Security in Critical Communication for Mobile Edge Computing Based IoE Applications

Tanmoy Maitra, Debasis Giri, and Arup Sarkar

Abstract The new era of the Internet of Everything (IoE) applications demands low latency along with security into the networks. The cloud-based architecture alone cannot provide low response time to the users or mobile devices (like phone, laptop, sensors device, etc.). Therefore between mobile devices and cloud, edge devices (known as Fog device) are introduced as middleware device. From the edge devices, users can get information from local devices without interacting with the cloud via the Internet or radio. In such complicated networks, security preservation in communications becomes a challenging task. The security protocols for critical communication in such applications (e-medical, e-banking) are based on the architecture of the networks which can be centralized or distributed or hybrid (a mixture of centralized and distributed). This book chapter discusses the different security protocols in communications for the aforementioned architectures which can be designed for Mobile Edge Computing (MEC) based IoE applications. Moreover, this chapter covers (a) architectures and their security threats, (b) necessity of security model in such applications, (c) different secure communication protocols for those applications, (d) challenges to design security protocols to reduce response time, and latency (e) the future direction of this research domain which can be explored more.

Keywords Internet of Everything (IoE) · Edge device · Security · Communication protocol · Privacy

T. Maitra (✉) · A. Sarkar
School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India
e-mail: tanmoy.maitrafcs@kiit.ac.in; arup.sarkarfcs@kiit.ac.in

D. Giri
Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Nadia, West Bengal, India

A. Mukherjee et al. (eds.), *Mobile Edge Computing*,
https://doi.org/10.1007/978-3-030-69893-5_13

1 Introduction

According to Gartner [1], in 2015, the Internet of Everything (IoE) was recorded as one of the lid trends. IoE can be defined as it "*is bringing together people, process, data, and things to make networked connections more relevant and valuable than ever before-turning information into actions that create new capabilities, richer experiences, and unprecedented economic opportunity for businesses, individuals, and countries*" (Cisco, 2013).

The Internet of Everything (IoE) expresses a world where billions of objects along with sensors to determine and evaluate their location; connected to public or private networks using all standard and proprietary protocols. Edge computing is changing the way we manage, process, and distribute data from millions of devices worldwide. The tremendous growth of Internet-connected devices in IoE, along with new applications that need simultaneous computing power, continues to drive edge-computing systems. Accelerated networking technologies, such as 5G wireless, artificial intelligence, auto-driving cars, allow video processing and analytics, and robotics to accelerate the design or hold up of real-time applications to edge and computing systems, to name a few. Due to the growth of IoE-generated data, the initial aim of edge computing was to address bandwidth costs for long-distance travel data, with the emergence of real-time applications advancing the need for processing technology [2].

Edge computing defined by Gartner as "data processing as part of a distributed computing topology located near the edge – where things and people produce or receive that information" [2]. In its early stages, edge computing did not depend on any central location thousands of miles away but rather brings computing and data storage closer to assembling devices. This is done in such a way that data, especially real-time data, does not suffer from delayed issues that can affect the performance of an application. Besides, companies can preserve money by completing processing locally, reducing the amount of processing required either centrally or in cloud-based locations.

Edge computing was created because of the significant extension of IoE devices, which are wirelessly connected through the Internet to fetch data from the cloud or return data to the cloud. Many IoE devices produce large amounts of data during their activities. Also, edge computing may provide new functionality that was not previously available. For example, an organization can use an edge computer to analyze their data on the edge, which makes it possible in real-time. Typically, the major benefits of edge computing are low latency, low bandwidth usage and low associated costs, and low use of resources in the server.

A drawback of edge computing is that it can increase attack vectors. As the devices are connected to each other wirelessly, authentication is the key factor in communication for such cloud-edge infrastructure. In [3], it has been reported that edge computing has increased dramatically in recent years which is targeting aging. Among all the security attacks, the most remarkable attacks occurring in the practical world is the Mirai virus [3]. Mirai virus captures more than 65000 IoE

devices within the first 20 h after its deliverance in August 2016. A few days later, these compromised devices shut down over 178 000 domains and turned to Botnet to run Distribution Denial Services (DDoS) attacks against edge servers. Within a short period, a variety of Mirai, such as the IORPitter and Hazim, were captured, and they are believed to infect 3 million IoE devices in 2017 [4]. Since the discovery of the first Mirai botnet in 2016, the IoE botnet attacks were disclosed to have caused more than $100 million in damage as of September 2018 [4]. It is noted that these numbers only indicate attacks and property damage that were officially pointed out and enlisted, but the total amount of unauthorized attacks/damage may be very high.

The arrangement of this chapter is maintained as follows. Section 2 discusses some mobile edge computing-based IoE based applications and their security. Section 3 demonstrates the different architecture used in MEC. Section 4 discusses possible attacks on communication in MEC and list out the cryptographic solution. Section 5 illustrates a secure communication protocol in an edge-cloud environment that can be applied to the healthcare system. The brief discussion on some other related existing secure communication protocols is given in Sect. 6. In Sect. 7, the security challenges of MEC discuss. At the end, the conclusion is given.

2 Applications and Security

Edge computing applications, data, and services can be used to push the logical end of a network away from central computing. This enables additional data sources to be in the age of analysis and data. Edge encompasses a wide range of computing technologies, such as remote sensing systems, filling traditional data stocks, and augmented reality.

It is easy to search clarifications for what edge computing is and how it works. Most companies need to know how it can affect their business. Internet of Everything (IoE) gadgets is now available on the market in large numbers. Thus, agencies require seeing how new evolutions in edge computing practice can be made more convenient for them. Figure 1 shows some mobile edge computing-based IoE applications.

Here, some of the most novel applications in the mobile edge computing are addressed:

Manufacturing: By putting data storage and registering in industrial equipment, manufacturers can collect data that will consider better perception and adequacy of redundancy, while reducing costs and requirements while maintaining better stability and remunerative time. Common manufacturing frameworks guided by consistent data diversity and will help more companies make changes to the order created to meet prospects for operational requirements.

Smart Cities: The edge computing architecture responds to real-time changes on behalf of devices that control utilities and other public administrations. With the increasing number of autonomous e-devices and the ever-increasing IoE, smart

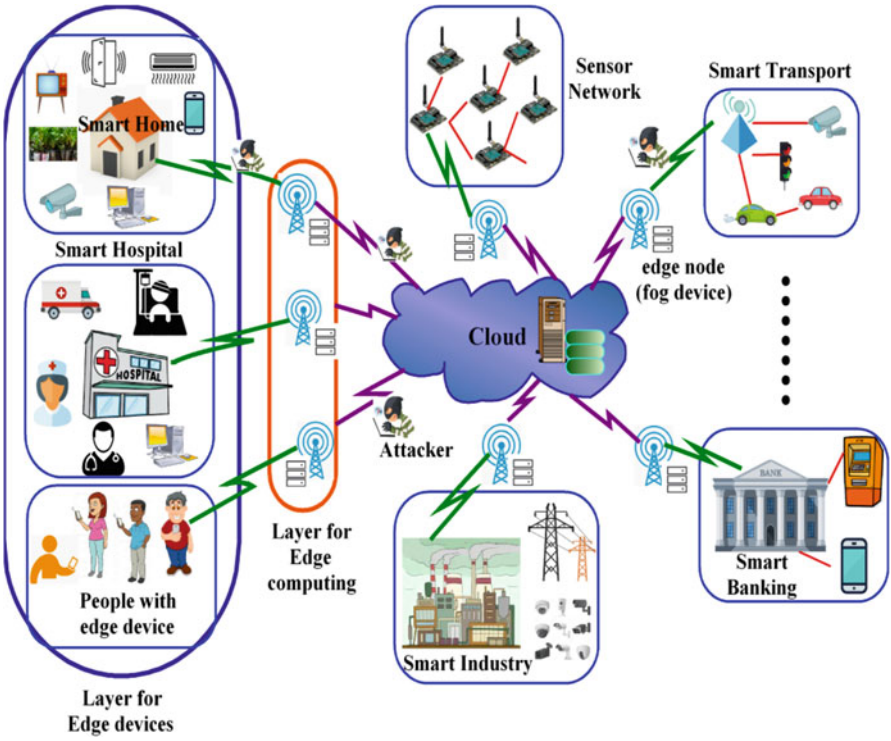

Fig. 1 The layered architecture of mobile edge computing-based IoE

cities [5] can change how people survive and benefit from urban environments. Since all end computing applications rely on gadgets to collect data to perform basic processing tasks, they will have the ability to react rapidly with the changing circumstances occurring in the future city.

Healthcare: IoE gadgets are perfect for providing a vast array of patient-borne health information (PGHD) [6, 7], allowing healthcare providers to access essential data about their patients rather than interface with intermediate and fragmented databases. Treatment devices can be similarly determined to determine and collect information about the entire treatment. Regulatory requirements for the exchange and risk of medical data make it challenging to implement any edge solution.

Augmented Reality: Wearable augmented reality (AR) gadgets such as smart eyeglasses and headsets are sometimes used to create this effect; however, most customers have run into AR via their mobile displays. Anyone who has made a noise like Pokémon Go or used a channel on Snapchat or Instagram has used AR. The innovation behind AR is that devices expect to process visual information and are incorporated into pre-rendered visual elements. Without an edge computing design, this visual information will be distributed back to a

centralized cloud server where digital components can be added before being sent back to the gadget. This course of sequence inevitably leads to significant delays.

AI Virtual Assistant: By incorporating edge systematization into the systems, organizations can completely improve performance and reduce inactivity. Instead of sending AI virtual assistants to a focused server and sending data requests, they can locally spread weights between edge data centers playing some processing capabilities. It can be said that the multiplication of localized data servers for both cloud and edge computing has made it easier than ever for the association to be in a position to expand its network and maximize the benefits of its data resources.

Smart Transport: Smart transport [8] is the future of the world transportation system. With comparison to before, today we have more and better transportation options, and we have new ideas to enhance, invest, and consume transportation services. To reduce traffic congestion and improve living standards, the city government aims to promote green, efficient transportation systems. With the help of IoE, cloud and edge computing makes it easier.

Smart Building: Adaptability is crucial in smart building [9] because, it interacts with the systems, people, and exterior elements around them with the help of IoE devices then stores in the cloud. Data are collected through Edge computing devices. Edge devices have learned from past experience and real-time input. It enhances comfort, efficiency, flexibility, and security to facilitate the needs of people and trade between them. Here is the use of Edge Computing.

Smart Industry: The world of industry is turning into a trend that goes by various names including Industry 4.0, Industrial Internet of Things (IoT), and Smart Power Grid [10]. It is a safer, more experimental, more environmentally friendly design of smart industrial factories and functions. With factories accounting for 40% of the world's energy consumption, reducing their energy consumption will play a significant role in bringing the planet on a more sustainable path. Machines are evolving to be aware of the people around them and provide new interfaces such as smart interfaces, augmented reality, touchless interfaces for easy and secure communication. The devices are being integrated inside the factory and with the cloud, enabling optimal planning and flexibility for production and maintenance. Here Edge computing can help in a better way.

Autonomous Vehicles: The choice to stop or not for a pedestrian crossing in front of an autonomous vehicle [11] should be taken instantly. In that case, it is not appropriate to rely on a remote server to handle this decision. However, the vehicles that use edge computing can interconnect more systematically because they first communicate with each other to prevent accidents by sending data on the first trip to a remote server. Edge computing can be used here to overcome the said problem in autonomous vehicles applications.

Surveillance: Security systems can detect possible threats and then can notify users to abnormal activities in real-time. Responding to a threat within seconds, the security monitoring systems can also be benefited by incorporating edge computing mechanism.

Retail Advertising: Targeted advertisements for retailers and data fields are based on key parameters such as the population data set on the device. In this case, the edge computing can help to preserve user privacy. It can keep the source instead of encrypting the data and not sending secure information to the cloud.

Smart Speakers: Speakers with smart sensors can gain the potential to interpret voice commands locally. Adjust the thermostat settings on or off or even if the Internet connection fails. Edge technology is rapidly used in such an application.

Video Conferencing: Delay in audio, poor video quality, a slow link to the icy screen-cloud video conferencing can produce a lot of frustration. By keeping the server-side of the video conferencing software to the contributors, quality problems can be minimized. While edge computing is in many cases a wise alternative to cloud computing, there is always room for enhancement. But, according to [12], the existing IoE security protocols need to be enhanced so that it can be used in practical scenarios.

In the above-mentioned applications, the security in communication for edge technology is a primary concern. Besides, a possible solution to further secure IoE-generated data is an IoE management component known as a security agent. This new piece will use routers and other near-edge boxes that cannot accommodate IoE devices. As well as being more secure, it will also make it easier to manage the key. The security agent box can operate a large number of sensors that are difficult to use. The researchers said that IoE applications would fail if the required verification was not done quickly.

3 Architecture for MEC

In this section, the layered architecture of edge computing will be described. According to the communication, architecture can be divided into three layers, (a) layer for edge devices, (b) layer for computation, and (c) cloud layer. Figure 1 shows the layered architecture of mobile edge computing-based IoE applications.

a. ***Layer for edge devices***: In this layer, edge devices like mobile, sensors, and laptop are connected to each other. These devices may use short communication interfaces like Bluetooth, ZigBee depending upon application and availability of the connection. For this purpose, a personal area network (PAN) can be used. The edge devices transmit data to the local edge server for processing (see Fig. 1).
b. ***Layer for computation***: After collecting data from edge devices, in this layer, the edge server like fog server processes the data. The edge server periodically collects data from the edge devices. Sometimes, depending on the application, if any person wants to access fresh and real-time data, then after proper verification, he/she can get data from this layer. However, this is the local data as the edge server is connected to the edge devices locally. After processing data, the edge servers send the data to the cloud so that users can access data globally (see Fig. 1). For this purpose, the edge server uses the Internet for communication.

c. *Cloud layer*: After getting data from each edge server, in this layer, the cloud server stores the data in a secure way so that users can get data whenever they want via the Internet. However, the data in this layer may not be fresh because the edge servers do not send data periodically to the cloud.

All the communications are done in public channels like Bluetooth and the Internet; therefore, an attacker alters the messages and hampers the communications (see Fig. 1). Even the adversary may try to extract the secret information of edge devices, servers. Not only that, but the attacker may also try to access data from the cloud and edge server. Thus, the protection of unauthorized access is a key term in such critical communications. However, later, this chapter will discuss the security challenges and issues for mobile edge computing-based IoE applications.

3.1 Network Model

To design a secure communication protocol based on edge-cloud architecture, the network model plays an important role by which the flow of data and authentication can be achieved. For this purpose, researchers generally use two types of network model (a) single server environment, and (b) multi-server environment. The details are described as follows:

a. ***Single server environment***: Edge devices are connected to the local edge server and each local edge server connected to a global cloud server. In this regard, the global cloud server controls all the communications and edge servers and edge devices. The global server serves all the requests and services to the users globally. Figure 2a shows the single server environment.

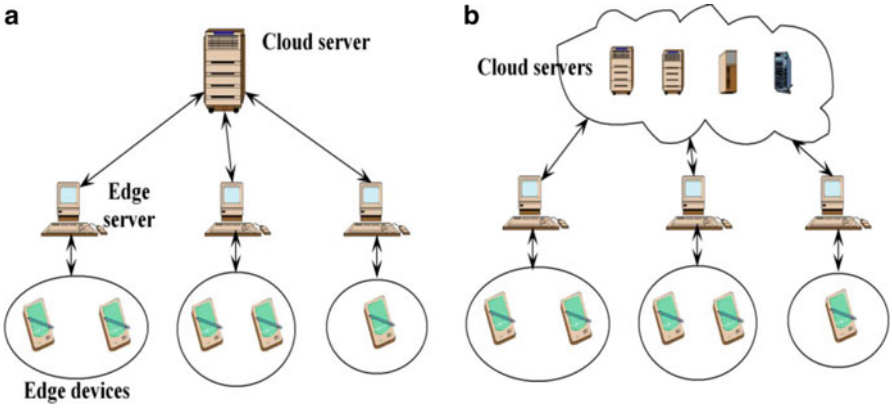

Fig. 2 Network model: (**a**) server environment, and (**b**) multi-server environment

b. ***Multi-server environment***: Edge devices are connected to the local edge server and each local edge server connected to the corresponding global cloud server depending on the service provider. In this regard, the cloud servers distribute their tasks depending upon the availability of the resources. Figure 2b shows the single server environment.
c. ***Hybrid***: In such an environment, edge servers and cloud servers are decentralized. One registration center (maybe part of the governing body) controls the total networks. The networks are divided into several sub-networks as a company based and provides several services.

4 Possible Attacks and Cryptographic Solution

This section discusses the possible attacks on Mobile Edge Computing (MEC) during communication. Then a brief cryptographic solution is given on that direction. The possible attacks during communication in MEC listed below:

1. *DDoS attacks*: The goal of a DDOS attack is to connect all available resources and bandwidth to the target, and prevents malicious users from using the compromised system. The attacker constantly sends a large number of packets to the target (also known as 'flooding'), ensuring that all of the target's resources are exhausted to handle the corrupted packet, and therefore the actual requests cannot be processed. Such attacks are more important on edge computing paradigms because they are comparatively less powerful (compared to cloud servers), and therefore cannot run robust defenses.
2. *Malware attacks*: The inability to install a complete firewall on resource-limited edge devices makes them vulnerable to malware injection attacks, allowing an attacker to secretly install malicious programs on a target system.
3. *Authorization attacks*: Authentication processes in Edge computing systems can also be vulnerable to attacks. These types of attacks can be categorized into four different categories: dictionary attacks, attacks targeting vulnerabilities in authentication systems, attacks that exploit sensitivity to authorization protocols, and extra-privileged attacks.
4. *Side channel attacks*: Common examples of such attacks include capturing contact signals (such as packets or wave signals) to get user's personal data, monitoring the power consumption of edge devices to disclose usage patterns, and targeting end devices on file system and sensors like microphones, and cameras.

Cryptographic protocols used to protect privacy on secret information as well as to eliminate the possible attacks. The protocols used in MEC is categorized into (a) public key based, (b) secret key based, (c) only one-way hash function based, and (d) public plus secret key based (see Fig. 3). Depending on applications in MEC, public plus secret key based cryptosystem is used. For an example, an application where, wireless sensor devices are used in communication, in that case public key

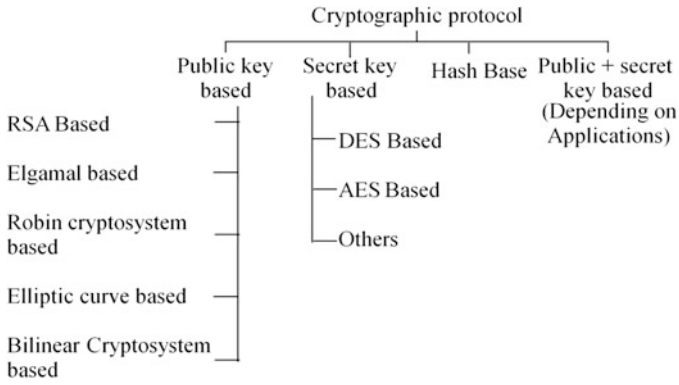

Fig. 3 Different cryptographic protocols used in MEC

cryptography cannot be used due to high computation cost. It results more energy consumption in sensor device during communication among IoE devices and edge server. In such case, secret key based protocol is used and in the higher level (i.e., edge to cloud communication), public key cryptography is used to provide more security during communication. This is because, edge and cloud servers have unlimited power as well can they can do the high computation operations.

In the next section, an Elliptic curve (ECC) based secure protocol [13] for communication in MEC environment has been discussed. This chapter picks ECC because; it can produce same security level with smaller key size. This work refers article [14] to know more about ECC. Moreover, in the protocol [13], sensor to edge server secure communication and vice versa has been done using secret key cryptography to reduce energy consumption of sensor devices. The remaining communication (edge to cloud and vice versa) has been done using ECC.

5 Secure Communication Protocol

This section discusses an edge-cloud based security protocol [13] which is applicable in the healthcare system. The protocol used in [13] is based on the elliptic curve cryptosystem [14].

5.1 Architecture

Before going to discuss the protocol [13] in detail, this section will discuss the architecture of the protocol (see Fig. 4). Sensors (the layer for edge devices) send messages periodically to the local edge server. The edge server forwards the message to the cloud server for authentication. After, correct verification, the cloud

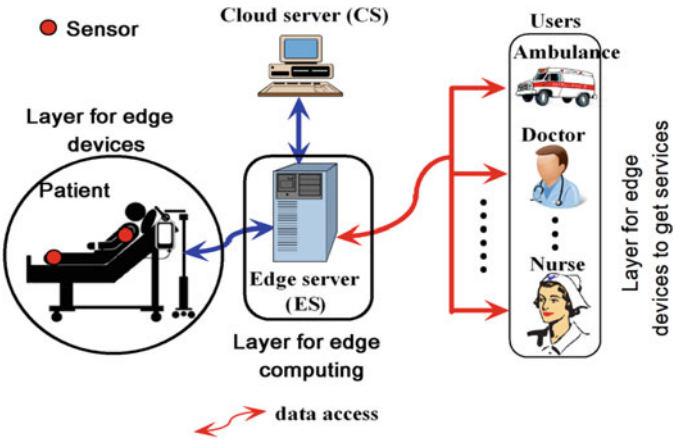

Fig. 4 Network structure of the existing secure protocol [13]

server replies back to the edge server. Upon getting a reply back from the cloud server, the edge server, checks the message and if the message is correct, then it forwards to the sensors. Finally, a secure session will be established between the edge server and the sensor (i.e., patient) for secure data transmission. However, in this protocol, how the other users like, doctors, nurses will get data from edge server is not demonstrated. But, they can get access to data from the edge server after proper authentication procedure.

5.2 Protocol in Details

The protocol [13] has four phases: (a) startup phase, (b) enrollment phase, (c) verification phase, and (d) data transmission phase.

a. ***Startup phase***: A cloud server (CS) picks a long prime number y and makes an elliptic curve on a finite field of order m with a base point X. CS randomly selects a secret key $k \in_R [1, m-1]$ and computes the corresponding public key $P = [k]X$. CS selects three cryptographic hash functions: $hf_1(.) : \{0,1\}^* \rightarrow \{0,1\}^n$ for a fixed n bits, $hf_2(.) : G_y \rightarrow \{0,1\}^{n1}$ for a fixed $n1$ bits and $hf_3(.) : G_y \rightarrow \{0,1\}^n$. Then CS announces $\langle X, m, p, hf_1(.), hf_2(.), hf_3(.) \rangle$ and k has been kept as a secret.
b. ***Enrollment phase***: In this phase, CS supplies the information regarding registration to the edge servers as well as the healthcare sensors.

   Enrollment of edge servers: An edge server ES_i selects its unique identity EID_i and sends it to CS. After getting EID_i, CS selects a random number $a_i \in_R [1, m-1]$ which is a secret key of ES_i and calculates a public key $EPK_i = [a_i]X$. CS then sends $\langle a_i \rangle$ to ES_i through secure channel and announces $\{EID_i, EPK_i\}$ publicly. Upon getting a_i, ES_i stores it securely.

Enrollment of sensors: Before going to place a healthcare sensor S_i on the patient's body, *CS* chooses an unique identity S_ID_i for S_i and calculates its key Key_i as $hf_2(e_i \,||\, S\_ID_i)$, where e_i is a random number chosen by *CS*. *CS* again calculates a pseudo identity PS_ID_i as $hf_2(S\_ID_i \,||\, k)$ for S_i and stores in its database as $Sensor\_DB = \{PS\_ID_i, ENC[S\_ID_i \,||\, Key_i]_k\}$, where $ENC[.]_k$ means encrypted using a secret key k. Then *CS* burns $\langle PS\_ID_i, S\_ID_i, Key_i\rangle$ into the memory of S_i as temper resist.

c. ***Verification phase***: If a healthcare sensor S_i has data to send, it sends a request to send message as $\langle EID_i, PS\_ID_i, V_i, W_i\rangle$ to ES_i after calculating $V_i = ENC[z_i \,\|\, SID_i \,\|\, EID_i]_{Key_i}$ and $W_i = hf_1(z_i \| S\_ID_i \| V_i)$ where, z_i is a random number chosen by S_i.

After receiving $\langle EID_i, PS\_ID_i, V_i, W_i\rangle$, ES_i forwards the message as $\langle EID_i, A_i, C_i, Q_i\rangle$to *CS* through the Internet after calculating $A_i = [l_i]X$, $B_i = [l_i]P$, $C_i = (PS\_ID_i \,||\, V_i \,||\, W_i \,||\, EID_i) \oplus hf_2(B_i)$ and $Q_i = [hf_1(C_i)]X + [a_i]P$, where l_i is a random number chosen by ES_i.

After receiving $\langle EID_i, A_i, C_i, Q_i\rangle$ from ES_i, *CS* calculates $B_i^{\#} = [k]\,A_i$, $\mathrm{PSID_i^{\#}} \,\|\, \mathrm{V_i^{\#}} \,\|\, \mathrm{W_i^{\#}} \,\|\, \mathrm{EID_i^{\#}} = \mathrm{C_i} \oplus \mathrm{hf_2}\left(\mathrm{B_i^{\#}}\right)$ and extracts $S\_ID_i \,||\, Key_i$ from its $Sensor\_DB$ by decrypting $ENC[S\_ID_i \,||\, Key_i]_k$ using its secret key k corresponding to $\mathrm{PSID_i^{\#}}$ if it exists into the database. *CS* then decrypts $\mathrm{V_i^{\#}}$ using Key_i to extract $z_i^{\#} \,\|\, SID_i^{\#} \,\|\, EID_i^{\#\#}$ as $DEC\left[V_i^{\#}\right]_{Key_i}$ and, checks extracted $SID_i^{\#} \stackrel{?}{=} SID_i$ and $EID_i^{\#\#} \stackrel{?}{=} EID_i^{\#} \stackrel{?}{=} EID_i$. For the equality, *CS* calculates $W_i^{\#\#} = hf_1\left(z_i^{\#} \| SID_i \| V_i^{\#}\right)$ and $Q_i^{\#} = \left[hf_1(C_i)\right]X + [k]\,EPK_i$. *CS* then further checks $W_i^{\#\#} \stackrel{?}{=} W_i^{\#}$ and $Q_i^{\#} \stackrel{?}{=} Q_i$. For the equality, *CS* transmits a reply message $\langle CS_1, CS_2, CS_3\rangle$ to ES_i via the Internet after calculating $CS_1 = [u_i]X$, $CS_2 = z_i^{\#} \oplus hf_3\left([u_i]\,EPK_i\right)$ and $CS_3 = [z_i^{\#}]\,X + [k]EPK_i$, where u_i is a random number chosen by *CS*.

After receiving $\langle CS_1, CS_2, CS_3\rangle$, ES_i calculates $z_i^* = CS_2 \oplus hf_3([a_i]CS_1)$ and checks $CS_3 \stackrel{?}{=} \left[z_i^*\right]X + [a_i]\,P$. For the equality, ES_i transmits a clear to transmit message $\langle Y_i, H_i\rangle$to S_i after computing $Session^k = hf_1(z_i^* || \, t_i)$, $Y_i = t_i \oplus z_i^*$and $H_i = hf_1(Session^k \,||\, Y_i)$ where, t_i is a random number chosen by ES_i.

After receiving$\langle Y_i, H_i\rangle$, S_i calculates t_i as $Y_i \oplus z_i$, $Session^k = hf_1(z_i \,||\, t_i)$ and verifies the received $H_i \stackrel{?}{=} hf_1\left(\mathrm{Session^k} \,\|\, Y_i\right)$. For the equality, S_i agrees on the common secret session key $Session^k$ in data transmission phase.

d. ***Data transmission phase***: After agreement on $Session^k$, S_i transmits its sensed data as a cipher $CIPHER\_DATA = ENC[DATA]_{Sessionk}$ to ES_i. After receiving $CIPHER\_DATA$, ES_i de-cipher it by using the same session key $Session^k$ as $DATA = DEC\ [CIPHER\_DATA]_{Sessionk}$ and analyzes the data. ES_i stores the data as cipher form using its secret key a_i corresponding to PS_ID_i as $\{PS\_ID_i, ENC[DATA]_{ai}\}$ for future reference to the users like doctors and nurses.

A flow chart of verification and data transmission phases of the proposed scheme in [13] is given in Fig. 5.

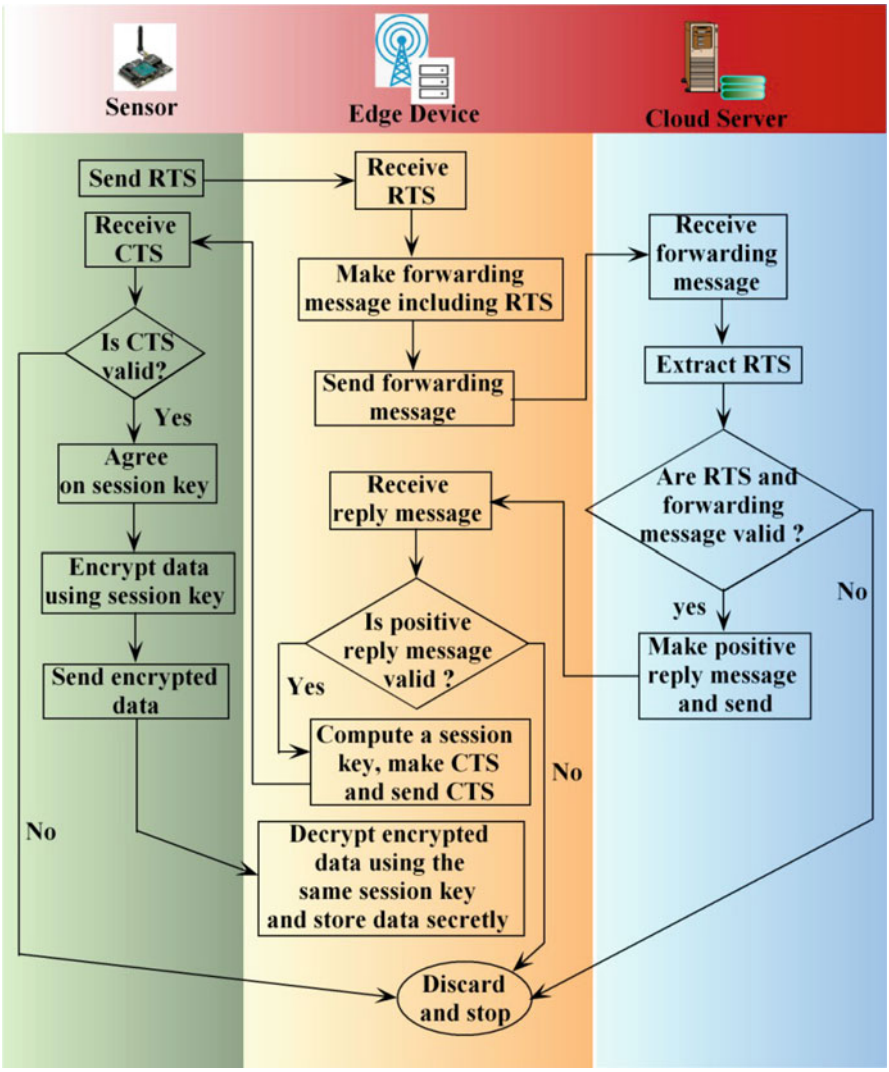

Fig. 5 Flow chart for authentication and data transmission phases [13]

6 Other Security Protocols: A Comparison

This section demonstrates the existing security protocols for a cloud-edge environment. This section also compares the related existing security schemes. By decreasing end-to-end delay and enhanced position perception with mobile facilities, *Mobile Edge Computing* (MEC) furnishes smooth services. Since MEC progressed from cloud computing, it has subsequently inherited many security and

privacy issues. Besides, decentralized testing and diversified installation environments on MEC platforms exacerbate the problem; the research causes great concern for the community. So, in 2019, Kaur *et al*. [15] have proposed an efficient and lightweight mutual verification protocol for the environment of MEC; based on cryptography based on elliptic curves (ECC), cryptographic hash function and work with content. The designed protocol also presents the advantages of counteracting individual computational Diffie-Helman, logarithm problems, random numbers and time-stamps, multi-attack-resistant attacks, replay attacks, and man-in-the-middle attacks. The work in [15] claims that it is suitable for acquiring resource hindrance MEC environments. Omala et al. [16], Cheng et al. [17] and He et al. [18] introduced their security protocols that can enable a patient to securely transmit their data directly to application servers (mainly cloud servers) using their mobile application. However, such a situation is not always possible, as no patient may be able to manage his mobile application in his critical situation. So, an automated system is needed to handle this problem, where sensors can send their data securely from time to time. Recently, Maitra and Roy [19] suggested a secure communication scheme for patient monitoring system, known as *SecPMS*. In the approach [19], the end users such as doctors and nurses get patients' information securely from a local server (i.e., edge server) after performing authentication procedure.

On the other hand, IP-based communication is a serious security threat for MEC. Thus, secure information sharing between diverse communication agents has become an important concern in smart grid environments. In particular, to enable secure communication among smart meters and utilities, managing the key before authentication is the most important task. Mehmood *et al*. [20] proposed an identity-based signature to represent an anonymous key agreement protocol for smart grid infrastructure. The protocol [20] enables smart meters to be interconnected to anonymous utility controls for the services they provide. Smart meters recognize this purpose with a secret key in the absence of reliable authority, where the trusted officers are only intricate in the enrollment phase.

On the Internet of Thing (IoT) systems, large amounts of data are accumulated at any given time, which can capture human privacy, mostly when the system is used in medical or everyday environments. Privacy protection is an important issue and high privacy claims usually demand a weak identification. The earlier researches have stated that well built security demands strong identification, particularly in authentication processes. Therefore, defining a better business between privacy and security remains a challenging issue. Wang [21] introduced a security, accountability, privacy-protection, efficiency, and dynamic removal necessity for weakly identified IoT end-of-device authentication frameworks. For this purpose, the author in [21] used Shamir's secret sharing project [22] for a basic installation and distribution project for secure communication between the end device and the end device. A small-group signature scheme [22] has then been used to make a privacy-preserving and accountable verification protocol for weakly identified IoT end-devices.

Not only secure communication but secure database access also important in MEC. In this regard, Pang and Tan [23] have proposed an edge that creates a

Table 1 Existing security protocols for MEC: A comparison

Purpose	Protocols	Computational cost	Latency	Security	Network model
Secure Communication + Authentication	[15]	high	high	medium	single server
	[16]	high	medium	high	single server
	[20]	high	high	medium	multi server
	[21]	medium	medium	medium	single server
Authentication + Secure Data Store	[13]	medium	low	medium	single server
Secure database access	[23]	-	-	-	hybrid

validation object (VO) to verify the integrity of the result of each query generated by an edge server – the results of which do not tamper with the values; even though any attacker enthusiasts add fake tuples. The primary advantage of the proposed system [23] is that it is unique compared to the size of the VO database and those relevant activities can still be performed by the edge server. The said mechanism turns down the communication load and processing complexity at the client end.

Table 1 gives a summary of the aforementioned existing secure protocols, where, latency is considered with respect to the number of bits transmitted.

7 Issues and Challenges to Design Security Protocols

This section discusses the challenges to design a security protocol for the edge-cloud environment.

IoE Vulnerabilities at the Edge: Edge computing fixes a variety of IoE networking traffic issues; however, it often introduces new weaknesses that contribute to an overall wider attack surface, that is, the total number of access points for a network that can be used by an adversary. Networks become more vulnerable at ends and edges due to the condition of existing platforms. Some attacks may occur as end-users generally don't change their default passwords. This creates a path for malicious people to have access to the user's end devices, as they are now exposed to attack.

Internet resources that are not secure can be found easily and are accessible. In a 2017 "botnet barrage" bots were introduced to check for devices running default passwords at university campus. In the year 2013, an application was released that could scan for unsecured IoE devices around the world. Around 5,000 IoT devices have been hacked by 5,000 individual systems because these devices had default or weak passwords.

The above attacks have been carried out due to the weakness present at the end points, nonetheless edge computing complicates things by exposing new attack surfaces. IoE devices that link to the public Internet violate protection protocols at the edge of the network. This is partially attributed to the existing state of edge computing in which full-stack systems like sensors, applications and protected components are not common. Many of the approaches used to protect IoT networks at the edge can be ineffective. LPWAN protocols can become unstable if encryption keys are stolen. VPNs are vulnerable to man-in-the-medium attacks.

Physical Tampering: Edge computing being distributive in nature often leads to opening up of new, unexpected frontier of physical risks. Although servers and computers that drive conventional networks are typically located in large, sometimes extremely protected warehouses, the very tiny data centers that render edge computing such a massive leap forward may often be a security nightmare.

Instead of keeping in data centers, such micro-centers are mostly installed in an area that, as we think about IoE edge, may be a corporate office, a garden, and everything in between. An intruder who physically tampers with an edge system may bring down a network, or even damage one of its operators. Securing these systems is also far from straightforward – as they need to be protected against physical threats, it is often a tradeoff between reliability, expense, and ease of updating and maintaining edge data centers. Device manufacturers need to be aware of the threats to ensure the systems can be conveniently monitored to trigger remote and local alerts at any indication of interference.

Lacking Reflection of Secure Design: The primary aim of edge computing is to furnish a more powerful and lightweight computing environment for evolving technologies such as IoE and smart cities [3]. While building designs, device designers prefer to rely more on efficiency than on the security part, when building the application-specific edge computing architecture. Such a lackadaisical attitude towards security explicitly uncovers the edge computing infrastructures to larger attack sides.

Non-migratability of Security Frameworks: The security framework for general-purpose computer systems have been widely researched for a long time and are known to be capable of offering good security assurances in the defense against numerous threats [3]. Nonetheless, such security architectures cannot be explicitly transferred to edge computing platforms due to a variety of irresolvable differences, such as competing processing resources, diverse OSs and applications, specific network architectures, and incompatible protocols. Also, security frameworks outlined for an edge computing application may not be directly transferred to another scenario such as diversity of edge devices as well as diversity in intelligent transmission protocols.

Coarse-Grained and Fragmented Access Control: Current access management frameworks for edge computing are inconsistent and coarse-grained [3]. They are fragmented since various edge computing contexts can follow specific access management models that may be configured in a fully distinct way for segregating, granting, and obtaining permissions. This condition hinders the

creation of a coherent and functional access control platform for different edge computing systems. Recent access control mechanisms for edge computing are also coarse-grained because, with compare to coarse-grained, permissions in fine-grained are largely complex and underexplored.

8 Conclusion and Future Direction

Based on the basic computing reasons, the status queues, and the magnificent challenges of achieving edge computing systems, this chapter can conclude that research on the security domain in edge computing technology is far from the delighted result. Future research focuses should lie in the grand challenges and should overcome the existing weaknesses. For such edge-based applications, more robust defense solutions are needed to reduce personal attacks, especially preventive measures; on the other hand, new architectures are needed that can integrate the entire system and can incorporate security measures to protect the secure information from an outsider when online communication will be done. Most significantly, the philosophy of safety by design should be widely adopted and always returned. Inspired from the article [3], below, this chapter outlines a basic concept that seeks to secure edge computing systems with integrated structure and current future directions along this line of research. The structure consists of three layers: (a) a fine-grained outer access control layer, (a) a medium-security function layer, and (c) an internal hardware-isolated OS layer.

The outer layer focuses on fine granular access control, which acts as a gate to prevent intruders from entering. If properly designed and strictly implemented, such fine access control systems can potentially reduce protocol-level design errors, implementation-level errors, and attacks generated by weak access control. It can carry flood-based DDoS, controllable side channels, malware injection attacks, and attacks in the verification process.

There are plans to implement medium level full security measures. This chapter proposes the adoption of software-defined networking (SDN) and network function virtualization (NFV) at the edge server level, where SDN is adopted to filter out malicious traffic on a per-packet basis. In contrast, NFV adopts more advanced algorithms such as intensive learning to detect malicious behaviors in autonomous and self-developed methods. SDN and NFV-enabled edge servers can prevent packet-based attacks such as DDoS, attacks arising from connected data (requiring learning-based detection), and poor access control (which can lead to attacks such as malware injection).

The inner layer notices unnecessary code-level vulnerabilities. Moreover, the IT and telecommunications worlds have experienced real ideological changes over the years. The concept of mobile edge computing has recently been published, applying fog computing (edge-on-cloud) to mobile network domains. However, edge technology will have a real impact on the way new services are installed as they will benefit from a combination of SDN plus NFV. Either way, IoT, which is

highly connected with mobile networks, will benefit by expanding the concept of mobile edge agent computers to other areas such as VANET and WSN.

This chapter has first described some edge technology-based applications that are recently under consideration in the research domain. Then the system architectures have been discussed concerning the design of edge-based applications. Then one secure communication protocol has been highlighted for a cloud-edge based health-care system. After that brief overview of recent secure communication protocols for edge-based applications has been compared. The designing issues and challenges have been enlisted then. Lastly, future direction and probable solutions have been discussed in this section. After enlightening all the things, and then also this chapter can say that the developing research in edge computing security is still under construction and there have so many scopes to re-design the security protocols. Inspired by emerging applications and advances in modern cryptography, innovative design, and applications to secure edge computing systems will be enriched in the distant future.

References

1. Bradley, J., Barbier, J., Handler, D.: Embracing the Internet of Everything To Capture Your Share of $14.4 Trillion. published by cisco (2013). https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoE_Economy.pdf
2. Cisco IoT. http://www.cisco.com/web/solutions/trends/iot/overview.html
3. Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., Lv, W.: Edge Computing Security: State of the Art and Challenges. in Proceedings of the IEEE. vol. 107. no. 8 (2019) 1608–1631.
4. Antonakakis et al.: Understanding the Mirai botnet. in Proc. of 26th USENIX Secur. Symp. Vancouver, BC, Canada: USENIX Association (2017) 1093–1110.
5. Dutta, J., Roy, S., C. Chowdhury, C.: Unified framework for IoT and smartphone based different smart city related applications. Microsyst Technol. Vol. 25. (2019) 83–96.
6. Maitra, T., Giri, D.,: An Efficient Biometric and Password-Based Remote User Authentication using Smart Card for Telecare Medical Information Systems in Multi-Server Environment. Journal of Medical Systems. Vol. 38. no. 12, article no. 142, (2014) 1–19.
7. Giri, D., Maitra, T., Amin, R., Srivastava, P. D.: An Efficient and Robust RSA-Based Remote User Authentication for Telecare Medical Information Systems. Journal of Medical Systems. Vol. 39. article no. 145, 2015.
8. Hu, L., Ni, Q.: IoT-Driven Automated Object Detection Algorithm for Urban Surveillance Systems in Smart Cities. IEEE Internet of Things Journal. Vol. 5, no. 2, (2018) 747–754.
9. Dutta, J., Wang, Y., Maitra, T., Islam, SK. H., Rawal, B. S., Giri, D.: ES3B: Enhanced Security System for Smart Building using IoT. in Proc. of The 3rd IEEE International Conference on Smart Cloud (SmartCloud 2018). New York, USA, (2018) 158–165.
10. Ou, Q., Zhen, Y., Li, X., Zhang, Y., Zeng, L.: Application of Internet of Things in Smart Grid Power Transmission. in the proc. of 2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing. Vancouver, BC, (2012) 96–100.
11. Kong, L., Khan, M. K., Wu, F., Chen, G., Zeng, P.: Millimeter-Wave Wireless Communications for IoT-Cloud Supported Autonomous Vehicles: Overview, Design, and Challenges. IEEE Communications Magazine. Vol. 55. no. 1. (2017) 62–68.
12. Hsu, R., Lee, J., Quek, T. Q. S., Chen, J.: Reconfigurable Security: Edge-Computing-Based Framework for IoT, IEEE Network. Vol. 32. no. 5. (2018) 92–99.

13. Giri, D., Obaidat, M. S., Maitra, T.: SecHealth: An Efficient Fog based Sender Initiated Secure Data Transmission of Healthcare Sensors for e-Medical System. in Proc. of IEEE GLOBECOM 2017. Singapore, (2017) 1–6.
14. Maitra, T., Obaidat, M. S., Islam, SK, H., Giri, D., Amin, R.: Security analysis and design of an efficient ecc-based two-factor password authentication scheme. Security and Communication Networks. Vol. 9. no. 17. (2016) 4166–4181.
15. Kaur, K., Garg, S., Kaddoum, G., Guizani, M., Jayakody, D. N. K.: A Lightweight and Privacy-Preserving Authentication Protocol for Mobile Edge Computing. in Proc. of IEEE GLOBECOM 2019. Waikoloa, HI, USA, (2019) 1–6.
16. Omala, A.A, Kibiwott, K.P., Li, F.: An efficient remote authentication scheme for wireless body area network. Journal of Medical Systems. Vol. 41. no. 2. (2016) 1–9.
17. Cheng, Q., Zhang, X., and Ma, J.: Icasme: An improved cloud-based authentication scheme for medical environment. Journal of Medical Systems, Vol. 41. no. 3 (2017) 1–14.
18. He, D., Zeadally, S., Kumar, N., Lee, J.H.: Anonymous authentication for wireless body area networks with provable security. IEEE Systems Journal, Vol. 11. no. 4, (2016) 1–12.
19. Maitra, T., Roy, S.: Secpms: An efficient and secure communication protocol for continuous patient monitoring system using body sensors. in proc. of 9th International Conference on Communication Systems and Networks (COMSNETS 2017), Bangalore, India, (2017) 322–329.
20. Mahmood, K., Li, X., Chaudhry, S.A., Naqvi, H., Kumari, S., Sangaiah, A.K., Rodrigues, J.J.P.C.: Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure. Future Generation Computer Systems. Vol. 88. (2018) 491–500.
21. Wang, Z.: A privacy-preserving and accountable authentication protocol for IoT end-devices with weaker identity. Future Generation Computer Systems, Vol. 82. (2018) 342–348.
22. Shamir, A.: How to Share a Secret. Com. ACM, Vol. 22. no. 11. (1979) 612–613.
23. Pang, H.H., and K. Tan, K.: Authenticating query results in edge computing. in proc. of 20th International Conference on Data Engineering. Boston, MA, USA, (2004) 560–571.