldsymbol{\beta} * \boldsymbol{tl_i}}{\boldsymbol{l_{avg}}} \right\rceil, \qquad (5.3)$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are influencing factors, $\boldsymbol{tl_i}$ is the total load of the i^{th} miner node and $\boldsymbol{l_{avg}}$ is the average load of miners in fog layer 1.

The total number of replicas $\boldsymbol{Nr_k}$ is directly affected by the influencing factors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, which are correlation coefficients set according to the network connectivity and communication of miners and fog nodes in a network for experimental analysis [62, 309]. The influencing factor $\boldsymbol{\alpha}$ represents the total ratio of nodes communicating in the network with the i^{th} miner node. The influencing factor $\boldsymbol{\beta}$ is the type of connectivity connection for i^{th} miner node.

The total load $\boldsymbol{tl_i}$ is derived from the formula proposed in [62, 309], which considers disk utilization denoted as $\boldsymbol{ds_i}$ and service load is $\boldsymbol{L_i}$ of the i^{th} miner node. Then the total load is $\boldsymbol{tl_i = ds_i * L_i}$. The disk utilization $\boldsymbol{ds_i}$ in the formula is the ratio of disk size of the i^{th} miner node to the total disk space of the i^{th} miner node, and the calculation method is $\boldsymbol{ds_i = \frac{d_i}{td_i}}$.

The average load $\boldsymbol{l_{avg}}$ of the miner nodes in the fog layer 1 is the sum of loads of all miners to the total number of miners, represented as $\boldsymbol{l_{avg} = \frac{\sum_{i=1}^{n} L_i}{Num\ of\ Nodes}}$.

In the data replica creation scheme, we consider data-owner defined LoP for each $\boldsymbol{file_l}$ data replica creation instead of file popularity and access heat used in [62, 309, 310]. The reason for using LoP is to provide authority to data-owners to define a privacy level for their data files. Based on the LoP, data files are divided into blocks as the calculation in [42] and then data blocks are replicated. In other words, LoP's is used to identify the popularity of $\boldsymbol{file_l}$ in terms of privacy instead of access heat.

Equation 5.3 becomes:

$$\boldsymbol{Nr_k = \left\lceil \frac{\alpha * nr_{min_k} * LoP_k}{LoP_{max}} + \frac{\beta * \frac{d_i}{td_i} * L_i}{\frac{\sum_{i=1}^{n} L_i}{Num\ of\ Nodes}} \right\rceil} \qquad (5.4)$$

After calculating the total number of replicas for a block $\boldsymbol{b_k}$ belonging to $\boldsymbol{file_l}$ at a targeted miner node, the replicas for a block $\boldsymbol{b_k}$ can be created based on $\boldsymbol{Nr_k}$, as $\boldsymbol{b_k} = (\boldsymbol{b_{k_0}}, \dots, \boldsymbol{b_{k_{p-1}}})$ where $(1 \leq p \leq \boldsymbol{Nr_k})$

3.4. Data Replica Placement Scheme Based on Level of Priority

In this subsection, we propose a replica placement scheme based on a level of priority of fog nodes in fog layer 2. The division factors based on the number of blocks and service capacity of fog nodes are used to measure the priority level of fog nodes. A fog node with the highest priority is selected as a suitable node for the placement of a replica. The level of priority for each fog node is represented in a priority table, which includes a comparison of service capacity $\boldsymbol{m_{Sc_i}}$ and data blocks among all fog nodes. Figure 31 shows the architecture of our replica placement scheme and the Algorithm for replica placement is shown in Algorithm 9.

Algorithm 9: Replica Placement

Input: $\boldsymbol{LoP}, \boldsymbol{D.F}, \boldsymbol{f}, \boldsymbol{bl}, \boldsymbol{bl_{k0}}, .., \boldsymbol{bl_{kl-1}}, \boldsymbol{FogNodes}$ // data blocks' Level of privacy, division factor, data file, data blocks, replicas of data blocks, and number of total fog nodes

Output: $\boldsymbol{dest_{fog_{node}}}$

Initialize $\boldsymbol{f_{pri}}$: **Fog node's priority initialise**

Initialize $\boldsymbol{SB_{ij}}$: **Size of block transmitted between miner node and fog node**

// $\boldsymbol{Fog_{node}}$ Represents a fog node

// $\boldsymbol{Miner_{node}}$ Represents a miner node

1. $\boldsymbol{Compute(Fog_{node}, m_{Sc})}$; //similar to Algorithm 8, //compute the service capacity
2. for $\boldsymbol{i \leftarrow 1\ to\ len\ (Fog_{node})\ do}$
3. for j← $\boldsymbol{1\ to\ len\ (Miner_{node})\ do}$
4. if $\boldsymbol{i \neq j}$ then
5. Repeat steps 5 to 11 of Algorithm 8
6. end for
7. Return $\boldsymbol{m_{Sc}}$;
8. Compute $(\boldsymbol{Fog_{node}},\ \boldsymbol{B_i},\ \boldsymbol{D.F})$; // For each fog nodes, the miner node computes the division factor based on the number of blocks
9. for k← $\boldsymbol{1\ to\ len\ (file)\ do}$
10. $\boldsymbol{B_i = Search\ (Num\ \left(\sum_{k=1}^{len(file)} B_k\right)}$); // calculates the total number of blocks of the kth file at each fog node
11. for $\boldsymbol{LoP \geq 4}$ do
12. if $\sqrt{\boldsymbol{DF_l}} > \boldsymbol{B_i}$ then // checks the lth file blocks at ith fog nodes is less than the Division factor for the lth file
13. $\boldsymbol{f_{pri} \leftarrow Fog_{node_i}}$ // keep the ith fog node in the priority table
14. else $\boldsymbol{f_{pri} \leftarrow 0}$
15. Sort ($\boldsymbol{f_{pri}, m_{Sc}}$); //sort the priority fog nodes by service capacity
16. Select Priority fog node with highest $\boldsymbol{m_{Sc}}$ for replica placement
17. $\boldsymbol{dest\_fog\_node \leftarrow Fog_{node_i}}$
18. end for
19. for $\boldsymbol{LoP = 3}$ do
20. if $\frac{\boldsymbol{DF_l}}{\boldsymbol{2}} > \boldsymbol{B_i}$ then // checks the lth file blocks at ith fog nodes is less than the Division factor for the lth file
21. $\boldsymbol{f_{pri} \leftarrow Fog_{node_i}}$ // keep the ith fog node in the priority table
22. Repeat step 18 to 21
23. end for
24. for $\boldsymbol{LoP < 3}$ do
25. if $\boldsymbol{DF_l > B_i}$ then // checks the lth file blocks at ith fog nodes is less than the Division factor for the lth file
26. $\boldsymbol{f_{pri} \leftarrow Fog_{node_i}}$ // keep the ith fog node in priority table
27. Repeat step 18 to 21

28. end for
29. Update (f_{pri}, m_{Sc}); //updates the values after each replica placement fog node searched
30. For $l \leftarrow 1$ *to* $len\ (bl_k)$ *do*
31. $dest_{fog_{node}} \leftarrow f_{pri}$;
32. Get ($bl_{k0}, .., bl_{kl-1}$); //get the block replicas
33. Insert ($dest_{fog_{node}}, bl_{kl-1}$); //sending replica blocks to selected fog nodes $dest_{fog_{node}}$
34. Repeat steps 13 to 33 unless all replicas placed in $dest_{fog_{node}}$
35. end for
36. end for

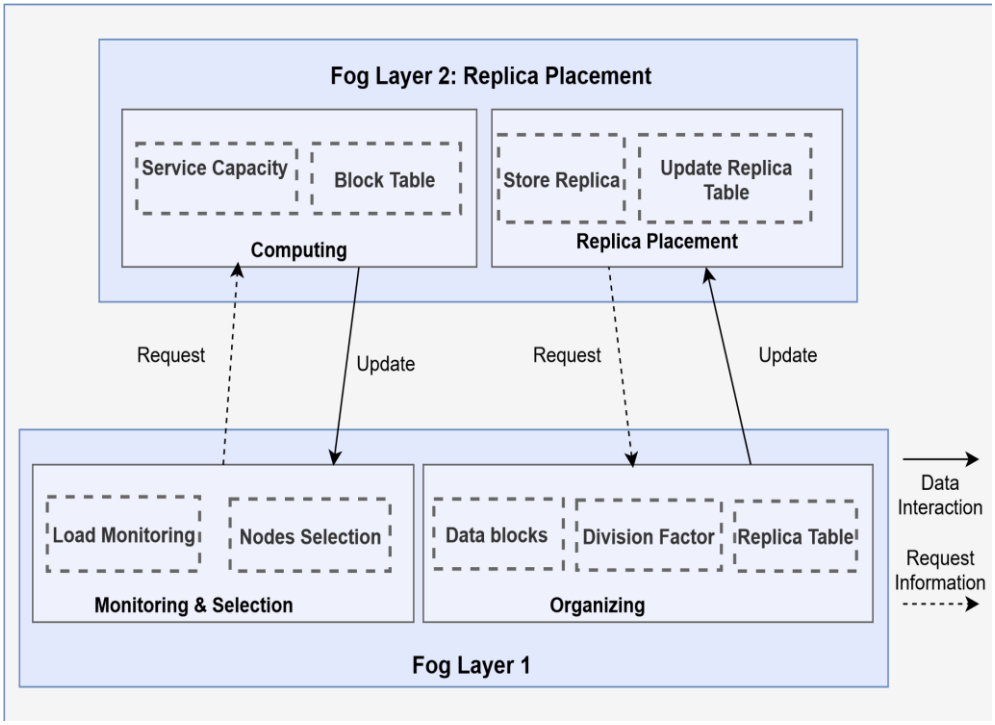


Figure 31 Architecture of Replica Placement

3.4.1. Data Replica Placement at Fog nodes

After the creation of replicas of data blocks in fog layer 1, the data replica placement scheme considers the placement of replicas in a distributive manner in fog layer 2. For replica placement, first, the miner head monitors the service capacity m_{Sc_i} and the total number of blocks B_i of a file $file_l$ residing at fog nodes in fog layer 2. Then, the miner head compares the B_i with D.F of LoP for a $file_l$. Based on the comparison, the priority of each fog node is set by a miner head for replica placement. The flow of the processes at the miner head for replica placement is illustrated in Figure 32 and discussed in detail below.

The miner head checks the total number of blocks B_i at each fog node by counting the number of blocks B_k, represent as $B_i = Num\left(\sum_{k=1}^{len(file)} B_k\right)$. Then the miner head node compares the blocks B_i for each fog node with the D.F based on the LoP setting for a file $file_l$. For each LoP levels, the comparison of B_i with D.F varies, as discussed below.

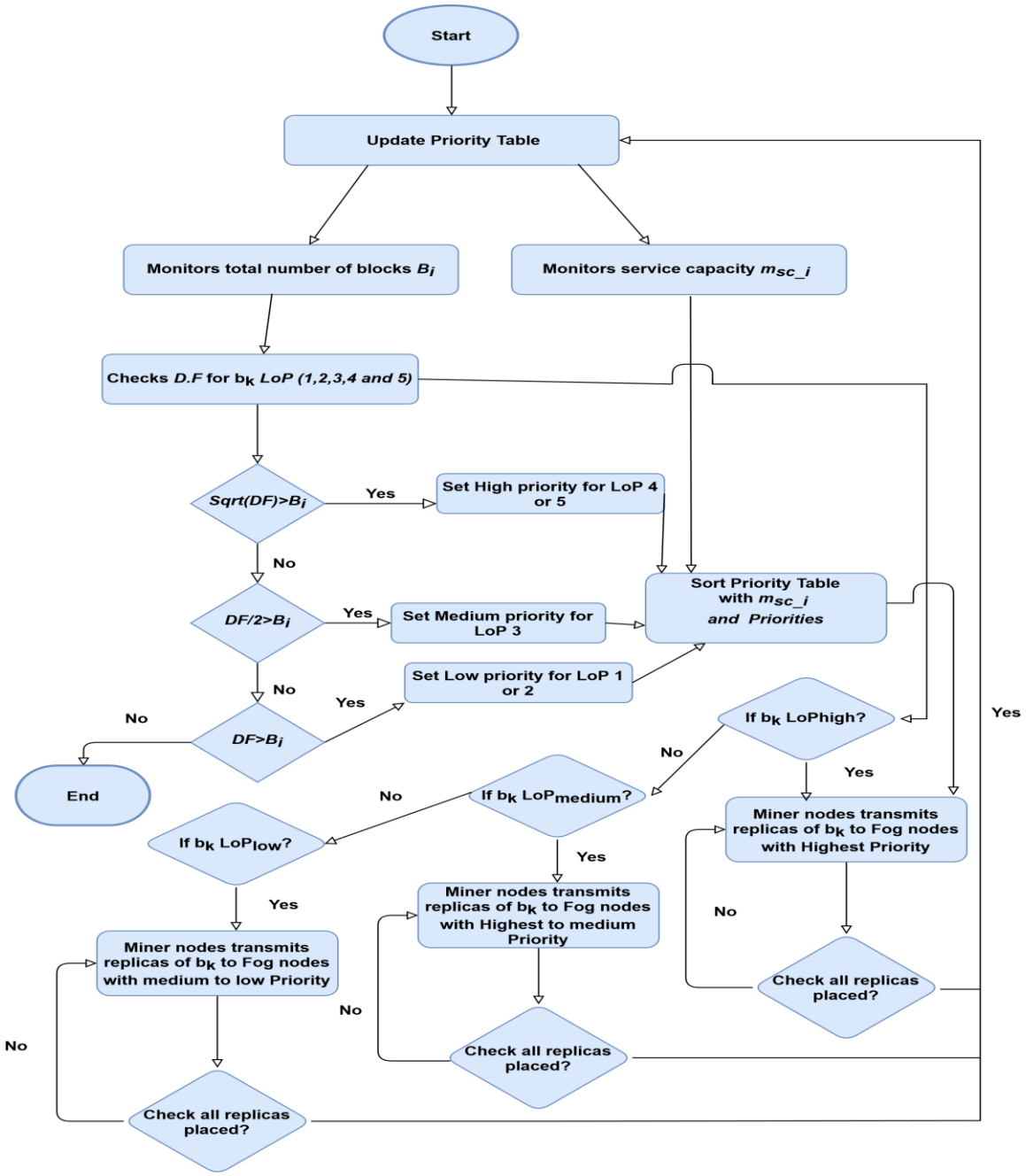


Figure 32 Flowchart for Data Placement Processes at Miner head

1. For $\boldsymbol{LoP_l}$= high, i.e. LoP is 4 or 5, the miner head checks whether the B_i on the k^{th} fog node is less than the squared root of D.F or not, such as $\lfloor\sqrt{DF_k} > B_i$.
2. For $\boldsymbol{LoP_l}$= medium, i.e. LoP is 3; the miner head checks whether the B_i on the k^{th} fog node is less than half of D.F or not, such as $\frac{DF_k}{2} > B_i$.
3. For $\boldsymbol{LoP_l}$=low, i.e. LoP is 1 or 2; the miner head checks whether the B_i on k^{th} fog node is less than the D.F or not, such as $DF_k > B_i$.

In the first-round, the miner head node set the priority of each fog node based on the B_i comparison with D.F. If a value of B_i is less than the D.F then the priority of a fog node is set as an initial highest priority node, i.e. k=1. The fog node also qualifies for the next round.

Otherwise, the fog node is assigned a medium or low priority. For the medium priority, i.e. 1< k<len (fog node), B_i value in comparison with D.F is nominally high, whereas for the low priority, i.e. k=len (fog node), B_i value is significantly high. For low priority, there is a significantly high difference between B_i and D.F.

In a second round, the miner head considers the $\boldsymbol{m_{Sc_i}}$ of each fog node and adds $\boldsymbol{m_{Sc_i}}$ to the priority table. Miner head also sorts the priority level with $\boldsymbol{m_{Sc_i}}$ and B_i in ascending order for each fog node. The highest priority number assigned to the fog node is due to higher service capacity and a smaller number of $\boldsymbol{file_l}$ blocks B_i as compared with D.F.

After setting up the priorities of each fog node, the miner head requests miner nodes containing replicas of $B_k$to send the number of replicas of a block $\boldsymbol{b_k}$ to targeted fog node. Once a successful replica placement to a targeted fog node is completed, the $\boldsymbol{m_{Sc_i}}$ of the targeted fog node is calculated again and updated with the newest priority into the priority table. To preserve the privacy of replica blocks at fog layer 2, the miner head also considers the following placement procedures according to LoP defined for $\boldsymbol{b_k}$.

1. For $\boldsymbol{LoP_{high}}$ defined for the data block $\boldsymbol{b_k}$, the miner head checks the fog node that has the highest priority in the priority table and requests the miner node containing a replica of B_k to send a replica to the targeted fog node. This procedure repeats until all replicas of a block $\boldsymbol{b_k}$ are placed at all targeted fog nodes with the highest priority.
2. For $\boldsymbol{LoP_{med}}$ defined for the data block $\boldsymbol{b_k}$, the miner head searches and selects the fog nodes as the targeted fog nodes that have the highest priority in a table for replicas placement. In the case of the fog nodes with the highest priority are not available for placement of the remaining replicas of a block $\boldsymbol{b_k}$, the miner head then continues the search and selects the fog nodes with medium priority. The miners are requested to place the remaining replicas on the fog nodes with medium priority.
3. This procedure repeats itself until all replicas of a block $\boldsymbol{b_k}$ are placed at targeted fog nodes with highest to medium priority.
4. For $\boldsymbol{LoP_{low}}$ defined for a data block $\boldsymbol{b_k}$, the miner head selects the fog nodes that have medium priority in the priority table. If the number of replicas exceeds the number of fog nodes with medium priority for replica placement, then the priority table is searched for the lowest priority nodes. This procedure repeats itself until all replicas of a block $\boldsymbol{b_k}$ are placed at targeted highest to medium priority fog nodes.

3.5. Time Complexity Analysis of the Proposed Schemes

This Section provides the time complexity analysis of our proposed schemes in detail as follows.

3.5.1. Time complexity Analysis of Data Replica Creation Scheme

Initialization and computation are two main operations involved in Algorithm 8 for the data replica creation scheme. In the initialization operation, the time complexity of the response request rate m_{rate} and size of the blocks SB_{ij} is constant O (*1*). The time complexity of computation operations is analyzed for service capacity, the number of replicas, sorting, and replica creation. The complexity analysis of these computation operations is given below.

1. For the service capacity m_{sc_i} operation as mentioned in Section 3.3, computation of the load m_{l_i} is for n number of miner nodes and computation of the response time $m_{t_{ij}}$ is for n number of miner nodes and m number of other miner nodes and sensor nodes. Given $n\ and\ m$ nodes, the complexity to compute the service capacity m_{sc_i}becomes O ($n \times m$).
2. From Equation 5.3, we can deduce that the time analysis of the number of replicas Nr_k is constant O (1) for the execution of predefined LoP and value of α and O (n) is for computing load of n miners. Considering the fastest-growing term n, the total time complexity becomes O (n).
3. The time taken to sort miner nodes based on the service capacity m_{sc_i} is O (1). The complexity of replica creation is also constant as there is only one execution plan for the replica creation based on computed Nr_k.

Thus, the overall time complexity of the proposed data replica creation scheme is summarized as O ($n \times m$).

3.5.2. Time complexity Analysis of Data Replica Placement Scheme

Similar to Algorithm 8, the operations considered in Algorithm 9 for data replica placement are initialization and computation. The time complexity of initialization of fog node's priority f_{pri} and size of the blocks SB_{ij} is constant O (*1*). The time complexity for computation operation involves the time taken for computing service capacity m_{sc_i}, the total number of blocks B_i , sorting, update, and insertion. For the computation of service capacity $\mathrm{m_{sc_i}}$, the time complexity is the same as Algorithm 8 i.e. O ($n \times m$). The time taken for computation of the total number of blocks B_i depends on the summation of k blocks of a file $file_l$. Given k blocks, the time complexity for B_i becomes O (k).

Likewise sorting operation of Algorithm 8, sorting in Algorithm 9 also takes O (*1*) time to sort f_{pri} and m_{sc_i}. The time taken to compute the update and insert operation is also O (*1*) as there is a constant execution for updating the priority table and inserting replica blocks. Considering fastest-growing term $n, m \text{ and } k$ from operations, the time complexity of data replica placement is summarized as O ($(n \times m) + k$).

4. Simulation and Evaluation

In this section, we first introduce our simulation setup and then provide different network scenarios that are used to evaluate our proposed schemes. The aspects of data replica utilization, storage usage, and the correlation between LoP and the number of replicas, and between the number of replicas and the service capacity are used for evaluation. Second, we provide a privacy analysis of our proposed schemes. Third, we provide a time series analysis of our replica creation and replica placement schemes. Then we discuss a performance analysis based on storage, computational and replica utilization of the proposed strategies vs other existing schemes.

4.1. Simulation Setup

Our simulations are conducted with Linux based system on a P.C. with Intel (R) core (i7), RAM 16.0 GB, and CPU 3.40 GHz. We develop Algorithms of our replica creation and placement schemes using Network Simulator (NS3) in C++ and Python languages. We perform extensive simulations to analyze and evaluate the performance of our Algorithms. The number of sensor devices, miner nodes, and fog nodes varies from 2 to 100. Each sensor contains a file with a data size ranging from 1KB to 100MB. We consider different scenarios with varying data sizes and the number of nodes. In a basic scenario, we have five sensor devices (s_1, s_2, s_3, s_4, s_5), one data-owner node and ten fog nodes. Out of ten fog nodes, three of them are miner nodes ($m_0, ---, m_3$) for replica creation and one node is a miner head node. Miner nodes are close to sensors, and six of fog nodes ($f_0, ---, f_5$) are considered for data replica placement.

4.2. Experimental Results and Analysis

To evaluate our proposed Algorithms of replica creation and replica placement strategies, we analyze how the performance of Algorithms is affected by varying the parameters that influence the simulation results. Parameters that exert influence on the replica creation and replica placement Algorithms are provided in Table 14.

Table 14 Sensitive Parameters

Parameter	Definition	Parameter	Definition
$\boldsymbol{m_{Sc}}$	Service Capacity of each node		
$\boldsymbol{LoP}$	Level of Privacy of each data block	f_{pri}	fog node priority
$\boldsymbol{\alpha}$	the ratio of nodes communicating in the network with the miner node		

4.2.1. Privacy Analysis of the Proposed Schemes

We consider Shacham and Chen *et al.* privacy models [63, 64] to perform the formal privacy analysis of the proposed strategies. The privacy analysis is based on the following definitions and proofs to fulfill the privacy preservation of the proposed strategies.

Definition 1. The proposed models achieve data privacy preservation in the following cases.

1) The data replica placed at fog nodes do not leak any private information about the original data.
2) For a probabilistic polynomial time (PPT), it is computationally hard for an adversary who acts like a fog node to extract private information about data blocks from other fog nodes in a fog layer 2.

Theorem 1. *In the random oracle model, if data is CPA-secure, the privacy preservation of data is achieved as defined in Definition 1.*

Proof. Before data replica creation, the data coming from sensor nodes is encrypted using the Advanced Encryption Algorithm (AES). The encrypted data is also divided into blocks based on the LoP setting. Then the block's replicas are created using the LoP setting and replicas are placed on fog nodes in a distributive manner. Replica's placement in a distributive manner is carried out using the priority level of a fog node, which is also based on LoP. The participating fog nodes in replicas placement cannot learn the private information of the data block as long as they do not have blocks hash and symmetric keys information. To preserve data in blocks as private as possible, the symmetric keys for the decryption of data blocks are not distributed by a miner head node to fog nodes in fog layer 2. The fog nodes only store replicas of blocks that do not reveal any information as blocks and their replicas are an encrypted division of data.

To prove privacy preservation of data block's replicas at fog nodes, we assume that an adversary A acts like a fog node FN_k in a random oracle $h(T)$. A holds a replica of data block $\boldsymbol{b_k}$. For revealing information inside a replica of $\boldsymbol{b_k}$, A has to provide an output of correct hashes H for next and previous blocks (i.e. $\boldsymbol{b_k}\ and\ \boldsymbol{b_{k-1}}$) to complete blocks chain such as $file_l = (b_k + b_{k-1})$. A also has to output symmetric key ky to decrypt the blocks chain $file_l$. We assume that a PPT A randomly computes H' and ky' for revealing blocks chain $file_{l'}$ in $h(T)$. The probability that an A computes are equal to original H and ky for revealing $file_l$ is no more than negligible probability ε

$$Pr\left[A\left(Dec_{ky'}(file_l{}' = H'(b_k + b_{k-1})\right) = \left(Dec_{ky}(file_l = H(b_k + b_{k-1})\right)\right] < \varepsilon.$$

Hence A knows nothing about the miner head computed H and ky for revealing information inside a replica of data block $\boldsymbol{b_k}$. A has to also eavesdrop on a high number of fog nodes for

compromising the maximum number of replicas of blocks $\boldsymbol{b_k}$ and $\boldsymbol{b_{k-1}}$ in order to decrypt data inside blockchains E^k. Otherwise, compromise of any replica of $\boldsymbol{b_{k-1}}$ not all $\boldsymbol{b_k}$ will not reveal much information about the whole blocks chain E^k.

Definition 2. The proposed models guarantee the replica reliability in which the original data block can be recovered from replica blocks placed at fog nodes.

Proof. The number of replicas is created by miner nodes and distributed among fog nodes to keep the data blocks $\boldsymbol{b_k}$ reliable in case of any fog node FN_k failure/compromise. The setting of priority level makes sure that the FN_k does not contain a high number of blocks b_k of the $file_l$ as compared to the D.F for each LoP. The reason behind this is to keep $\boldsymbol{b_k}$ and the replicas secure and to reduce maximum subsets of $\boldsymbol{b_k}$ belonging to $\boldsymbol{file_l}$ at same fog node that may expose too much information inside $\boldsymbol{file_l}$.

4.2.2. Performance Analysis for Data Replica Creation

In our proposed replica creation scheme, the $\boldsymbol{m_{sc}}$, $\boldsymbol{LoP}$, $\boldsymbol{\alpha}$, and $\boldsymbol{f_{pri}}$ are vital parameters influencing the performance of the scheme. In this subsection, we discuss the detailed analysis of each parameter.

1) Service Capacity (m_{sc}):

The service capacity of a node has an impact on the creation of a replica for a data block. A miner head node selects a miner node for replica creation depending upon the load capabilities and transmission time of a miner node. In principle, a miner node with the lowest amount of load and transmission time would be the best suitable miner node for processing a higher number of replicas. To depict the number of replicas creation on each miner node, Figure 33 shows the box plot for a relationship between the number of replicas and service capacity (%). The service capacity has a positive correlation with the number of replicas. With the increase in service

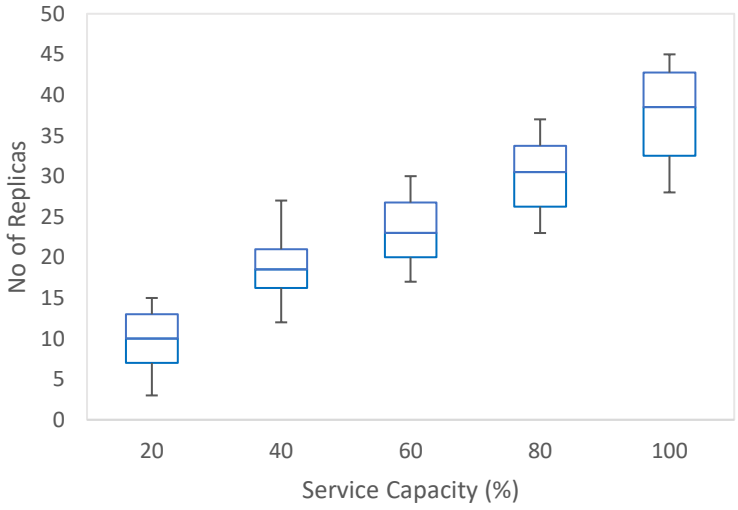


Figure 33 Number of Replica Vs Service Capacity

capacity on average from 20 to 100%, the average number of replicas created on miner nodes becomes higher, i.e. from 2 to 45. Replica creation is directly impacted by the service capacity of a miner node, the higher the service capacity of a node, the high number of blocks' replicas processed on a miner node.

2) Level of Privacy (LoP):

LoP has a higher impact on $\boldsymbol{Nr_k}$ as compared to other parameters such as average load, total load, $\boldsymbol{\alpha}\ and\ \boldsymbol{\beta}$ factors in Equation 5.3. The minimum level of replicas $\boldsymbol{nr_{min_k}}$ is also formulated based on LoP. The LoP identifies the level of data privacy, the higher the level, the higher number of block replicas are created to keep the data reliable for recovery in case of a fog node failure or malicious activity in fog layer 2. Based on the basic scenario as discussed in subsection 4.1, Table 15 presents a range of the total number of replicas $\boldsymbol{Nr_k}$ for each block w.r.t LoP.

It can be depicted from a Table that the LoP as 5 has the maximum number of replicas creation i.e. nine. The highest privacy level 5 is used for the most private data block $\boldsymbol{b_k}$, which means a block needs to be kept secure with more data reliability as compared to the other LoPs. The block $\boldsymbol{b_k}$ with LoP 4 is considered as the second-highest private data blocks, creating on average seven replicas. On a contrary, LoPs as 1 and 2 are the lower privacy levels with a smaller number of replicas creation on average one, two, or three, which is less than other LoPs.

Table 15 Level of Privacy Vs Total Number of Replicas

$\boldsymbol{LoP}$	$\boldsymbol{Nr_k}$ **for each block** $\boldsymbol{B_k}$
1	$1, 2$
2	$2, 3$
3	$2, -4$
4	$3, --, 7$
5	$4, - - ,9$

3) Ratio of communicating nodes (α):

We evaluate the impact of influencing factor $\boldsymbol{\alpha}$ on $\boldsymbol{Nr_k}$, which is the ratio of fog/miner nodes communicating with a targeted miner node. The value of α varies as it depends upon the number of fog/miner nodes corresponding to a target miner node out of total miner nodes.

For the basic scenario, α varies from a minimum value of 0.25 to a maximum value of 2.25. In the network, the minimum and maximum values are the variations of a ratio of the total number of miner/fog nodes interacting with the targeted miner node to the total number of miner nodes. At least one miner head node communicates with a targeted miner node, the total number of miner nodes in the basic scenario is four, so the ratio becomes 1:4. Whereas, at most

3 miner nodes and 6 fog nodes are interacting with the targeted miner node, so the ratio becomes 9:4.

At the start of an execution cycle for replica creation, the α's value is 0.25 as only the miner head node interacts with a targeted miner node to check its service capacity. Then, the value of α increases after the first replica creation $\boldsymbol{Nr_k}$. The increase in α value is due to an increase of interaction between fog and miner nodes. α value causes a slight increase in $\boldsymbol{Nr_k}$ value, but not much significant as compared to the impact of LoP and other parameters on replica creation.

4.2.3. Performance Analysis for Data Replica Placement

In our proposed Algorithm for replica placement scheme, f_{pri}, is a crucial parameter influencing the experiments. In this subsection, we discuss the detailed analysis of this parameter.

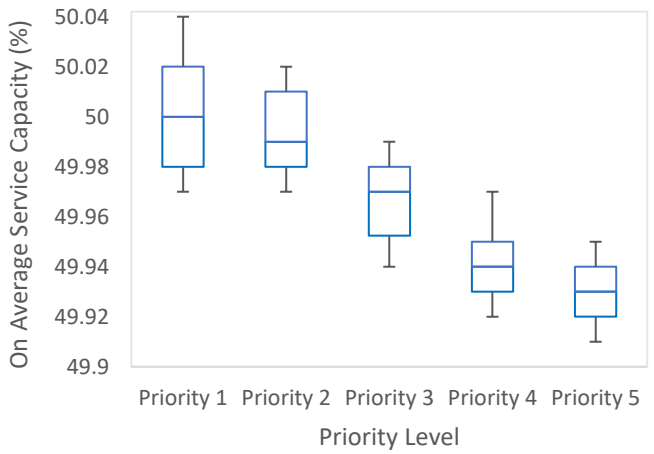


Figure 34 On Average Service Capacity for Different Priority level

1) Fog Node Priority (f_{pri}):

Total number of blocks B_i and service capacity m_{Sc} exerts the influence on fog nodes priority f_{pri} for replica placement. We define priority as 1, ---, FN_k (where FN_k is a total number of fog nodes). The highest priority is 1, which is given to a node to place a replica of a block, whereas FN_k represents the lowest priority given to a fog node for replica placement. The highest priority shows that the fog node contains the highest service capacity m_{Sc} and a smaller number of blocks B_i of the kth file as compared with the D.F Equations as mentioned in Section III. For example, from our basic scenarios, six fog nodes are participating in replica placement. Priority f_{pri} of replica placement on the fog node varies from 1 to 6, one is the highest priority for replica placement consideration, whereas six is the lowest priority.

The service capacity m_{Si} of each fog node remains within close range of 49.92 % to 50.04% for f_{pri} priority level, as shown in Figure 34. It is due to the reason that the fog nodes considered

for data blocks and replica storage do not process the replica creation or distribution on fog nodes, which could affect the overall service capacity of a fog node. Hence the m_{Si} of all fog nodes remain almost the same with no noteworthy difference on priority levels.

The number of blocks of the k[th] file present at the fog nodes has a significant impact on priority levels f_{pri}. For replica placement of a block B_k, the highest priority is assigned to the fog node with a few blocks of the k[th] file kept at the fog node. The priority level ensures the privacy of the k[th] file by minimizing the possibility of a high number of B_k and replicas placed at the same node. Also, the priority level balances the storage usage among fog nodes.

4.2.4. Time Series Analysis of the Proposed Replica Creation and Placement Schemes with Autoregressive Integrated Moving Average (ARIMA)

We consider the Autoregressive Integrated Moving Average (ARIMA) model to help verify the accuracy of our proposed replica creation and replica placement strategies [65]. ARIMA is a statistical analysis model that uses time-series data to understand the trend of data better or predict the future trends of the data for any problem solution [311]. In our experiments, we perform ten predictions for each miner and fog node and obtain the AIC (Akaike Information Criterion) values to measure the fitness of data to the ARIMA model.

We present, one of the results of the diagnostic in Figure 35, which indicates that there is no unusual pattern of data noticed in residual distribution. The diagnostic also suggests that the data residuals of proposed strategies into Arima models are normally distributed. In the top right histogram graph, the KDE red line follows the yellow line closely with a standard deviation of 1 and an average mean of 0, which is represented as standard notation N (0,1). The KDE red line indicates that the residuals are near normally distributed with a yellow N (0,1) line. Also, the

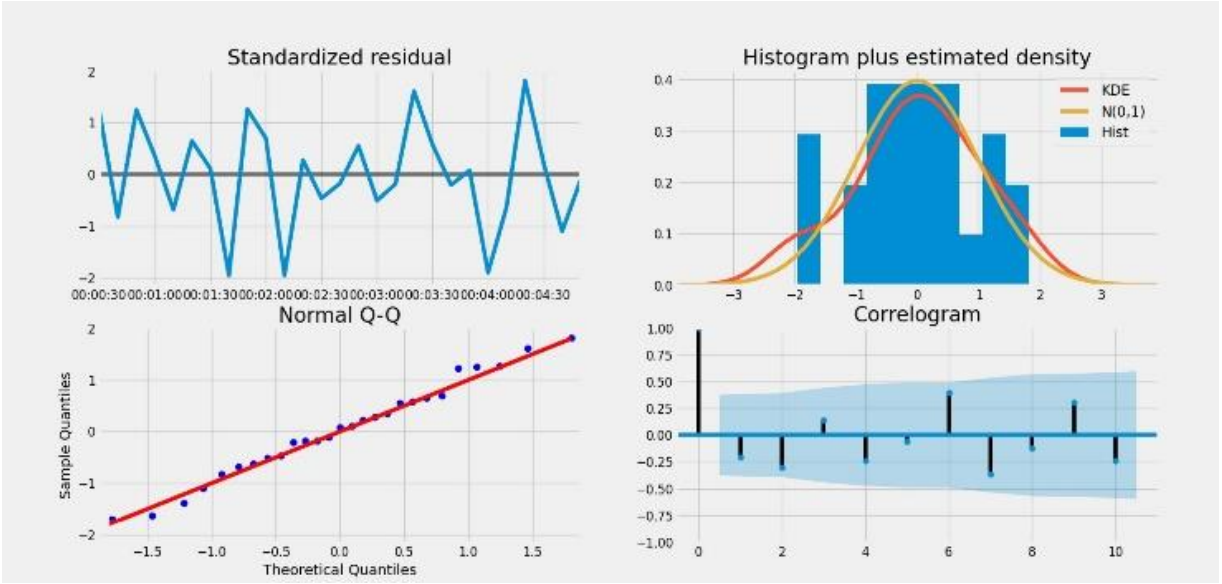


Figure 35 Diagnostic Result of the Replica creation model

bottom left q-q plot strongly suggests that the residuals' normal distribution (blue dots) is close to N (0, 1) (red line). The data residuals in the top left plot do not display any evident seasonality over a time interval of 5 minutes (300 seconds). This plot is summarised by the correlation in the bottom right plot, which displays the correlogram with 1 to 10 lags.

The plot shows that the ACF (Auto-correlation function) for the data residuals is flat, indicating that all the information is captured within the shaded blue area and showing no significance at different lags. Likewise, all the other miner and fog nodes' model diagnostic observations also lead to the conclusion that the model indicates a normal distribution of residuals and the model is a satisfactory fit for validating and forecasting time series data.

We compare models of predicted replica creation and replica placement strategies using dependant and independent variables. The duration of the models' forecast is set from 20 seconds to the end of 300 seconds (5 minutes). Overall, the forecast of miner and fog nodes in both models aligns well with the actual values. We present a few of the forecasting results of miner and fog nodes in Figure 36 and Figure 37. Figure 36 shows an average forecast (actual data vs predicted data) of the total number of replicas at miner nodes. It is clear from Figure 36 that the expected number of replicas being processed by miner nodes on average is close to the actual number of replicas processed by all the miner nodes. Figure 37 shows the average forecasts for the priority level of fog nodes in a model of replica placement scheme. The predicted priority level of the fog node on average is aligned very well to the actual priority value.

Further, it is useful to analyze the forecasts' accuracy in terms of MSE (Mean Squared Error) and RMSE (Root Mean Squared Error), which quantifies the average error of our forecasts. Table 16 presents the summary statistics of both models' MSE and RMSE's minimum, maximum,

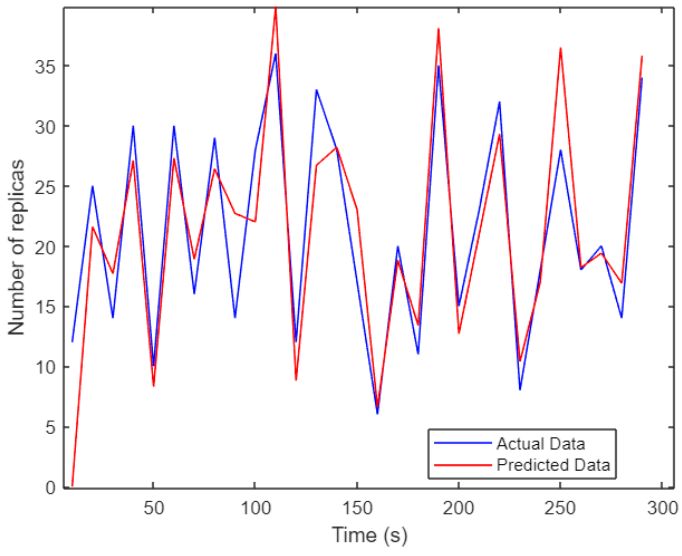


Figure 36 On Average Actual Vs Predicted Forecast at Fog Layer 1

average, and standard deviation values. At fog layer 1 (miner nodes), the MSE forecast of the proposed replica creation model yields a value of approximately 65, whereas the RMSE's value is almost 8. For fog layer 2 (fog nodes), the MSE and RMSE values of the replica placement model vary from 2 to 7 and 1 to 3, respectively. These MSE and RMSE values yield that the distance of predicted values from the actual values is close enough to validate the time series models.

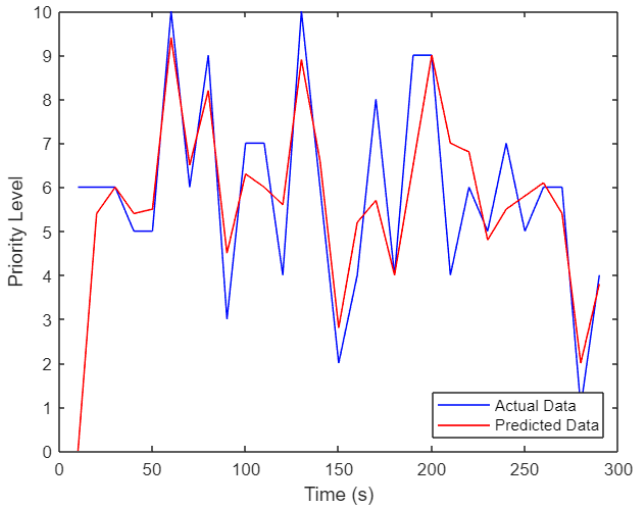


Figure 37 On Average Actual Vs Predicted Forecast at Fog Layer 2

Table 16 Summary Statistics

Node	Min	Max	Average	SD
fog Layer 1				
MSE	65.06	65.32	65.25	0.2803
RMSE	8.07	8.08	8.07	0.008
fog Layer 2				
MSE	2.85	6.75	5.25	1.2622
RMSE	1.69	2.6	2.27	0.2979

4.3. Performance Analysis of the Proposed Replica Creation Scheme vs DRC-AH and DRCA schemes

In this section, we compare the proposed replica creation scheme with the DRC-AH [62] and the DRCA schemes [312]. The DRC-AH replica creation scheme depends upon file access popularity. The Algorithm first analyses the access frequency of the data blocks then predicts the future access frequency according to the previously accessed rate. After that, it creates and adjusts the number of replicas based on data block access [62]. In the DRC-AH scheme, the Grey Markov chain model is used for data prediction and correction. Dynamic replicas are created based on the predicted values from the Grey Markov chain. For the DRCA Algorithm, data dynamic replica creation depends upon the file access popularity and node workload *[312]*. Further, three historical periods of file popularity and access frequency are utilized to predict the required replicas for replica creation. We compare our proposed replica creation scheme in terms of

computational cost with DRC-AH and DRCA strategies, as shown in Figure 38. For the comparison analysis, we consider 300 seconds time interval, a data file of 1 M.B and 5 sensor nodes, 4 miner nodes, and 6 fog nodes as provided for the basic scenario layout.

It is clear from Figure 38 that the average computational cost of our replica creation Algorithm is less than that of DRC-AH and DRCA. The reason behind this is that our replica creation scheme creates several replicas for each block based on the formula provided in Equation 5.4, which considers the LoP defined by data-owner, influencing factors ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$), total and average load. A miner node computes these parameters for each block replica creation. The computation required by each miner node is balanced as the service capacity of miner nodes is monitored by the miner's head to balance a load of each miner node during replica creation. Therefore, the computational cost of each miner node on average varies between 0.05 to 0.9 (s).

Whereas, for replica creation in the DRC-AH scheme, first of all, the frequency of data access is forecasted using a Markov chain. Then several replicas are created based on the forecast values. The prediction and accuracy of values for data access at each miner node requires a high computational cost. The computational cost varies from 0.4 to 1.6 (s). At 180 s, on average computational cost is around 1.9 (s) due to delay in data access prediction. Also, in the DRCA Algorithm, the file popularity and access frequency of blocks are predicted based on historical values that slow down the processing of replica creation. The condition of node load and access to data block frequency affect the overall replica creation performance, which varies from 0.5 to 1.89 (s). On the contrary, our scheme requires an average of 0.8 (s) to compute replica creation.

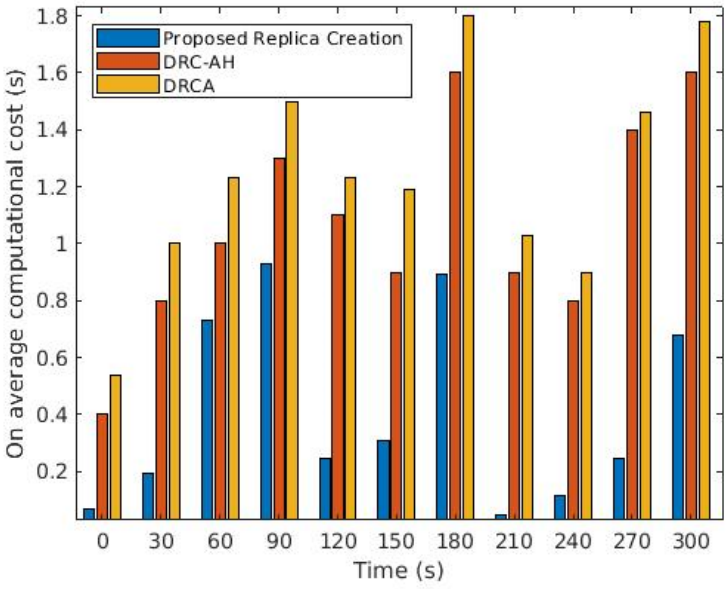


Figure 38 Computational Cost Comparison

The utilization of replica creation in terms of memory cost comparison is shown in Figure 39. We can see that the average memory cost of our replica creation Algorithm is significantly less than that of DRCA and DRC-AH. This is because of the balance of replica memory (in bytes) utilization at each miner node during the process of replica creation. On average, 42 bytes are used for each replica creation at each miner node. Whereas, the DRC-AH scheme for storage usage in replica creation requires on average 60 bytes using the Markov chain rule. DRCA requires higher storage usage than the DRC-AH and our proposed scheme. DRCA requires an average storage usage of 65 bytes to compute three periods of optimal replicas for dynamic replica creation.

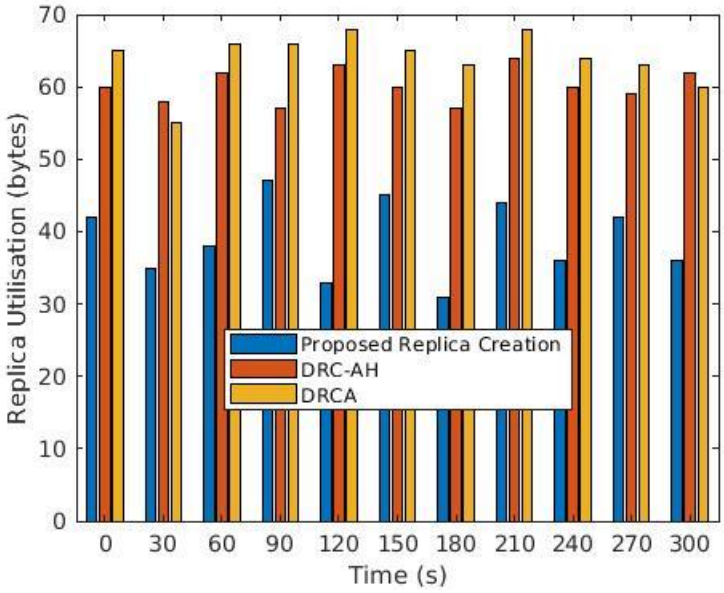


Figure 39 Memory Cost Comparison

Overall, the communication costs of the DRC-AH and DRCA are less than our replica creation Algorithm, as shown in Figure 40. This is because the processing in our replica creation scheme includes multiple nodes, continuous communication between fog layer 1's miner nodes, and sensor nodes. Our miner head node communicates with miner nodes to monitor service capacity and transmission of sensor data for replica creation. On the contrary, DRC-AH involves an application manager for predicting file access frequency. The node manager in the DRC-AH scheme is used to monitor the application manager's activities based on the prediction that is used for replica creation. Then the resource manager receives replica data, which is a process using the Grey Markov chain to accurately predict the data popularity and create a dynamic number of replicas. DRCA utilizes slightly higher communication overhead than DRC-AH, because of hybrid cloud and edge communication during the computation for file popularity and

access frequency. Still, the communication cost of DRCA is slightly less than our replica creation Algorithm.

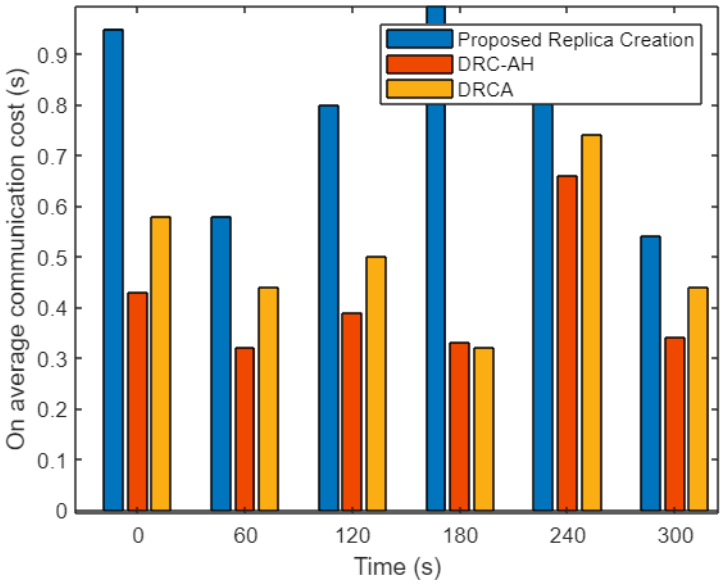


Figure 40 Communication Cost Comparison

5. Conclusion and Future Work

Aiming at the problem of preserving the privacy of data replicas which is essential for replica's data protection, reliability, and authentication, we propose privacy-preserving data replica creation and placement schemes in fog computing. In the data replica creation scheme, firstly, the service capacity of nodes is determined, and the node with the highest service capacity is selected to create a data replica. Then data replicas are generated based on the Level of Privacy (LoP) defined by data-owners. In the data placement scheme, the priority level based on the LoP and service capacity of each node is determined. We consider a node with the highest priority level and service capacity for replica placement. We discuss the experimental and comparative analysis of both schemes. Time-series analysis of each scheme verifies the accuracy of the proposed models. The comparative analysis of the proposed schemes shows that performance efficiency in terms of computational and memory costs is better than that of the state-of-the-art schemes.

In the future, we will improve the performance of our proposed schemes in terms of communication overhead. Also, consider a replica selection scheme for data requests by end-users/cloud as the main aim of our current schemes is to preserve the privacy of replication within the fog layer, the privacy of replication at end-user/cloud is out of the scope. Therefore, we aim to improve our schemes to preserve end-to-end replication privacy.

'One worthwhile task carried to a successful conclusion is better than 50 half-finished tasks.'

---B.C. Forbes

Chapter 6 Conclusion and Future Work

In this Chapter, the conclusion of the thesis along with limitations and future directions of the proposed framework are discussed. First, Section 6.1 presents the summary of the thesis contributions in detail. Then the limitations and recommended future research of this thesis work are presented in Section 6.2.

6.1. Summary

Considering privacy concerns over fog networks for data aggregation and replication of IoT applications, this thesis presented a privacy-preserving framework in a fog-enabled IoT network. For this framework, lightweight data aggregation and replication schemes are proposed. Experimental evaluation and simulation studies are performed, which proves the effectiveness and efficiency of the proposed schemes. The schemes for preserving data privacy in the fog computing paradigm are investigated using analytical modelling in this thesis.

The proposed data aggregation scheme utilizes a lightweight Divide-and-Conquer method that divides and distributes encrypted data to strengthen data privacy. In addition, the proposed data aggregation scheme is extended to optimize the time and energy consumption of the Divide-and-Conquer method. The optimization method used in the proposed scheme has significantly reduced the time and energy consumption of the data aggregation. This thesis is advanced further to preserve the privacy of data replication using the system model of the data aggregation scheme. Data replica creation scheme and data replica placement scheme are proposed for efficient and effective data replication in the fog computing paradigm.

In this thesis, the results of all experiments and simulations have been thoroughly discussed and presented. Different network scenarios are considered to analyze the correlation between influencing factors used in the framework. The privacy vulnerabilities of the proposed schemes are also analyzed. The essential investigations performed for statistical and privacy evaluations give solid proof for the efficiency and effectiveness of the proposed framework. This thesis has discussed these contributions in detail in Chapters 2 to 5. A summary of the contributions is presented below.

An initial part of this thesis investigated the privacy requirements and analyzed the privacy-preserving schemes and challenges in Chapter 2. The analysis outlined the research gaps, which have motivated developing a framework for efficient and effective data privacy in the fog computing paradigm. Then a Divide-and-Conquer method is proposed for a data aggregation scheme based on the data division strategy in Chapter 3. The level of Privacy (LoP) defined by the data-owner is considered for the data division strategy. In this strategy, lightweight cryptographic operations are first used to secure the data, and then encrypted data is divided and distributed among fog nodes for data storage and aggregation processing. For encrypted data division and distribution, we applied linear and tabular divisions according to the LoPs. The data division strategy divides data according to LoP and distributes it among participating fog nodes for data aggregation processing and reduces computational and memory

overhead in the processing simultaneously compared to the other schemes. Also, an additive aggregation for the data aggregation is considered for computing the sum aggregate of data.

The computational and memory overhead of the proposed scheme in terms of data encryption, division, and aggregation is reasonably low compared to state-of-the-art schemes. The proposed scheme is based on a symmetric cipher AES algorithm, which requires less computational time than other asymmetric ciphers based schemes. Further, data distribution among fog nodes for data aggregation reduces the computational overhead on a single fog node, speeds up the aggregation process, and minimizes a single point of failure risk.

In this Chapter, the formal privacy analysis is also performed that considers active and passive attack scenarios that prove data privacy's effectiveness in a Fog-enabled IoT network. Propositions based on learning, challenge, and guess phases in the privacy analysis model proved that the proposed data aggregation scheme satisfies the Indistinguishability-based untraceable Privacy (INDPriv) of data.

In Chapter 4, the data aggregation scheme is extended to optimize the time and energy consumption during data aggregation in the fog computing paradigm. In this Chapter, mathematically modelling of time and energy consumption in terms of execution, transmission, and precursor waiting time is carried out. NSGA III method is used to solve the optimization problem of the formulated time and energy consumption. In the NSGA III method, the reference-point-based SAW and MCDM are employed to select an optimal solution to minimize the joint objective (i.e. time and energy consumption).

In this chapter, the test-case scenarios to test the performance efficiency of the proposed scheme are also provided. The impact of evaluation factors including the number of fog nodes, the execution and transmission power, the computing capacity, the data size, the degree of workload imbalance, and the standard deviation of the workload imbalance on the time and energy consumptions are used to evaluate the performance efficiency of the proposed scheme. Further, the performance efficiency of the proposed scheme is evaluated in terms of data size and power consumption compared to state-of-the-art schemes. The experiment results showed that the proposed scheme can always obtain the Pareto optimal solutions within the extreme values. It outperforms the state-of-the-art schemes in solving the optimization problem.

Chapter 5 further advanced the data aggregation scheme for data replication in the Fog computing paradigm. Data replica creation and placement schemes are considered for secure data replication. These schemes guaranteed data replica’s protection, reliability, and authentication. The mathematical modelling of these schemes is based on the level of privacy

defined by the data-owner, service capacity, and level of priority of each fog node in a network. In this Chapter, the time complexity analysis of the proposed schemes is also discussed, which shows that the time utilized for computation tasks is linear as the data replica creation, and data replica placement schemes require O ($n \times m$) and O (($n \times m$) + k) time, respectively.

Statistical analysis of the proposed schemes is also performed to evaluate the correlation between sensitive parameters, comparison, and time series analysis. Parameters considered for exerting impact on the data replica creation and replica placement schemes are service capacity, Level of Privacy (LoP), fog node priority, and the ratio of nodes communicating in the network. The correlation analysis showed that the service capacity and LoP exert a high impact on the data replica creation scheme as these parameters determine the load capacity and the total number of replicas to be created at a fog node. For the replica placement scheme, the number of blocks for the fog nodes priority has a significant impact compared to the service capacity. The fog nodes considered for data replica placement do not compute the replica creation or distribution, which could impact the overall service capacity of a fog node. In the comparative analysis, the performance efficiency of the proposed schemes in terms of computational and memory costs is better than that of the state-of-the-art schemes.

Further, the accuracy of the proposed schemes is verified using Autoregressive Integrated Moving Average (ARIMA) model. In the experiments, ten data predictions for each node in the fog network are performed. The AIC (Akaike Information Criterion) values are obtained to measure the fitness of data to the ARIMA model. The results indicated that the data residuals are normally distributed with no unusual patterns, and the model is a satisfactory fit for the validation and forecasting of time series data. Furthermore, the forecast data aligned well with the actual data into both replica creation and placement models. Forecast accuracy is also analyzed using Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). The resultant values yield that the distance of forecast values from the actual values is close enough to validate the time series models.

The formal analysis of privacy is also provided in this Chapter. The privacy analysis proved that the proposed schemes are CPA-secure. Since the computational complexity for an attacker to extract private information about a data replica block is high in probabilistic polynomial time, the replica privacy remains intact. Also, the privacy analysis emphasis on the data replica reliability in which the original data block can be recovered from replica blocks placed at fog nodes.

6.2. Thesis Limitations and Recommendations for Future Research

This Section summarizes the limitations and future directions of the proposed framework. The limitations and future work discussed in this section do not in any way undermine the validity of the research contributions undertaken in this thesis. However, they suggest research work that can be carried out in the future to enhance the proposed framework and accommodate diverse fog-enabled IoT applications.

The proposed framework is implemented and evaluated using a network simulator. In a simulation, the consideration of fog node's fault tolerance due to failures including defective calibration, environmental interference, instability of transmission link, software problems, and physical damage cannot be measured accurately compared to real-time evaluation. In a real-time implementation, the devices are deployed and interconnected in an open environment where these failures can be measured accurately. In addition, the data size considered for the evaluation of the proposed framework is between 1 KB to 100 MB. In a real-time simulation, the sensor devices generate more than 100 MB of data, which must be aggregated and replicated securely in a wireless network. Therefore, these limitations entail the real-time simulation of the proposed framework to consider the larger data-set and fault tolerance failures.

Further, an additive aggregation operation for the summation of data is used in the proposed data aggregation scheme in Chapter 3. The operation sums all the data generated from IoT devices and provides a resultant summation to the end-user. A possible research direction is to apply both additives and non-additive aggregation operations, including percentile, maximum, minimum, ratio, average, and mean to the proposed data aggregation scheme. The consideration of these data aggregation operations increases the diversity of analyzing data trends for IoT applications.

Another limitation of this thesis is that the communication overhead of the proposed data aggregation and replication scheme is significantly higher than the state-of-the-art schemes, as discussed in chapters 2 and 4. In both proposed schemes, processing of computation tasks, including data aggregation, replica creation, and storage, is performed on multiple fog nodes. For performing these computation tasks, continuous interaction of fog nodes for accessing, processing, storing data, and authenticating each other increases the overall communication overhead of the fog network. The optimization of communication overhead for the data aggregation scheme is considered in Chapter 4. The optimization method minimizes the fog network's communication cost (transmission time) using the NSGA III method. The optimization method to reduce the communication cost of the data replication scheme has not been considered in this thesis. Therefore, In the future, optimization algorithms for reducing

the communication overhead of data replication schemes will enhance the performance efficiency of the proposed framework.

This thesis demonstrated the effectiveness and efficiency of preserving data privacy within the fog network. The data privacy at the sensors and end-users or cloud is out of this thesis's scope. Therefore, a possible research direction is to provide end-to-end data privacy for efficient data aggregation and replication. This thesis only considered data replica creation and placement schemes for creating and placing data replicas securely in the Fog network. Data replicas selection for replicas placed at the fog layer requested by end-users is considered as one of the promising future works. For a secure end-to-end framework, the privacy of data replica for replica selection and transmission to end-user can be added to the proposed schemes. In a data replica selection scheme, decision-making methods can be utilized to search and select the best suitable replica for the end-user.

References

1. Airehrour, D., J. Gutierrez, and S.K. Ray, *Secure routing for internet of things: A survey.* Journal of Network and Computer Applications, 2016. **66**: p. 198-213.
2. Alaba, F.A., et al., *Internet of Things security: A survey.* Journal of Network and Computer Applications, 2017. **88**: p. 10-28.
3. Lin, J., et al., *A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications.* IEEE Internet of Things Journal, 2017. **4**(5): p. 1125-1142.
4. Yang, Y., et al., *A survey on security and privacy issues in Internet-of-Things.* IEEE Internet of Things Journal, 2017. **4**(5): p. 1250-1258.
5. Puthli, R. *The Internet of Things will be much, much bigger than mobile*. 2015; Available from: https://blog.itude.com/2015/03/06/the-internet-of-things-will-be-much-much-bigger-than-mobile/.
6. *Connected IoT devices Forecast*. Help Net Security 2019 [cited 2019; 41.6 billion IoT devices will be generating 79.4 zettabytes of data in 2025]. Available from: https://www.helpnetsecurity.com/2019/06/21/connected-iot-devices-forecast/.
7. Guan, Z., et al., *APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT.* Journal of Network and Computer Applications, 2019. **125**: p. 82-92.
8. Rayes, A. and S. Salam, *Internet of things from hype to reality.* Springer, 2017.
9. Wang, H., Z. Wang, and J. Domingo-Ferrer, *Anonymous and secure aggregation scheme in fog-based public cloud computing.* Future Generation Computer Systems, 2018. **78**: p. 712-719.
10. Bonomi, F., et al. *Fog computing and its role in the internet of things*. in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012. ACM.
11. Bonomi, F. *Connected vehicles, the internet of things, and fog computing*. in *The eighth ACM international workshop on vehicular inter-networking (VANET), Las Vegas, USA*. 2011.
12. Sharma, P.K., M.-Y. Chen, and J.H. Park, *A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT.* IEEE Access, 2018. **6**: p. 115-124.
13. Yang, M., et al., *Machine Learning Differential Privacy with Multifunctional Aggregation in a Fog Computing Architecture.* IEEE Access, 2018.
14. Naha, R.K., et al., *Fog computing: Survey of trends, architectures, requirements, and research directions.* IEEE access, 2018. **6**: p. 47980-48009.
15. Luan, T.H., et al., *Fog computing: Focusing on mobile users at the edge.* arXiv preprint arXiv:1502.01815, 2015.
16. Lu, R., et al., *A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT.* 2017. **5**: p. 3302-3312.
17. Jayaraman, P.P., et al., *Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation.* Future Generation Computer Systems, 2017. **76**: p. 540-549.
18. Ara, A., et al., *A Secure Privacy-Preserving Data Aggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems.* IEEE Access, 2017. **5**: p. 12601-12617.
19. Solove, D.J. and D.K. Citron, *Risk and anxiety: A theory of data-breach harms.* Tex. L. Rev., 2017. **96**: p. 737.
20. *The rapid growth of IoT devices, which are equipped with processing power and sensors are predicted to approximately reach 43 billion by year 2020.* . 2016; Available from: http://www.businessinsider.com/iot-ecosystem-internet-of-things-forecasts-and-business-opportunities-2016-2/?r=AU&IR=T.
21. Sarwar, K., et al., *Lightweight, Divide-and-Conquer privacy-preserving data aggregation in fog computing.* Future Generation Computer Systems, 2021. **119**: p. 188-199.

22. Vahedi, E., et al., *A secure ECC-based privacy preserving data aggregation scheme for smart grids.* Computer Networks, 2017. **129**: p. 28-36.
23. Adat, V. and B.J.T.S. Gupta, *Security in Internet of Things: issues, challenges, taxonomy, and architecture.* 2018. **67**(3): p. 423-441.
24. Hossain, M.M., M. Fotouhi, and R. Hasan. *Towards an analysis of security issues, challenges, and open problems in the internet of things*. in *2015 IEEE World Congress on Services*. 2015. IEEE.
25. Nepal, S., R. Ranjan, and K.-K.R.J.I.C.C. Choo, *Trustworthy processing of healthcare big data in hybrid clouds.* 2015. **2**(2): p. 78-84.
26. Haghighat, M., S. Zonouz, and M.J.E.S.w.A. Abdel-Mottaleb, *CloudID: Trustworthy cloud-based and cross-enterprise biometric identification.* 2015. **42**(21): p. 7905-7916.
27. Pandit, K., et al., *Adaptive traffic signal control with vehicular ad hoc networks.* IEEE Transactions on Vehicular Technology, 2013. **62**(4): p. 1459-1471.
28. Zhang, L., et al., *Distributed aggregate privacy-preserving authentication in VANETs.* IEEE Transactions on Intelligent Transportation Systems, 2017. **18**(3): p. 516-526.
29. Xia, Z., et al., *A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing.* IEEE Transactions on Information Forensics and Security, 2016. **11**(11): p. 2594-2608.
30. Yuan, J. and S. Yu. *Efficient privacy-preserving biometric identification in cloud computing*. in *INFOCOM, 2013 Proceedings IEEE*. 2013. IEEE.
31. Yang, C., et al., *Towards product customization and personalization in IoT-enabled cloud manufacturing.* Cluster Computing, 2017. **20**(2): p. 1717-1730.
32. Sand, G., et al. *A big data aggregation, analysis and exploitation integrated platform for increasing social management intelligence*. in *2014 IEEE International Conference on Big Data (Big Data)*. 2014. IEEE.
33. Ghugal, S. and M. Vaidya, *Military Network Security for Data Retrieval with Touch of Smell Technology.* 2016.
34. Li, S., et al., *PPMA: Privacy-preserving multisubset data aggregation in smart grid.* IEEE Transactions on Industrial Informatics, 2018. **14**(2): p. 462-471.
35. Xu, C., et al., *PAVS: a new privacy-preserving data aggregation scheme for vehicle sensing systems.* Sensors, 2017. **17**(3): p. 500.
36. Han, S., et al., *PPM-HDA: Privacy-preserving and multifunctional health data aggregation with fault tolerance.* IEEE Transactions on Information Forensics and Security, 2016. **11**(9): p. 1940-1955.
37. Chen, C.-M., et al., *RCDA: Recoverable concealed data aggregation for data integrity in wireless sensor networks.* IEEE Transactions on parallel and distributed systems, 2012. **23**(4): p. 727-734.
38. Castelluccia, C., et al., *Efficient and provably secure aggregation of encrypted data in wireless sensor networks.* ACM Transactions on Sensor Networks (TOSN), 2009. **5**(3): p. 20.
39. Othman, S.B., et al., *Confidentiality and integrity for data aggregation in WSN using homomorphic encryption.* Wireless Personal Communications, 2015. **80**(2): p. 867-889.
40. Tonyali, S., et al., *Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled smart metering systems.* 2018. **78**: p. 547-557.
41. Sharma, P.K., M.-Y. Chen, and J.H.J.I.A. Park, *A software defined fog node based distributed blockchain cloud architecture for IoT.* 2018. **6**: p. 115-124.
42. Sarwar, K., et al., *Lightweight, Divide-and-Conquer privacy-preserving data aggregation in fog computing.* Future Generation Computer Systems, 2021.
43. Bhalaji, N., *Efficient and secure data utilization in mobile edge computing by data replication.* Journal of ISMAC, 2020. **2**(01): p. 1-12.
44. Shakarami, A., et al., *Data replication schemes in cloud computing: a survey.* Cluster Computing, 2021: p. 1-35.

45. Milani, B.A. and N.J. Navimipour, *A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions.* Journal of Network and Computer Applications, 2016. **64**: p. 229-238.
46. Salem, R., et al., *An artificial bee colony algorithm for data replication optimization in cloud environments.* IEEE Access, 2019. **8**: p. 51841-51852.
47. Mansouri, N. and M.M. Javidi, *A new prefetching-aware data replication to decrease access latency in cloud environment.* Journal of Systems and Software, 2018. **144**: p. 197-215.
48. Mansouri, N. and M.M. Javidi, *A review of data replication based on meta-heuristics approach in cloud computing and data grid.* Soft Computing, 2020: p. 1-28.
49. Sivasankari, M.A., M.D. Abirami, and M.K. Ayesha, *Division and Replication of Data in Cloud for Optimal Performance and Security using Fragment Placement Algorithm.*
50. Huang, T., et al., *A latency-aware multiple data replicas placement strategy for fog computing.* Journal of Signal Processing Systems, 2019. **91**(10): p. 1191-1204.
51. Aral, A. and T. Ovatman, *A decentralized replica placement algorithm for edge computing.* IEEE transactions on network and service management, 2018. **15**(2): p. 516-529.
52. Shao, Y., C. Li, and H. Tang, *A data replica placement strategy for IoT workflows in collaborative edge and cloud environments.* Computer Networks, 2019. **148**: p. 46-59.
53. Naas, M.I., et al. *iFogStor: an IoT data placement strategy for fog infrastructure*. in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*. 2017. IEEE.
54. Truong, H.-L. and S. Dustdar, *Principles for engineering IoT cloud systems.* IEEE Cloud Computing, 2015. **2**(2): p. 68-76.
55. Wang, L. and R. Ranjan, *Processing distributed internet of things data in clouds.* IEEE Cloud Computing, 2015. **2**(1): p. 76-80.
56. Camillo, G.L., et al., *Preserving privacy with fine-grained authorization in an identity management system.* 2017: p. 86.
57. Ouafi, K. and R.C.-W. Phan. *Privacy of recent RFID authentication protocols*. in *International Conference on Information Security Practice and Experience*. 2008. Springer.
58. Gope, P. and B. Sikdar, *Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids.* IEEE Transactions on Information Forensics and Security, 2018. **14**(6): p. 1554-1566.
59. Xu, X., et al., *A computation offloading method over big data for IoT-enabled cloud-edge computing.* Future Generation Computer Systems, 2019. **95**: p. 522-533.
60. Afshari, A., M. Mojahed, and R.M. Yusuff, *Simple additive weighting approach to personnel selection problem.* International Journal of Innovation, Management and Technology, 2010. **1**(5): p. 511.
61. Aruldoss, M., T.M. Lakshmi, and V.P. Venkatesan, *A survey on multi criteria decision making methods and its applications.* American Journal of Information Systems, 2013. **1**(1): p. 31-43.
62. Li, C., J. Tang, and Y. Luo, *Scalable replica selection based on node service capability for improving data access performance in edge computing environment.* The Journal of Supercomputing, 2019. **75**(11): p. 7209-7243.
63. Chen, D., et al., *BOSSA: A Decentralized System for Proofs of Data Retrievability and Replication.* IEEE Transactions on Parallel and Distributed Systems, 2020. **32**(4): p. 786-798.
64. Shacham, H. and B. Waters, *Compact proofs of retrievability.* Journal of cryptology, 2013. **26**(3): p. 442-483.
65. Pal, A. and P. Prakash, *Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. 2017: Packt Publishing Ltd.

66. Bellendorf, J. and Z.Á. Mann, *Classification of optimization problems in fog computing.* Future Generation Computer Systems, 2020. **107**: p. 158-176.
67. Huang, Q., Y. Yang, and L. Wang, *Secure Data Access Control With Ciphertext Update and Computation Outsourcing in Fog Computing for Internet of Things.* IEEE Access, 2017. **5**: p. 12941-12950.
68. Stergiou, C., et al., *Secure integration of IoT and cloud computing.* 2018. **78**: p. 964-975.
69. Fernández-Alemán, J.L., et al., *Security and privacy in electronic health records: A systematic literature review.* Journal of biomedical informatics, 2013. **46**(3): p. 541-562.
70. Zhou, J., et al., *Security and Privacy for Cloud-Based IoT: Challenges.* IEEE Communications Magazine, 2017. **55**(1): p. 26-33.
71. Lopez, J., et al., *Evolving privacy: From sensors to the Internet of Things.* Future Generation Computer Systems, 2017. **75**: p. 46-57.
72. Sarwar, K., S. Yongchareon, and J. Yu. *A Brief Survey on IoT Privacy: Taxonomy, Issues and Future Trends*. in *International Conference on Service-Oriented Computing*. 2018. Springer.
73. Celik, Z.B., et al., *Program analysis of commodity IoT applications for security and privacy: Challenges and opportunities.* ACM Computing Surveys (CSUR), 2019. **52**(4): p. 74.
74. Aleisa, N. and K. Renaud, *Privacy of the Internet of Things: A Systematic Literature Review.* 2017.
75. Tewari, A. and B.J.F.G.C.S. Gupta, *Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework.* 2018.
76. Mukherjee, M., et al., *Security and privacy in fog computing: Challenges.* IEEE Access, 2017. **5**: p. 19293-19304.
77. Liu, X., et al., *Security and Privacy Challenges for Internet-of-Things and Fog Computing.* Wireless Communications and Mobile Computing, 2018. **2018**.
78. Yu, D., et al., *A survey on security issues in services communication of Microservices-enabled fog applications.* Concurrency and Computation: Practice and Experience, 2018: p. e4436.
79. Zhang, P., M. Zhou, and G. Fortino, *Security and trust issues in Fog computing: A survey.* Future Generation Computer Systems, 2018. **88**: p. 16-27.
80. Stojmenovic, I. and S. Wen. *The fog computing paradigm: Scenarios and security issues*. in *2014 Federated Conference on Computer Science and Information Systems*. 2014. IEEE.
81. Bai, L., et al., *A Survey on Cryptographic Security and Information Hiding Technology for Cloud or Fog-Based IoT System.* 2019.
82. Broenink, G., et al., *The privacy coach: Supporting customer privacy in the internet of things.* arXiv preprint arXiv:1001.4459, 2010.
83. Hu, C., J. Zhang, and Q. Wen. *An identity-based personal location system with protected privacy in IoT*. in *2011 4th IEEE International Conference on Broadband Network and Multimedia Technology*. 2011. IEEE.
84. Jhumka, A., M. Leeke, and S. Shrestha, *On the use of fake sources for source location privacy: Trade-offs between energy and privacy.* The Computer Journal, 2011. **54**(6): p. 860-874.
85. Raij, A., et al. *Privacy risks emerging from the adoption of innocuous wearable sensors in the mobile environment*. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011. ACM.
86. Wei, W., F. Xu, and Q. Li. *Mobishare: Flexible privacy-preserving location sharing in mobile online social networks*. in *2012 Proceedings IEEE INFOCOM*. 2012. IEEE.

87. Elaluf-Calderwood, S. and J. Liebenau, *Privacy, identity and security concerns: enterprise strategic decision making and business model development for mobile payments in NFC.* 2012.
88. Alcaide, A., et al., *Anonymous authentication for privacy-preserving IoT target-driven applications.* Computers & Security, 2013. **37**: p. 111-123.
89. Poslad, S., M. Hamdi, and H. Abie. *Adaptive security and privacy management for the internet of things (ASPI 2013)*. in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. 2013.
90. Virkki, J. and L. Chen, *Personal perspectives: Individual privacy in the IoT.* 2013.
91. Kai, K., Z.-b. Pang, and W. Cong, *Security and privacy mechanism for health internet of things.* The Journal of China Universities of Posts and Telecommunications, 2013. **20**: p. 64-68.
92. Yao, L., et al., *Protecting the sink location privacy in wireless sensor networks.* Personal and ubiquitous computing, 2013. **17**(5): p. 883-893.
93. Su, J., et al., *ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things.* 2014. **33**: p. 11-18.
94. Denning, T., Z. Dehlawi, and T. Kohno. *In situ with bystanders of augmented reality glasses: Perspectives on recording and privacy-mediating technologies*. in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014. ACM.
95. Bonetto, M., et al. *Privacy in mini-drone based video surveillance*. in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. 2015. IEEE.
96. Zöscher, L., et al. *Concept for a security aware automatic fare collection system using HF/UHF dual band RFID transponders*. in *Solid State Device Research Conference (ESSDERC), 2015 45th European*. 2015. IEEE.
97. Salonikias, S., I. Mavridis, and D. Gritzalis. *Access control issues in utilizing fog computing for transport infrastructure*. in *International Conference on Critical Information Infrastructures Security*. 2015. Springer.
98. Samani, A., H.H. Ghenniwa, and A. Wahaishi, *Privacy in Internet of Things: A model and protection framework.* Procedia Computer Science, 2015. **52**: p. 606-613.
99. Aditya, P., et al. *I-pic: A platform for privacy-compliant image capture*. in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 2016. ACM.
100. Yang, X., et al., *A novel temporal perturbation based privacy-preserving scheme for real-time monitoring systems.* Computer Networks, 2015. **88**: p. 72-88.
101. Neisse, R., et al., *SecKit: a model-based security toolkit for the internet of things.* 2015. **54**: p. 60-76.
102. Seo, S.-H., J. Won, and E.J.T.o.D.P. Bertino, *pCLSC-TKEM: a Pairing-free Certificateless Signcryption-tag Key Encapsulation Mechanism for a Privacy-Preserving IoT.* 2016. **9**(2): p. 101-130.
103. Ivaşcu, T., M. Frîncu, and V. Negru. *Considerations towards security and privacy in internet of things based ehealth applications*. in *2016 IEEE 14th International Symposium on Intelligent Systems and Informatics (SISY)*. 2016. IEEE.
104. Mao, Y., et al., *Fully secure fuzzy identity-based encryption for secure IoT communications.* 2016. **44**: p. 117-121.
105. Laikin, J.F., et al. *Towards fake sources for source location privacy in wireless sensor networks with multiple sources*. in *2016 IEEE International Conference on Communication Systems (ICCS)*. 2016. IEEE.
106. Bradbury, M. and A. Jhumka. *Understanding source location privacy protocols in sensor networks via perturbation of time series*. in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. 2017. IEEE.
107. Diyanat, A., A. Khonsari, and H. Shafiei, *Preservation of temporal privacy in body sensor networks.* Journal of Network and Computer Applications, 2017. **96**: p. 62-71.

108. Wang, N. and J. Zeng, *All-direction random routing for source-location privacy protecting against parasitic sensor networks.* Sensors, 2017. **17**(3): p. 614.
109. Hu, P., et al., *Security and Privacy Preservation Scheme of Face Identification and Resolution Framework Using Fog Computing in Internet of Things.* IEEE Internet of Things Journal, 2017. **4**(5): p. 1143-1155.
110. Apthorpe, N., et al., *Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic.* 2017.
111. Wu, Q., et al., *Privacy-aware multipath video caching for content-centric networks.* 2016. **34**(8): p. 2219-2230.
112. Kumar, P., et al., *Anonymous Secure Framework in Connected Smart Home Environments.* IEEE Trans. Information Forensics and Security, 2017. **12**(4): p. 968-979.
113. Amin, R., et al., *A robust and anonymous patient monitoring system using wireless medical sensor networks.* 2018. **80**: p. 483-495.
114. Kumar, P., et al., *Anonymous secure framework in connected smart home environments.* IEEE Transactions on Information Forensics and Security, 2017. **12**(4): p. 968-979.
115. Hong, Y., W.M. Liu, and L. Wang, *Privacy preserving smart meter streaming against information leakage of appliance status.* IEEE transactions on information forensics and security, 2017. **12**(9): p. 2227-2241.
116. Dorri, A., et al. *Blockchain for IoT security and privacy: The case study of a smart home*. in *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*. 2017. IEEE.
117. Gope, P., J. Lee, and T.Q. Quek, *Resilience of DoS attacks in designing anonymous user authentication protocol for wireless sensor networks.* IEEE Sensors Journal, 2017. **17**(2): p. 498-503.
118. Jayaraman, P.P., et al., *Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation.* 2017. **76**: p. 540-549.
119. Le, J., et al., *Full autonomy: A novel individualized anonymity model for privacy preserving.* 2017. **66**: p. 204-217.
120. Du, M., et al., *A differential privacy-based query model for sustainable fog data centers.* IEEE Transactions on Sustainable computing, 2017.
121. Huo, Y., et al., *LoDPD: A location difference-based proximity detection protocol for fog computing.* IEEE Internet of Things Journal, 2017. **4**(5): p. 1117-1124.
122. Bradbury, M., A. Jhumka, and M. Leeke, *Hybrid online protocols for source location privacy in wireless sensor networks.* Journal of Parallel and Distributed Computing, 2018. **115**: p. 67-81.
123. Gope, P., et al., *Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment.* 2018. **83**: p. 629-637.
124. Lu, Y. and N. Sun, *A resilient data aggregation method based on spatio-temporal correlation for wireless sensor networks.* EURASIP Journal on Wireless Communications and Networking, 2018. **2018**(1): p. 157.
125. De Capitani di Vimercati, S., et al., *Enforcing authorizations while protecting access confidentiality.* 2018(Preprint): p. 1-33.
126. Cheng, K., Y. Hou, and L. Wang. *Secure Similar Sequence Query on Outsourced Genomic Data*. in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 2018. ACM.
127. Yan, H., et al., *Centralized duplicate removal video storage system with privacy preservation in IoT.* 2018. **18**(6): p. 1814.
128. Li, X., et al., *A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments.* 2018. **103**: p. 194-204.
129. Liu, X., et al., *Hybrid privacy-preserving clinical decision support system in fog–cloud computing.* Future Generation Computer Systems, 2018. **78**: p. 825-837.

130. Thota, C., et al., *Centralized fog computing security platform for IoT and cloud in healthcare system*, in *Exploring the convergence of big data and the internet of things*. 2018, IGI Global. p. 141-154.
131. Karopoulos, G., C. Ntantogian, and C. Xenakis, *MASKER: Masking for privacy-preserving aggregation in the smart grid ecosystem.* Computers & Security, 2018. **73**: p. 307-325.
132. Wang, H., Z. Wang, and J.J.F.G.C.S. Domingo-Ferrer, *Anonymous and secure aggregation scheme in fog-based public cloud computing.* 2018. **78**: p. 712-719.
133. Koo, D. and J. Hur, *Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing.* Future Generation Computer Systems, 2018. **78**: p. 739-752.
134. Yekta, N.I. and R. Lu. *Xrquery: Achieving communication-efficient privacy-preserving query for fog-enhanced iot*. in *2018 IEEE International Conference on Communications (ICC)*. 2018. IEEE.
135. Belguith, S., et al., *Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot.* Computer Networks, 2018. **133**: p. 141-156.
136. Challa, S., et al., *An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks.* Computers & Electrical Engineering, 2018. **69**: p. 534-554.
137. Pasquier, T., et al., *Data provenance to audit compliance with privacy policy in the Internet of Things.* Personal and Ubiquitous Computing, 2018. **22**(2): p. 333-344.
138. Tao, M., et al., *Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes.* 2018. **78**: p. 1040-1051.
139. Kapusta, K., G. Memmi, and H.J.A.o.T. Noura, *Additively homomorphic encryption and fragmentation scheme for data aggregation inside unattended wireless sensor networks.* 2019: p. 1-9.
140. Saha, R., et al., *Privacy Ensured ${e} $-Healthcare for Fog-Enhanced IoT Based Applications.* IEEE Access, 2019. **7**: p. 44536-44543.
141. Yin, X.C., et al., *An IoT-Based Anonymous Function for Security and Privacy in Healthcare Sensor Networks.* Sensors, 2019. **19**(14): p. 3146.
142. Gu, J., et al., *A Fog Computing Solution for Context-Based Privacy Leakage Detection for Android Healthcare Devices.* Sensors, 2019. **19**(5): p. 1184.
143. Zhang, H., et al., *Video denoising for security and privacy in fog computing.* Concurrency and Computation: Practice and Experience, 2019. **31**(22): p. e4763.
144. Liu, J.-N., et al., *Enabling Efficient and Privacy-Preserving Aggregation Communication and Function Query for Fog Computing-Based Smart Grid.* IEEE Transactions on Smart Grid, 2019. **11**(1): p. 247-257.
145. Wang, L., Z. Hu, and L. Liu. *Privacy-Preserving and Dynamic Spatial Range Aggregation Query Processing in Wireless Sensor Networks*. in *International Conference on Database Systems for Advanced Applications*. 2019. Springer.
146. Wang, L., et al., *A novel privacy-and integrity-preserving approach for multidimensional data range queries in two-tiered wireless sensor networks.* International Journal of Distributed Sensor Networks, 2019. **15**(6): p. 1550147719855893.
147. Gai, K., et al., *Multi-access filtering for privacy-preserving fog computing.* IEEE Transactions on Cloud Computing, 2019.
148. Alemneh, E., et al., *A two-way trust management system for fog computing.* Future Generation Computer Systems, 2020. **106**: p. 206-220.
149. Junejo, A.K., N. Komninos, and J.A. McCann, *A Secure Integrated Framework for Fog-Assisted Internet of Things Systems.* IEEE Internet of Things Journal, 2020.
150. Liu, Y., J. Zhang, and J. Zhan, *Privacy protection for fog computing and the internet of things data based on blockchain.* Cluster Computing, 2020: p. 1-15.
151. Gu, K., et al., *Reusable Mesh Signature Scheme for Protecting Identity Privacy of IoT Devices.* Sensors, 2020. **20**(3): p. 758.

152. Perera, C., et al., *Designing privacy-aware internet of things applications.* Information Sciences, 2020. **512**: p. 238-257.
153. Qu, Y., et al., *Decentralized privacy using blockchain-enabled federated learning in fog computing.* IEEE Internet of Things Journal, 2020. **7**(6): p. 5171-5183.
154. Baniata, H., A. Anaqreh, and A. Kertesz, *PF-BTS: A Privacy-Aware Fog-enhanced Blockchain-assisted task scheduling.* Information Processing & Management, 2021. **58**(1): p. 102393.
155. Arif, M., et al., *Sdn based communications privacy-preserving architecture for vanets using fog computing.* Vehicular Communications, 2020. **26**: p. 100265.
156. Deebak, B. and F. Al-Turjman, *Robust Lightweight Privacy-Preserving and Session Scheme Interrogation for Fog Computing Systems.* Journal of Information Security and Applications, 2021. **58**: p. 102689.
157. Razaq, M.M., et al., *Privacy-Aware Collaborative Task Offloading in Fog Computing.* IEEE Transactions on Computational Social Systems, 2021.
158. Zhou, C., et al., *Privacy-preserving federated learning in fog computing.* IEEE Internet of Things Journal, 2020. **7**(11): p. 10782-10793.
159. Chen, S., et al., *Efficient privacy preserving data collection and computation offloading for fog-assisted IoT.* IEEE Transactions on Sustainable Computing, 2020. **5**(4): p. 526-540.
160. Shen, X., et al., *A privacy-preserving data aggregation scheme for dynamic groups in fog computing.* Information Sciences, 2020. **514**: p. 118-130.
161. Zhao, S., et al., *Smart and Practical Privacy-Preserving Data Aggregation for Fog-Based Smart Grids.* IEEE Transactions on Information Forensics and Security, 2020. **16**: p. 521-536.
162. Li, W., et al., *Using Granule to Search Privacy Preserving Voice in Home IoT Systems.* IEEE Access, 2020.
163. Tang, W., et al., *Functional Privacy-preserving Outsourcing Scheme with Computation Verifiability in Fog Computing.* KSII Transactions on Internet & Information Systems, 2020. **14**(1).
164. Mohanty, S.N., et al., *An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy.* Future Generation Computer Systems, 2020. **102**: p. 1027-1037.
165. Wang, Z., et al., *LiPSG: Lightweight Privacy-Preserving Q-Learning Based Energy Management for the IoT-Enable Smart Grid.* IEEE Internet of Things Journal, 2020.
166. Li, H., D. Han, and M. Tang, *A Privacy-Preserving Charging Scheme for Electric Vehicles Using Blockchain and Fog Computing.* IEEE Systems Journal, 2020.
167. Qi, L., et al., *Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment.* IEEE Transactions on Industrial Informatics, 2020.
168. Kong, Q., et al., *Privacy-preserving continuous data collection for predictive maintenance in vehicular fog-cloud.* IEEE Transactions on Intelligent Transportation Systems, 2020.
169. Baniata, H., W. Almobaideen, and A. Kertesz. *A privacy preserving model for fog-enabled mcc systems using 5g connection*. in *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. 2020. IEEE.
170. Kong, Q., L. Su, and M. Ma, *Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain.* IEEE Transactions on Intelligent Transportation Systems, 2020.
171. Zeng, B., et al., *BRAKE: Bilateral Privacy-Preserving and Accurate Task Assignment in Fog-Assisted Mobile Crowdsensing.* IEEE Systems Journal, 2020.
172. Eckhoff, D. and I. Wagner, *Privacy in the Smart City—Applications, Technologies, Challenges, and Solutions.* IEEE Communications Surveys & Tutorials, 2017. **20**(1): p. 489-516.

173. Sun, L., et al., *Understanding metropolitan patterns of daily encounters.* Proceedings of the National Academy of Sciences, 2013. **110**(34): p. 13774-13779.
174. Iliopoulou, C.A., et al., *Identifying spatio-temporal patterns of bus bunching in urban networks.* Journal of Intelligent Transportation Systems, 2020. **24**(4): p. 365-382.
175. Liu, Y., et al., *Social sensing: A new approach to understanding our socioeconomic environments.* Annals of the Association of American Geographers, 2015. **105**(3): p. 512-530.
176. Huang, J., et al., *Tracking job and housing dynamics with smartcard data.* Proceedings of the National Academy of Sciences, 2018. **115**(50): p. 12710-12715.
177. Luo, S., J. Jin, and J. Li, *A smart fridge with an ability to enhance health and enable better nutrition.* International Journal of Multimedia and Ubiquitous Engineering, 2009. **4**(2): p. 69-80.
178. Bosma, H., *Increasing nutrient awareness with the Smart Kitchen Scale*. 2020, University of Twente.
179. Ståhlbröst, A., et al., *Design of smart city systems from a privacy perspective.* IADIS International Journal on WWW/Internet, 2015. **13**(1): p. 1-16.
180. Memos, V.A., et al., *An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework.* 2018. **83**: p. 619-628.
181. Gupta, B., D.P. Agrawal, and S. Yamaguchi, *Handbook of research on modern cryptographic solutions for computer and cyber security*. 2016: IGI Global.
182. Dehkordi, S.A., et al., *A survey on data aggregation techniques in IoT sensor networks.* Wireless Networks, 2020. **26**(2): p. 1243-1263.
183. Randhawa, S. and S. Jain, *Energy-Efficient Fuzzy-Logic-Based Data Aggregation in Wireless Sensor Networks*, in *Information and Communication Technology for Sustainable Development*. 2020, Springer. p. 739-748.
184. Rani, S. and P. Saini, *Fog computing: applications and secure data aggregation*, in *Handbook of computer networks and cyber security*. 2020, Springer. p. 475-492.
185. Gupta, B., S. Rana, and A. Sharma. *An Efficient Data Aggregation Approach for Prolonging Lifetime of Wireless Sensor Network*. in *International Conference on Innovative Computing and Communications*. 2020. Springer.
186. Shobana, M., R. Sabitha, and S. Karthik, *An enhanced soft computing-based formulation for secure data aggregation and efficient data processing in large-scale wireless sensor network.* Soft Computing, 2020: p. 1-12.
187. Hu, P., et al., *Efficient Location Privacy-Preserving Range Query Scheme for Vehicle Sensing Systems.* Journal of Systems Architecture, 2020: p. 101714.
188. Babu, S.S. and K. Balasubadra, *Revamping data access privacy preservation method against inside attacks in wireless sensor networks.* Cluster Computing, 2019. **22**(1): p. 65-75.
189. Hathaliya, J.J. and S. Tanwar, *An exhaustive survey on security and privacy issues in Healthcare 4.0.* Computer Communications, 2020. **153**: p. 311-335.
190. Puthal, D., et al., *Fog computing security challenges and future directions [energy and security].* IEEE Consumer Electronics Magazine, 2019. **8**(3): p. 92-96.
191. Chakraborty, B., S. Verma, and K.P. Singh, *Temporal Differential Privacy in wireless sensor networks.* Journal of Network and Computer Applications, 2020: p. 102548.
192. Danezis, G. *The traffic analysis of continuous-time mixes*. in *International Workshop on Privacy Enhancing Technologies*. 2004. Springer.
193. Gruteser, M. and D. Grunwald. *Anonymous usage of location-based services through spatial and temporal cloaking*. in *Proceedings of the 1st international conference on Mobile systems, applications and services*. 2003.
194. Mehta, K., D. Liu, and M. Wright, *Protecting location privacy in sensor networks against a global eavesdropper.* IEEE Transactions on Mobile Computing, 2011. **11**(2): p. 320-336.

195. Kamat, P., et al. *Temporal privacy in wireless sensor networks*. in *27th International Conference on Distributed Computing Systems (ICDCS'07)*. 2007. IEEE.
196. Yang, Y., et al. *Towards event source unobservability with minimum network traffic in sensor networks*. in *Proceedings of the first ACM conference on Wireless network security*. 2008.
197. Xu, M., et al. *Dynamic and Disjoint Routing Mechanism for Protecting Source Location Privacy in WSNs*. in *2019 15th International Conference on Computational Intelligence and Security (CIS)*. 2019. IEEE.
198. Mutalemwa, L.C. and S. Shin, *Achieving source location privacy protection in monitoring wireless sensor networks through proxy node routing.* Sensors, 2019. **19**(5): p. 1037.
199. Ozturk, C., Y. Zhang, and W. Trappe. *Source-location privacy in energy-constrained sensor network routing*. in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. 2004.
200. Kamat, P., et al. *Enhancing source-location privacy in sensor network routing*. in *25th IEEE international conference on distributed computing systems (ICDCS'05)*. 2005. IEEE.
201. Baroutis, N. and M. Younis, *Location Privacy in Wireless Sensor Networks*, in *Mission-Oriented Sensor Networks and Systems: Art and Science*. 2019, Springer. p. 669-714.
202. Hamad, S.A., et al., *Realizing an internet of secure things: A survey on issues and enabling technologies.* IEEE Communications Surveys & Tutorials, 2020. **22**(2): p. 1372-1391.
203. Ali, Z., et al., *A robust authentication and access control protocol for securing wireless healthcare sensor networks.* Journal of Information Security and Applications, 2020. **52**: p. 102502.
204. Henze, M., et al., *A comprehensive approach to privacy in the cloud-based Internet of Things.* 2016. **56**: p. 701-718.
205. Caballero, V., et al., *Ontology-Defined Middleware for Internet of Things Architectures.* Sensors, 2019. **19**(5): p. 1163.
206. Singh, J., et al. *Big ideas paper: Policy-driven middleware for a legally-compliant Internet of Things*. in *Proceedings of the 17th International Middleware Conference*. 2016.
207. Abed, A.A. *Internet of Things (IoT): architecture and design*. in *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*. 2016. IEEE.
208. Srinivas, J., et al., *Provably secure biometric based authentication and key agreement protocol for wireless sensor networks.* 2017: p. 1-21.
209. Song, W., et al., *A privacy-preserved full-text retrieval algorithm over encrypted data for cloud storage applications.* 2017. **99**: p. 14-27.
210. Wang, J., et al. *A scalable and privacy-aware IoT service for live video analytics*. in *Proceedings of the 8th ACM on Multimedia Systems Conference*. 2017.
211. Hasan, R., et al. *Cartooning for enhanced privacy in lifelogging and streaming videos*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017.
212. Zhang, T. and Q. Zhu, *Dynamic differential privacy for ADMM-based distributed classification learning.* IEEE Transactions on Information Forensics and Security, 2017. **12**(1): p. 172-187.
213. Ambrosin, M., et al., *On the feasibility of attribute-based encryption on internet of things devices.* IEEE Micro, 2016. **36**(6): p. 25-35.
214. Gope, P. and T.J.I.T.I.E. Hwang, *A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-Time Application Data Access in Wireless Sensor Networks.* 2016. **63**(11): p. 7124-7132.
215. Tewari, A. and B.J.T.J.o.S. Gupta, *Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags.* 2017. **73**(3): p. 1085-1102.

216. Sutrala, A.K., et al., *Secure anonymity-preserving password-based user authentication and session key agreement scheme for telecare medicine information systems.* 2016. **135**: p. 167-185.
217. Khan, Z., Z. Pervez, and A.G.J.F.G.C.S. Abbasi, *Towards a secure service provisioning framework in a smart city environment.* 2017. **77**: p. 112-135.
218. Kapusta, K., G. Memmi, and H. Noura, *Additively homomorphic encryption and fragmentation scheme for data aggregation inside unattended wireless sensor networks.* Annals of Telecommunications, 2019: p. 1-9.
219. Li, L., et al., *Flexible and Secure Data Transmission System based on Semi-Tensor Compressive Sensing in Wireless Body Area Networks.* IEEE Internet of Things Journal, 2018.
220. Gilbert, E.P.K., et al., *Trust based data prediction, aggregation and reconstruction using compressed sensing for clustered wireless sensor networks.* Computers & Electrical Engineering, 2018.
221. Troncoso-Pastoriza, J.R., D. Gonzalez-Jimenez, and F. Perez-Gonzalez, *Fully private noninteractive face verification.* IEEE Transactions on Information Forensics and Security, 2013. **8**(7): p. 1101-1114.
222. Wu, Q., et al., *Privacy-aware multipath video caching for content-centric networks.* IEEE Journal on Selected Areas in Communications, 2016. **34**(8): p. 2219-2230.
223. Marvin, S., et al., *Urban Living Labs: Experimenting with City Futures*. 2018: Routledge.
224. Korayem, M., et al. *Enhancing lifelogging privacy by detecting screens*. in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016.
225. Gope, P., et al., *Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment.* Future Generation Computer Systems, 2018. **83**: p. 629-637.
226. Li, R., et al., *IoT applications on secure smart shopping system.* IEEE Internet of Things Journal, 2017. **4**(6): p. 1945-1954.
227. Shao, M., et al. *Cross-layer enhanced source location privacy in sensor networks*. in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. 2009. IEEE.
228. Lin, X. and X. Li, *Achieving efficient cooperative message authentication in vehicular ad hoc networks.* IEEE Transactions on Vehicular Technology, 2013. **62**(7): p. 3339-3348.
229. Huang, Q., Y. Yang, and L.J.I.A. Wang, *Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things.* 2017. **5**: p. 12941-12950.
230. Chen, J.-h., et al., *Lightning location system and lightning detection network of China power grid.* High Voltage Engineering, 2008. **34**(3): p. 425-431.
231. Basudan, S., X. Lin, and K. Sankaranarayanan, *A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing.* IEEE Internet of Things Journal, 2017. **4**(3): p. 772-782.
232. Gheisari, M., G. Wang, and S. Chen, *An edge computing-enhanced internet of things framework for privacy-preserving in smart city.* Computers & Electrical Engineering, 2020. **81**: p. 106504.
233. Ma, Z., et al., *Lightweight privacy-preserving medical diagnosis in edge computing.* IEEE Transactions on Services Computing, 2020.
234. Zhao, P., et al., *P 3: Privacy-Preserving Scheme Against Poisoning Attacks in Mobile-Edge Computing.* IEEE Transactions on Computational Social Systems, 2020. **7**(3): p. 818-826.
235. Zhao, Y., et al., *Privacy-preserving blockchain-based federated learning for IoT devices.* IEEE Internet of Things Journal, 2020.
236. Bao, H. and R. Lu, *A lightweight data aggregation scheme achieving privacy preservation and data integrity with differential privacy and fault tolerance.* Peer-to-Peer Networking and Applications, 2017. **10**(1): p. 106-121.

237. Bao, H. and R. Lu, *Comment on "privacy-enhanced data aggregation scheme against internal attackers in smart grid".* IEEE Transactions on Industrial Informatics, 2016. **12**(1): p. 2-5.
238. Chen, L., R. Lu, and Z. Cao, *PDAFT: A privacy-preserving data aggregation scheme with fault tolerance for smart grid communications.* Peer-to-peer networking and applications, 2015. **8**(6): p. 1122-1132.
239. Xiao, M., et al., *A hybrid scheme for fine-grained search and access authorization in fog computing environment.* Sensors, 2017. **17**(6): p. 1423.
240. Mitton, N., et al., *Combining Cloud and sensors in a smart city environment*. 2012, Springer.
241. Cui, J., et al., *Edge computing in VANETs-an efficient and privacy-preserving cooperative downloading scheme.* IEEE Journal on Selected Areas in Communications, 2020. **38**(6): p. 1191-1204.
242. Bi, M., et al., *A privacy-preserving mechanism based on local differential privacy in edge computing.* China Communications, 2020. **17**(9): p. 50-65.
243. Mitrokotsa, A., C. Onete, and S. Vaudenay, *Location leakage in distance bounding: Why location privacy does not work.* Computers & Security, 2014. **45**: p. 199-209.
244. Yeh, L.-Y., et al., *Cloud-Based Fine-Grained Health Information Access Control Framework for LightweightIoT Devices with Dynamic Auditing andAttribute Revocation.* IEEE transactions on cloud computing, 2018. **6**(2): p. 532-544.
245. Ab Rahman, N.H., et al., *Forensic-by-design framework for cyber-physical cloud systems.* IEEE Cloud Computing, 2016. **3**(1): p. 50-59.
246. Wu, F., et al., *A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications.* Journal of Ambient Intelligence and Humanized Computing, 2017.
247. Shen, H., M. Zhang, and J. Shen, *Efficient Privacy-Preserving Cube-Data Aggregation Scheme for Smart Grids.* IEEE Trans. Information Forensics and Security, 2017. **12**(6): p. 1369-1381.
248. Figueres, N.B., et al., *Efficient smart metering based on homomorphic encryption.* Computer Communications, 2016. **82**: p. 95-101.
249. Ivaşcu, T., M. Frîncu, and V. Negru. *Considerations towards security and privacy in Internet of Things based eHealth applications*. in *Intelligent Systems and Informatics (SISY), 2016 IEEE 14th International Symposium on*. 2016. IEEE.
250. Mahmood, K., et al., *An elliptic curve cryptography based lightweight authentication scheme for smart grid communication.* Future Generation Comp. Syst., 2018. **81**: p. 557-565.
251. Challa, S., et al., *An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks.* Computers & Electrical Engineering, 2017.
252. Song, T., et al., *A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes.* IEEE Internet of Things Journal, 2017. **4**(6): p. 1844-1852.
253. Chifor, B.-C., et al., *A security authorization scheme for smart home Internet of Things devices.* Future Generation Computer Systems, 2017.
254. Amin, R., et al., *A robust and anonymous patient monitoring system using wireless medical sensor networks.* Future Generation Comp. Syst., 2018. **80**: p. 483-495.
255. Askoxylakis, I., et al., *Computer Security–ESORICS 2016*. 2016: Springer.
256. Peng, M., et al., *Fog-computing-based radio access networks: issues and challenges.* Ieee Network, 2016. **30**(4): p. 46-53.
257. Peng, M., et al., *Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies.* IEEE Wireless Communications, 2014. **21**(6): p. 126-135.
258. Peng, M., et al., *System architecture and key technologies for 5G heterogeneous cloud radio access networks.* IEEE network, 2015. **29**(2): p. 6-14.

259. Liu, J., et al., *Secure intelligent traffic light control using fog computing.* Future Generation Computer Systems, 2018. **78**: p. 817-824.
260. Yang, R., et al., *Position based cryptography with location privacy: A step for fog computing.* Future Generation Computer Systems, 2018. **78**: p. 799-806.
261. Ali, B., M.A. Gregory, and S. Li, *Multi-Access Edge Computing Architecture, Data Security and Privacy: A Review.* IEEE Access, 2021.
262. Suri, B., et al. *Peering through the fog: an inter-fog communication approach for computing environment*. in *International conference on innovative computing and communications*. 2019. Springer.
263. Mostafavi, S. and W. Shafik, *Fog Computing Architectures, Privacy and Security Solutions.* Journal of Communications Technology, Electronics and Computer Science, 2019. **24**: p. 1-14.
264. Dubey, P. *If the IoT isn't secure, people won't use it*. 2016; Available from: https://inform.tmforum.org/internet-of-everything/2016/09/iot-isnt-secure-people-wont-use/.
265. Ruj, S. and A. Nayak, *A decentralized security framework for data aggregation and access control in smart grids.* IEEE transactions on smart grid, 2013. **4**(1): p. 196-205.
266. Chen, L., et al., *MuDA: Multifunctional data aggregation in privacy-preserving smart grid communications.* Peer-to-peer networking and applications, 2015. **8**(5): p. 777-792.
267. Bae, M., K. Kim, and H. Kim, *Preserving privacy and efficiency in data communication and aggregation for AMI network.* Journal of Network and Computer Applications, 2016. **59**: p. 333-344.
268. Fan, C.-I., S.-Y. Huang, and Y.-L. Lai, *Privacy-enhanced data aggregation scheme against internal attackers in smart grid.* IEEE Transactions on Industrial informatics, 2014. **10**(1): p. 666-675.
269. Guan, Z. and G. Si, *Achieving privacy-preserving big data aggregation with fault tolerance in smart grid.* Digital Communications and Networks, 2017. **3**(4): p. 242-249.
270. Tonyali, S., et al., *Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled Smart Metering systems.* Future Generation Comp. Syst., 2018. **78**: p. 547-557.
271. Badra, M. and S. Zeadally, *Lightweight and efficient privacy-preserving data aggregation approach for the Smart Grid.* Ad Hoc Networks, 2017. **64**: p. 32-40.
272. Zhang, J., et al., *LVPDA: A Lightweight and Verifiable Privacy-Preserving Data Aggregation Scheme for Edge-Enabled IoT.* IEEE Internet of Things Journal, 2020. **7**(5): p. 4016-4027.
273. Brust, M.R., H. Frey, and S. Rothkugel. *Dynamic multi-hop clustering for mobile hybrid wireless networks*. in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. 2008. ACM.
274. Jhaveri, R.H., et al., *Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in industrial IoT.* IEEE Access, 2018. **6**: p. 20085-20103.
275. Wang, X., et al., *A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems.* IEEE Transactions on Computational Social Systems, 2018. **5**(2): p. 481-492.
276. Chang, Z., et al., *Dynamic resource allocation and computation offloading for IoT fog computing system.* IEEE Transactions on Industrial Informatics, 2020.
277. Wang, T., et al., *Edge-based differential privacy computing for sensor–cloud systems.* Journal of Parallel and Distributed computing, 2020. **136**: p. 75-85.
278. Kemp, R., et al. *Cuckoo: a computation offloading framework for smartphones*. in *International Conference on Mobile Computing, Applications, and Services*. 2010. Springer.

279. Wang, Q. and S. Chen, *Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks.* Transactions on Emerging Telecommunications Technologies, 2020. **31**(12): p. e3880.
280. Yang, M., et al., *Machine learning differential privacy with multifunctional aggregation in a fog computing architecture.* IEEE Access, 2018. **6**: p. 17119-17129.
281. Pan, J., et al. *A Novel Fog Node Aggregation Approach for Users in Fog Computing Environment*. in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. 2020. IEEE.
282. Gu, F., et al., *Partitioning and offloading in smart mobile devices for mobile cloud computing: State of the art and future directions.* Journal of Network and Computer Applications, 2018. **119**: p. 83-96.
283. Bhattacharya, A. and P. De, *A survey of adaptation techniques in computation offloading.* Journal of Network and Computer Applications, 2017. **78**: p. 97-115.
284. Yousefpour, A., G. Ishigaki, and J.P. Jue. *Fog computing: Towards minimizing delay in the internet of things*. in *2017 IEEE international conference on edge computing (EDGE)*. 2017. IEEE.
285. Yousefpour, A., et al., *FogPlan: a lightweight QoS-aware dynamic fog service provisioning framework.* IEEE Internet of Things Journal, 2019. **6**(3): p. 5080-5096.
286. Yousefpour, A., et al., *On reducing IoT service delay via fog offloading.* IEEE Internet of Things Journal, 2018. **5**(2): p. 998-1010.
287. Deng, R., et al., *Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption.* IEEE internet of things journal, 2016. **3**(6): p. 1171-1181.
288. Liu, L., et al., *Multiobjective optimization for computation offloading in fog computing.* IEEE Internet of Things Journal, 2017. **5**(1): p. 283-294.
289. Jiang, Y.-L., et al., *Energy-efficient task offloading for time-sensitive applications in fog computing.* IEEE Systems Journal, 2018. **13**(3): p. 2930-2941.
290. Naqvi, S.A.A., et al. *Metaheuristic optimization technique for load balancing in cloud-fog environment integrated with smart grid*. in *International Conference on Network-Based Information Systems*. 2018. Springer.
291. Binh, H.T.T., et al. *An evolutionary algorithm for solving task scheduling problem in cloud-fog computing environment*. in *Proceedings of the Ninth International Symposium on Information and Communication Technology*. 2018.
292. Hussein, M.K. and M.H. Mousa, *Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization.* IEEE Access, 2020. **8**: p. 37191-37201.
293. Peng, K., et al., *An energy-and cost-aware computation offloading method for workflow applications in mobile edge computing.* EURASIP Journal on Wireless Communications and Networking, 2019. **2019**(1): p. 207.
294. Cui, L., et al., *Joint optimization of energy consumption and latency in mobile edge computing for Internet of Things.* IEEE Internet of Things Journal, 2018. **6**(3): p. 4791-4803.
295. Mao, Y., J. Zhang, and K.B. Letaief, *Dynamic computation offloading for mobile-edge computing with energy harvesting devices.* IEEE Journal on Selected Areas in Communications, 2016. **34**(12): p. 3590-3605.
296. Lei, K., et al., *Groupchain: Towards a Scalable Public Blockchain in Fog Computing of IoT Services Computing.* IEEE Transactions on Services Computing, 2020. **13**(2): p. 252-262.
297. Madi, M.K., et al. *A novel dynamic replica creation mechanism for Data Grids*. in *2015 Game Physics and Mechanics International Conference (GAMEPEC)*. 2015. IEEE.
298. Buccafurri, F., et al., *A privacy-preserving localization service for assisted living facilities.* IEEE Transactions on Services Computing, 2016.

299. Tos, U., et al., *Ensuring performance and provider profit through data replication in cloud systems.* Cluster Computing, 2018. **21**(3): p. 1479-1492.
300. Zhao, Y., et al. *Dynamic replica creation strategy based on file heat and node load in hybrid cloud*. in *2017 19th International Conference on Advanced Communication Technology (ICACT)*. 2017. IEEE.
301. Dai, W., I. Ibrahim, and M. Bassiouni. *An improved replica placement policy for Hadoop distributed file system running on cloud platforms*. in *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)*. 2017. IEEE.
302. Qu, K., L. Meng, and Y. Yang. *A dynamic replica strategy based on Markov model for hadoop distributed file system (HDFS)*. in *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*. 2016. IEEE.
303. Nivetha, N. and D. Vijayakumar. *Modeling fuzzy based replication algorithm to improve data availability in cloud datacenter*. in *International Conference on Computing Technologies & Intelligent Data Engineering, Kovilpatti, India. IEEE Computer Society, USA*. 2016.
304. Edwin, E.B., P. Umamaheswari, and M.R. Thanka, *An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center.* Cluster Computing, 2019. **22**(5): p. 11119-11128.
305. Mansouri, N., M.K. Rafsanjani, and M.M. Javidi, *DPRS: A dynamic popularity aware replication strategy with parallel download scheme in cloud environments.* Simulation Modelling Practice and Theory, 2017. **77**: p. 177-196.
306. Zhang, H., et al. *Data replication placement strategy based on bidding mode for cloud storage cluster*. in *2014 11th Web Information System and Application Conference*. 2014. IEEE.
307. Cui, L., et al., *A genetic algorithm based data replica placement strategy for scientific applications in clouds.* IEEE Transactions on Services Computing, 2015. **11**(4): p. 727-739.
308. Liu, X., et al., *An efficient privacy-preserving outsourced computation over public data.* IEEE Transactions on Services Computing, 2015. **10**(5): p. 756-770.
309. Li, C., Y. Zhang, and Y. Luo, *Adaptive Replica Creation and Selection Strategies for Latency-Aware Application in Collaborative Edge-Cloud System.* The Computer Journal, 2019.
310. Li, C., et al., *Scalable and dynamic replica consistency maintenance for edge-cloud system.* Future Generation Computer Systems, 2019. **101**: p. 590-604.
311. Hyndman, R.J. and G. Athanasopoulos, *Forecasting: principles and practice*. 2018: OTexts.
312. Shao, Y., et al., *Cost-effective replication management and scheduling in edge computing.* Journal of Network and Computer Applications, 2019. **129**: p. 46-61.