A Comprehensive Review on Secure IoT-Based E-Examination Systems Using Fog Computing

MR. Pradeep Nayak, Keerthana A[2], Karthika[3], Keerthana S[4], Keziya Mary[5]
[1]Faculty, Information Science and Engineering (ISE), Alvas Institute of Engineering and Technology, Karnataka, India
[2]Student, Computer science and Engineering (Internet of Things, Cybersecurity Including Blockchain Technology), Alvas Institute of Engineering and Technology, Karnataka, India
[3] Student, Computer science and Engineering (Internet of Things, Cybersecurity Including Blockchain Technology), Alvas Institute of Engineering and Technology, Karnataka, India
[4] Student, Computer science and Engineering (Internet of Things, Cybersecurity Including Blockchain Technology), Alvas Institute of Engineering and Technology, Karnataka, India
[5] Student, Computer science and Engineering (Internet of Things, Cybersecurity Including Blockchain Technology), Alvas Institute of Engineering and Technology, Karnataka, India

Emails: pradeep@aiet.org.in[1], keerthanaa0765@gmail.com[2], karthikakarthika4960@gmail.com[3], keerthanas0618@gmail.com[4] , keziamary1416@gmail.com[5]

Abstract

E-learning systems are in high demand, according to a recent important confirmation. Examiners, teachers, and students exchange a lot of data, which needs to be sent safely. The Internet of Things (IoT) is one of the most supportive infrastructures for combining cybersecurity, data analysis, and intelligent devices. In particular, when paired with fog and cloud potentials to improve the efficiency of different computing-intensive and latency-sensitive applications. The IoT-Fog-Cloud framework is presented in this paper to provide security factors for sharing E-exams, which present a number of security challenges, including fine-grained access control and E-exam security preservation. Additionally, the suggested framework is in favour of bringing services closer to the students. Additionally, by offloading a portion of the encryption cost to fog servers, this paper increases the efficiency of E-exam data analysis, lowers the encryption burden on user devices in terms of computation costs, and offers fine-grained access control to E-exam content through encryption using various cryptographic techniques. The two primary components of the IoT–fog–cloud framework are the layer processes and the layer components. The FGNs, cloud data centers , and GFNs are among the layer components that need to be integrated. A number of advantages can be obtained in layer processes because of distribution processes. This paper concludes by demonstrating that the suggested IoT-Fog-Cloud framework can guarantee secure procedures for implementing the framework by achieving data confidentiality, fine-grained access control, collusion resistance, and unforgeability.

Keywords: *fog computing, e-learning, Internet of Things (IoTs), e-exams, and security systems.*

1. INTRODUCTION

For clients using Web applications and batch processing, the "pay-as-you-go" cloud computing model is a cost-effective substitute for owning and operating private data centers (DCs). Mega DCs' economy of scale is a result of a number of factors, including lower OPEX from the deployment of homogeneous compute, storage, and networking components; a convenient location that takes advantage of cheap power; and higher predictability of massive aggregation, which permits higher utilization without compromising performance. Businesses and end users are liberated from specifying numerous details thanks to cloud computing. This joy turns into an issue for applications that are sensitive to latency, which need All or a portion of this work may be reproduced in hard copy or digital format for personal or educational purposes without charge, as long as copies are not made or distributed for financial gain or commercial advantage and include this notice and the complete citation on the first page. Any other kind of copying, republishing, posting on servers, or redistributing to lists necessitates prior specific permission and/or payment. nodes nearby to satisfy their delay needs. In addition to location awareness and low latency, a new wave of Internet deployments, particularly the Internet of Things (IoTs), also need mobility support and geo-distribution. We contend that in order to satisfy these demands, a new platform is required, which we refer to as Fog Computing [1], or simply Fog because the fog is a cloud near the ground. Additionally, we assert that Fog Computing facilitates a new breed of applications and services rather than consuming Cloud Computing, and that the two technologies work well together, especially in the areas of data management and analytics. The structure of this paper is as follows. We outline the features of the platform that supports Fog services as well as the Fog Computing paradigm in the second section. The section that follows closely examines a few noteworthy applications and services that support our claim that the fog is the essential platform component needed to support the Internet of Things. The fourth section looks at big data and analytics in relation to interesting applications. The understanding that some of these applications require both long-term global data mining and real-time analytics demonstrates how Fog and Cloud interact and play complementary roles. We wrap up by discussing upcoming work and offering remarks on the current state of fog computing.

2. THE FOG COMPUTING PLATFORM

2.1Characterization of Fog Computing

Fog computing is a highly virtualized platform that connects endpoints to conventional Cloud Computing Data Centers, usually but not always at the network edge, and offers networking, storage, and computation services. The idealized computing and information architecture supporting the upcoming Internet of Things applications is shown in Figure 1, which also highlights the function of fog computing. The foundation of both the Cloud and the Fog are computing, storage, and networking resources. However, the term "Edge of the Network" suggests several traits that make the Fog a more complex extension of the Cloud. Let's enumerate them along with suggestions for inspiring examples. • Low latency, location awareness, and edge location. Early plans to support endpoints with rich services at the network's edge, such as applications requiring low latency (e.g. gaming, video streaming, augmented reality), are where the Fog got its start. The three layers of Fog Computing's architecture are the edge layer, which houses sensors and actuators that gather and send data, the fog layer, which houses local devices, and the cloud layer, which manages intensive processing and long-term storage. Additionally, Fog Computing enhances system reliability and scalability by distributing computational tasks closer to end-users. This reduces bandwidth usage and ensures faster decision-making where real-time responses are crucial.

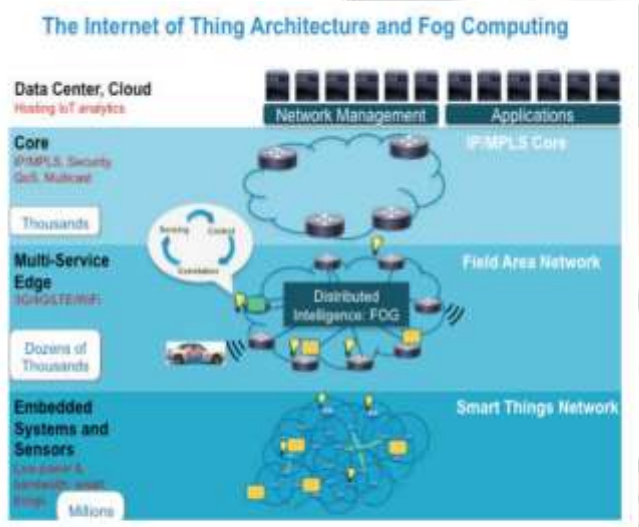


Figure 1: The Internet of Things and Fog Computing

• The distribution of geography. The services and apps that the Fog targets require widely dispersed deployments, which stands in stark contrast to the more centralized Cloud. For example, the Fog will actively contribute to providing high-quality streaming to moving automobiles via access points and proxies placed alongside roads and railroads.

• Other examples of systems that are inherently distributed and require distributed computing and storage resources include the Smart Grid and extensive sensor networks for environmental monitoring.

• A very high number of nodes, as demonstrated by sensor networks in general and the Smart Grid in particular, as a result of the wide geo-distribution.
• Assistance with mobility. Since direct communication with mobile devices is crucial for many Fog applications, mobility techniques like the LISP protocol 1 that separate host identity from location identity and call for a distributed directory system must be supported.

• Real-time interactions. Important Fog applications involve real-time interactions rather than batch processing.

•Wireless access is more common.
• Variability. Fog nodes will be used in a wide range of environments and have a variety of form factors.
• Federation and interoperability. The collaboration of various providers is necessary for the smooth operation of some services (streaming is a prime example). As a result, services must be federated across domains and Fog components must be able to work together.
• Assistance with online analysis and cloud interaction. Near the source, the Fog is positioned to be crucial to the ingestion and processing of the data. In section 1, we go into more detail about how Fog and Cloud interact with regard to big data.

2.2 Fog Players: Providers and Users

Figure 4. Challenges of fog computing.

At this early stage, it is difficult to predict how the various Fog Computing players will align. However, given the characteristics of the main services and applications, we expect that:
• The Fog (Infotainment in Connected Vehicle, Smart Grid, Smart Cities, Health Care, etc.) will heavily rely on subscriber models.
• New kinds of competition and collaboration between providers seeking to offer worldwide services will emerge as a result of the fog. Public administrations, transportation agencies, automakers, utilities, and other new incumbents will join the market as providers and consumers.

2. Challenges of Fog with the IoT

The fog computing paradigm presents numerous obstacles to its successful implementation, despite the fact that it offers numerous advantages for various Internet of Things applications. As the figure illustrates, these difficulties include scalability,complexity,dynamicity,heterogeneity, latency, and security. These challenges arise due to the distributed and decentralized nature of fog environments, where numerous devices operate at different layers with varying capacities.

(1) Scalability: With billions of IoT devices, a massive amount of data is generated and resources, including processing power and storage, are needed. As a result, fog servers ought to have enough resources to support each of these devices. The ability to adapt to the quick expansion of IoT applications and devices will be the true challenge.

(2) Complexity: Selecting the best parts is getting increasingly difficult because there are so many IoT devices and sensors made by various manufacturers, particularly when considering various hardware and software configurations as well as individual needs. Furthermore, certain applications with strict security requirements need particular hardware and protocols in order to operate, which makes operations more challenging.

(3) Dynamicity: The ability of IoT devices to adapt and dynamically alter the makeup of their workflow is one of their key characteristics. IoT devices' internal characteristics and functionality will change as this challenge. Furthermore, the aging of hardware and software in handheld devices causes changes in device properties and workflow behavior. Fog nodes will therefore require intelligent and automatic reconfiguration.

(4) Heterogeneity: Numerous IoT sensors and devices have been created by various manufacturers. These gadgets are equipped with a variety of sensors, computing power, storage, and communication radios. Choosing the right resources and managing and coordinating networks with such diverse IoT devices will become very difficult.

(5) Latency: Fog computing's low latency is a primary justification for replacing the cloud, particularly for applications that require quick responses. On fog computing platforms, a high latency of application or service performance can be attributed to a variety of factors. High latency fog will make users unhappy.

(6) Security: The fog environment itself is weaker and less secure than cloud computing, even though fog nodes will need to be protected using the same policies, controls, and procedures as well as the same physical security and cybersecurity solutions. The fog's mobility, heterogeneity, and extensive geo-distribution make it impossible to directly apply current cloud computing security and privacy metrics. Numerous studies concentrate on authentication and cryptography to strengthen network security and guard against cyberattacks in fog computing.

(7) Resource management: Network devices with extra storage and processing capacity are frequently referred to as fog end devices. Such devices, however, find it challenging to match the resource capacity of conventional servers, let alone the cloud. Therefore, for the fog computing environment to operate effectively, prudent management of fog resources is needed.

(8) Energy consumption: Numerous fog end devices are used in the fog environment; the distributed computation may use less energy than the centralized cloud computing model. As a result, lowering fog computing's energy consumption is a significant issue that requires attention.

4. Service Placement in Fog Computing

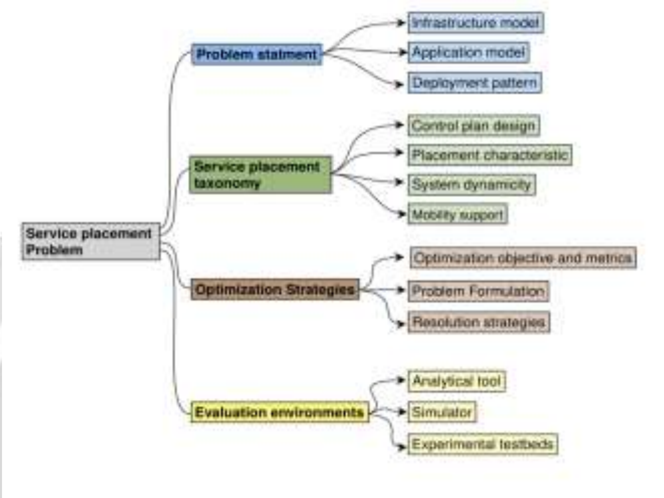
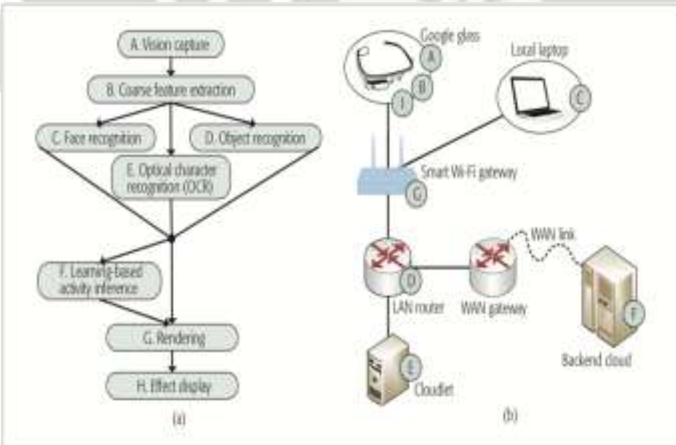


Figure 3: The structure of the section.

In this section, we propose to describe the methodology typically used to address the service placement problem (SPP) and provide an overview of the following aspects: problem statement, placement taxonomy, optimization strategies, and evaluation tools. The service placement problem has been extensively discussed in the literature, and a number of proposals have emerged.

4.1 Problem statement

The infrastructure model, the application model, and the deployment pattern with associated constraints are the three components that are described in the SPP problem statement.



4.1.1 Infrastructure model

Sensors and actuators are examples of devices without computational capabilities, while fog nodes and cloud data centers are examples of resources with computational power and/or storage capacity. The infrastructure network is typically represented as a connected graph, with the set of cloud servers, fog nodes, and Internet of Things devices represented by the vertices and the connections between the nodes by the edges.

4.1.2 Application(s) model

The following primary descriptions are identified based on the reviewed papers.

a)MonolithicService: A single, unified component that represents the entire application. In this model, all functionalities are combined into one service that directly interacts with end users or Internet of Things (IoT) devices.

b) Collection of Interconnected Services:
In this approach, the application is divided into multiple smaller services, each responsible for a specific function. These services communicate and work together to deliver the complete application's functionality.

c)InterconnectedGraph: In this model, the application is structured as a connected graph consisting of several interdependent components. Each component interacts with others to perform complex, distributed tasks across the network.

4.1.3 Deployment pattern

The mapping pattern by which links and components of applications are mapped onto an infrastructure graph (i.e., computing devices and physical edges) is defined by the application placement problem. An example of mapping a DAG-modeled application to available fog nodes is depicted in the figure.

Figure 4 : Cognitive assistance application [67], shown in (a), and deployed onto Fog network, shown in (b).

4.2 Service placement taxonomy

When creating deployment strategies, it is necessary to take into account certain specifics and criteria in order to address the SPP. As shown in Figure 5, we suggest in this paper to focus on the following four primary elements, which are represented as a service placement taxonomy.

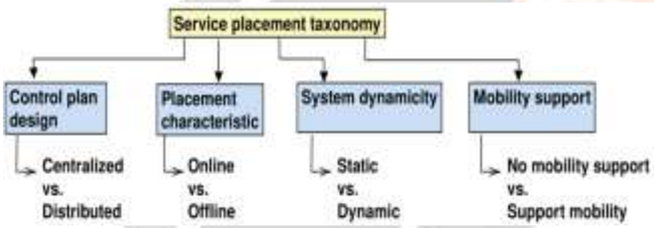


Figure 5: Service placement taxonomy.

4.3 Optimization strategies

First, we would like to offer a worldwide classification of optimization techniques that have been put forth in the literature. We differentiate between two types of optimization: mono-objective and multi-objective. Conversely, we offer the metrics that are most frequently taken into account when optimizing.

a) **Mono vs. Multi-objective optimization Mono-objective optimization**: suggests optimizing a single objective function. On the other hand, multi-objective suggests optimizing a number of objective functions at the same time [99]. Table 1 provides a preliminary classification of SPP solutions with respect to these two optimizations. Asterisks (*) indicate works that have examined both aspects (mono-objective and multi-objective).

Table 1: Classification regarding the optimization objectives.

b) Optimization metrics: In light of the works presented thus far, we present a list of optimization metrics that are typically taken into consideration in the literature. The following metrics can have their values maximized or minimized through optimization.

Optimization objective	References
Mono-objective	[10, 12, 14, 16, 17, 19, 20, 43, 44], [45]*, [46, 50, 57, 58, 62, 69, 71, 72, 75, 77, 79, 81, 86–89, 93, 94, 96, 97, 101–103, 107, 108, 115, 117, 127, 129, 132–135, 137, 139–141, 143, 145, 147–151, 154, 157, 161–164], [166]*, [167]
Multi-Objectives	[15, 36], [45]*, [98, 118, 122], [166]*

Metrics		References
Latency		[7, 12, 13, 15, 20, 26, 43, 50, 55, 58, 75, 85–87, 89, 98, 107, 108, 113, 115, 118, 119, 124, 137, 139, 140, 143, 147, 150, 154, 157, 161, 162, 166]
Resource utilization		[10, 15, 16, 71, 72, 81, 88, 88, 102, 117, 122, 133–135, 148]
Cost		[13–15, 17, 19, 28, 34, 36, 46, 57, 62, 69, 71, 79, 98, 103, 112, 114, 118, 132, 145, 149, 151, 156, 163, 168]
Energy consumption		[11, 15, 26, 43–45, 94, 94, 98, 101, 113, 122–124, 141]
Others	Quality-of-experience	[36, 55, 96, 127, 129]
	Congestion ratio	[93, 164, 167]
	Blocking probability / Failed requests / Number of computationally active FNs	[77] / [16] / [97]

Table 2: Optimization metrics

4.4 Evaluation environments

The research community tests their solutions in relevant environments, ideally in realistic setups, and conducts extensive experiments using various programming tools to assess the performance of their proposals. The most often used tools in the surveyed papers are shown below. The programming environment used in the literature is compiled in Table 3.

Environment	Evaluation Environment	References
Analytical tool	Java Tool	[29, 163]
	C++	[77, 126, 164]
	Matlab	[43, 44, 69, 79, 101]
	Optimisation engine (IBM CPLEX, Gurobi, Xpress-MP, ...)	[9, 17, 19, 34, 62, 108, 122, 133, 134, 137, 141, 164, 166]
	Others	[15, 43, 57, 103, 167]
Simulator	CloudSim	[118, 127, 135]
	iFogSim	[64, 78, 94, 96, 102, 107, 108, 108, 117, 131–134, 140, 166]
	SimGrid	[154]
	OMNeT++	[12, 26, 114]
	Others (FogTorchΠ, YAFS [90], ...)	[6, 7, 16, 31, 75, 81, 87, 87–89, 98, 103, 112, 115, 144, 145, 151, 167]
Testbed	Grid'5000 & Fit IOT-Lab	[46]
	OpenStack	[100]
	Adhoc testbed	[55, 58, 60, 71, 72, 149, 157]

Table 3: A Summary of evaluation environments.

5. Proposed IoT-FOG cloud framework

Framework Overview

The devices at the bottom of Figure 4 represent the Internet of Things (IoT) device layer, which can be deployed in a specific area, such as on students' portable devices in educational institutions. Each device is used to receive one electronic exam, and the student answers are sent to an aggregation point at the edge of the FC layer, which is the middle layer. Generally speaking, the proposed framework relies on IoT-based FC to improve the endpoint security, monitoring, and computation of IoT devices that students use to receive e-exams, such as laptops, smartphones, and tablets.

The FClayer contains the lowest level of energy-constrained devices. These devices can only react to queries gathered by the FC nodes at higher layers, and they are not appropriate for large-scale computing operations. Additionally, there are four different kinds of nodes in the fog layer: temporary fog storage nodes (TFSNs) and fog gateway nodes (FGNs).

5.1 Elements of the IoT-Fog cloud framework

Layer processes and layer components are the two main components of the integrated platform offered by the IoT-fog-cloud framework. Components of the Layer This section discusses in detail the components of the IoT, fog, and cloud framework: cloud datacenters, general fog nodes (GFNs), FGNs, and IoT devices. IoT devices, also referred to as portable gadgets, include laptops, tablets, and smartphones, which are the physical components that receive the e-examinations. These IoT devices can produce student responses to e-exams as raw data because of their constrained energy, computation, processing, and resource capacities. The IOT-fog-cloud framework enables IoT devices to establish connections with edge nodes via communication protocols such as Bluetooth.

Fog Organization nodes (FONs)

They are used for routine computation tasks and are made available via intermediary nodes, like the FON protection gateway. An intermediary mediator oversees the operations of FONs in distributed processing. and provide resources for e-exam solutions. Additionally, FONs use distinct clocks to identify the synchronous tasks relayed from several intermediary nodes. After that, the synchronous tasks are configured, and the response is sent to the intermediate nodes. Additionally, to guarantee application uniformity, FON sperforms only one application at a time. A repository for the collection and analysis of historical exam data is provided by TFSNs. They retain all application metadata, such as requirements, models, performance data, and soon. This meta-data can assist in completing any process in the event that problems arise. Additionally, each command, procedure, and piece of metadata is time-stamped and source-identified before being saved in a storage repository. Cloud data centers: The IoT-fog-cloud structure provides more scalable resources and computation if the fog layer has a wide range of computational services to perform tasks. As a result, IoT devices are better equipped to process and access the required storage on TFSNs, facilitating distribution that makes basic data access and analysis possible.

Software Processes The IoT-fog-cloud framework for processing e-examanswers offers several advantages, such as the capacity to carry out distributed processes in real-time, lower latency, respond quickly, enhance privacy and security, and scale, analyze, and filter data to deliver services with an extremely efficient architecture. This section describes the FC layers. The six layers—cloud deployment, security and privacy, data analysis, infrastructure/virtual, control, and temporal repository—are shown in Figure 5 in the following order, from bottom to top.

6. CONCLUSION

Our proposed protocols enable the deployment of fog-based IoT monitoring services in environments that are not trusted. We use the scalability and flexibility of ABE to plan the network that will provide the service, as well as the effectiveness of the symmetric cryptography used for the monitoring process, to balance security and low latency. When sensors are low-tier, energy-constrained IoT devices, the latter adds very little overhead to the system. Fog orchestration, on the other hand, eliminates the need to broadcast data updates and requests across the entire network.

In the future, we intend to investigate other fog-related architectures, like mist computing, and modify the orchestration protocol to accommodate other non-strictly hierarchical fog architectures taken into consideration in monitoring services (e.g., where fog nodes at the same level can communicate among themselves). Additionally, while preserving the network's hierarchical structure, we intend to investigate ways to accommodate mobile IoT devices (like wearables that can move between areas controlled by different fog nodes) and increase the suggested system's scalability in terms of the number of concurrent users by combining user requests for related services.

REFERENCE:

1. Dalia Khairy, Mohamed A, Amasha, Rania A, Abougalala, Salem Alkhalaf, and Marwa F, Areed. *A Security System for E-Exams Using an IoT and Fog Computing Environment***.** Published in Computers, Materials & Continua (CMC), Volume 71, Issue 1, 2022, pp.1497–1514.DOI: 10.32604/cmc.2022.019658.

2. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012).Fog Computing and Its Role in the Internet of Things. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, August 17, Helsinki, Finland, pp. 13–16.

3. Atlam, H. F., Walters, R. J., & Wills, G. B. (2018).
Fog Computing and the Internet of Things: A Review.
Big Data and Cognitive Computing, 2(2), 10.

4. Salaht, F. A., Desprez, F., & Lebre, A. (2020).
An Overview of the Service Placement Problem in Fog and Edge Computing. ACM Computing Surveys (CSUR), 53(3), 1–35.

5. Amor, A. B., Abid, M., & Meddeb, A. (2020). Secure Fog-Based E-Learning Scheme. IEEE Access, 8, 31920-31933.
6. Viejo, A., & Sánchez, D. (2020). Secure Monitoring in IoT-Based Services via Fog Orchestration.
Future Generation Computer Systems, 107, 443–457.