

Internet of Things

Chintan Bhatt  
Yulei Wu  
Saad Harous  
Massimo Villari *Editors*

# Security Issues in Fog Computing from 5G to 6G

Architectures, Applications and  
Solutions

 Springer

# **Internet of Things**

Technology, Communications and Computing

## **Series Editors**

Giancarlo Fortino, Rende (CS), Italy

Antonio Liotta, Edinburgh Napier University, School of Computing, Edinburgh, UK

The series *Internet of Things - Technologies, Communications and Computing* publishes new developments and advances in the various areas of the different facets of the Internet of Things.

The intent is to cover technology (smart devices, wireless sensors, systems), communications (networks and protocols) and computing (theory, middleware and applications) of the Internet of Things, as embedded in the fields of engineering, computer science, life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in the Internet of Things research and development area, spanning the areas of wireless sensor networks, autonomic networking, network protocol, agent-based computing, artificial intelligence, self organizing systems, multi-sensor data fusion, smart objects, and hybrid intelligent systems.

\*\* Indexing: *Internet of Things* is covered by Scopus and Ei-Compendex \*\*

Chintan Bhatt • Yulei Wu • Saad Harous •  
Massimo Villari  
Editors


# Security Issues in Fog Computing from 5G to 6G

Architectures, Applications and Solutions

 Springer

*Editors*

Chintan Bhatt  
Department of Computer Science and  
Engineering, School of Technology  
Pandit Deendayal Energy University  
Gandhinagar, Gujarat, India

Yulei Wu   
Department of Computer Science, College  
of Engineering, Mathematics and Physical  
Sciences  
University of Exeter  
Exeter, UK

Saad Harous  
Computer Sciences Department  
University of Sharjah  
Sharjah, United Arab Emirates

Massimo Villari   
Dip. di Ingegneria Civile  
Universita' di Messina  
Messina, Italy

ISSN 2199-1073  
Internet of Things

ISSN 2199-1081 (electronic)

ISBN 978-3-031-08253-5

ISBN 978-3-031-08254-2 (eBook)

<https://doi.org/10.1007/978-3-031-08254-2>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Foreword

Computing architectures and services experienced a great revolution in the last two decades, mostly based on the innovation brought forward by more and more sophisticated virtualization technologies and by the consequent evolution of the cloud computing paradigm. Cloud computing, together with the evolution of the terminals (smartphones, tablets, wearable devices, etc.), changed the way we use digital applications that now bring to our hands complex services and applications whose computation is mostly offloaded to some remote cloud infrastructure.

Nonetheless, this approach has limits when it comes to performance issues such as low latency because of the inherent time needed to reach data centers that may be very far away. The most obvious solution is to bring the computation closer to the end user. This is the motivation behind the high interest in fog and edge computing of the last years. Executing the computation that will produce the results relevant to a service as close as possible to the consumer may appear quite obvious to reduce latency and improve responsiveness of applications, but it is also very challenging from the technical point of view.

There is no one-size-fits-all answer to this problem, the technical solutions are many and are typically dependent on the specific application domain. Moreover, this approach opens the floor to a whole set of security challenges, related to the inherent distribution of computation and of information sharing. Therefore, dealing with fog computing solutions requires a careful investigation of the related cybersecurity challenges and possible countermeasures.

This book offers to the reader a series of specific chapters that aim at providing answers to these problems. It starts by giving an overview of the challenges and most studied solutions for fog computing. It then focuses on the network infrastructure, mobile networks considering 5G currently under development or the future 6G, as well as high-performance optical applications. Supporting fog computing scenarios is challenging for the network, and the chapters on these topics analyze these problems and provide possible solutions. Finally, the book provides insights into specific vertical domains, from medical applications to sensing and Internet of Things applications. In all these scenarios, the problem of trust and information

reliability is of paramount importance, and the focus of several chapters of the book is on the possible exploitation of the blockchain technology.

I believe that *Security Issues in Fog Computing from 5G to 6G* will be useful to readers who are starting to approach this complex technical topic, since it puts together many different perspectives, application examples, and specific solutions. At the same time, it will be a useful reference for the more experienced researcher who aims at going deeper into a specific vertical application of fog computing and/or blockchain, or who looks for possible open questions and/or future research topics to be explored.

Alma Mater Studiorum – Università di Bologna,  
Department of Computer Science and Engineering,  
Cesena, Italy

Franco Callegati

# Preface

The fog/edge computing paradigm has been widely adopted for improving the agility of service deployments, allowing for the use of opportunistic and cheap computing resources that can be used to take advantage of network latency and bandwidth diversities among such resources. There are a lot of challenges when it comes to using fog/edge resources, and it is important to revisit operating systems, virtualization and container technologies, and middleware techniques for fabric management in order to address these challenges. We need new ways of programming and storing data in order to create innovative applications that can take advantage of massive distributed and data-driven fog/edge systems. The integration of fog/edge computing and 5G, as well as machine and/or deep learning, will bring new opportunities and challenges for many emerging applications and domains, such as autonomous vehicles and intelligent health. It is of paramount importance to address security, privacy, and trust issues in fog/edge computing, while managing the resources and context of mobile, transient, and hardware-constrained resources.

This book entails ten chapters, including the following studies.

In Chap. 1, Seema and Shailesh aimed to systematically and statistically classify and analyze various well-known security challenges and attacks for the fog enabled Internet of Things (IoT) and Industrial IoT (IIoT) environment. In addition, the authors have considered the “trust” component in this study and examined various security issues.

In Chap. 2, Jinarajadasa et al. highlighted how evolutionary methods available in computational intelligence can be applied to overcome the issues of mobile ad hoc networks in terms of security and reliable communications.

In Chap. 3, Anusha et al. presented a detailed study of this fusion of fog computing with the blockchain technology for achieving the goal of increased security.

Alessandro, in Chap. 4, investigated the challenges and possibilities of wireless communications in fog computing. In particular, the authors examined the advantages of adopting physical layer security (PLS) techniques.

In Chap. 5, Badidi and Sabir proposed the blockchain-based architecture, which serves as a platform for secure data storage and data sharing in a smart city.



Chapter 6 revealed the benefits of integrating modern technologies (fog computing, blockchain, 6G, and IoT) to solve the problem of micropayment systems. Jamal et al. highlighted the various relationships among these technologies and surveyed the most relevant work to analyze how these disruptive technologies can improve the micropayment system's functionality.

In Chap. 7, Kakulapati et al. showcased a solution using one of the most prominent characteristics, that is, traceability, using which we can back-trace the route taken in the supply chain for the targeted drug on which the medical prescription is made.

The main objective of Chap. 8 was to provide a digital identity and anonymous access authentication mechanism based on blockchain, because of the data isolation problem in the network. The authors proposed a trusted multi-domain cooperation mechanism based on blockchain.

Chapter 9 introduced the fog computing architecture and critical features for IoT networks. Harbi and the co-authors presented typical applications of the fog computing paradigm in the context of IoT and provided a classification of these applications.

Chapter 10 is all about concluding remarks, written by Sidath.

This book is designed for researchers, engineers, and developers working in the fields of fog computing with emerging technologies like 5G and blockchain. Practitioners who conduct teaching and cutting-edge research in secure IoT and fog environments will be benefited from this book.

Special thanks to all contributors, respected referees, and our publisher, Springer.

Gandhinagar, Gujarat, India  
Exeter, UK  
Sharjah, UAE  
Messina, Italy

Chintan Bhatt  
Yulei Wu  
Saad Harous  
Massimo Villari

# Contents

<b>1</b>	<b>A Systematic Survey on Security Challenges for Fog-Enabled Internet of Things (IoT) and Industrial Internet of Things (IIoT) ....</b>	<b>1</b>
	Seema B. Joshi and Shaileshkumar D. Panchal	
<b>2</b>	<b>Evolutionary Algorithms for Enhancing Mobile Ad Hoc Network Security .....</b>	<b>15</b>
	G. M. Jinarajadasa and S. R. Liyanage	
<b>3</b>	<b>Blockchain-Based Fog Computing.....</b>	<b>31</b>
	Anusha Vangala and Ashok Kumar Das	
<b>4</b>	<b>Physical Layer Security Challenges and Solutions for Beyond 5G Fog Computing Networks.....</b>	<b>59</b>
	Alessandro Brighente, Mauro Conti, and Foroogh Mohammadnia	
<b>5</b>	<b>Blockchain for Secure Data Sharing in Fog-Based Smart City Systems.....</b>	<b>79</b>
	Elarbi Badidi and Essaid Sabir	
<b>6</b>	<b>Integrating Blockchain with Fog and Edge Computing for Micropayment Systems .....</b>	<b>93</b>
	Jamal Al-Karaki, Deepa Pavithran, and Amjad Gawanmeh	
<b>7</b>	<b>Medical Prescription Traceability Using Blockchain-Based Decentralized Application .....</b>	<b>113</b>
	V. Kakulapati and Parimi Shiva Kalyan	
<b>8</b>	<b>Optical and Wireless Convergence Network Based on Blockchain ...</b>	<b>131</b>
	Hui Yang	
<b>9</b>	<b>Fog Computing Security and Privacy for Internet of Things (IoT) and Industrial Internet of Things (IIoT) Applications: State of the Art.....</b>	<b>145</b>
	Yasmine Harbi, Zibouda Aliouat, and Saad Harous	

**10 Concluding Remarks: Current Challenges and Future Directions ... 159**  
S. R. Liyanage

**Index ..... 163**

## About the Authors

**Chintan Bhatt** is currently working as an Assistant Professor in Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University (PDEU). He is a member of IEEE, EAI, ACM, CSI, AIRCC, and IAENG (International Association of Engineers). His areas of interest include Internet of Things, data mining, networking, mobile computing, big data, and software engineering. Chintan has more than 10 years of teaching experience and research experience, having good teaching and research interests. He has more than 80 publications on the Internet of Things, computer vision, and software engineering, among which many publications are Scopus and WoS indexed. Chintan has been awarded with many CSI National Awards and a few CHARUSAT Research Paper Awards.

**Yulei Wu** is a senior lecturer in the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, United Kingdom. He received his PhD degree in computing and mathematics and BSc (First Class Hons.) in computer science from the University of Bradford, United Kingdom, in 2010 and 2006, respectively. His main research interests include networking, Internet of Things, edge intelligence, privacy and trust, and AI and ethics. Dr. Wu serves as an associate editor of *IEEE Transactions on Network and Service Management* and *IEEE Transactions on Network Science and Engineering*. He is a senior member of the IEEE and the ACM, and a fellow of the HEA (Higher Education Academy).

**Saad Harous** obtained his PhD in computer science from Case Western Reserve University, Cleveland, OH, USA, in 1991. He has more than 30 years of experience in teaching and research in three different countries: the USA, Oman, and UAE. Saad is currently a professor in the College of Computing and Informatics at the University of Sharjah. His teaching interests include programming, data structures, design and analysis of algorithms, operating systems, and networks. His research interests include parallel and distributed computing, P2P delivery architectures, wireless networks, VANET, and the use of computers in education and processing

Arabic language. He has published more than 200 journal and conference papers. Saad is an IEEE senior member.

**Massimo Villari** is Full Professor of Computer Science at the University of Messina (Italy). He is actively working as IT security and distributed systems analyst in cloud computing, virtualization, and storage and is one of the creators of Osmotic Computing Paradigm. For the EU project “RESERVOIR,” he led the IT security activities of the whole project. For the EU project “VISION-CLOUD” and H2020-BEACON, he covered the role of architectural designer for UniME. He was Scientific ICT responsible in the EU project frontierCities, the Accelerator of FIWARE on Smart Cities – Smart Mobility. He is strongly involved in EU Future Internet initiatives, specifically cloud computing and security in distributed systems. He is co-author of more of 190 scientific publications and patents in cloud computing (Cloud Federation), distributed systems, wireless network, network security, cloud security and cloud, and IoTs, and recently in osmotic computing. He was general chair of ESOC 2015 and IEEE-ISCC 2016. Since 2011, he is a fellow of IARIA, recognized as a cloud computing expert, and since 2011, he is also involved in the activities of the FIArch, the EU Working Group on Future Internet Architecture. In 2014, was recognized by an independent assessment (*IEEE Cloud Computing Transaction*, Issue April 2014) as one of worldwide active scientific researchers, top 27 classification, in cloud computing area. He was general chair of IEEE-ICFEC 2019 and workshop co-chair of IEEE-CIC 2018. He was general chair of IEEE CCGRID in 2021 in Messina (Sicily). Currently, he is the head of Computer Science School and Rector delegate on ICT for the entire University of Messina. He also covers the role of academic consultant for the City Council of Messina in the context of smart city.

# Chapter 1

## A Systematic Survey on Security Challenges for Fog-Enabled Internet of Things (IoT) and Industrial Internet of Things (IIoT)



Seema B. Joshi and Shaileshkumar D. Panchal

### 1.1 Introduction

The industry is going through the revolution phase; adoption of automation because of the significance of data is going to be increased nowadays. Almost every field is moving towards the adoption of better solutions to deal with the challenges of age. Industry 4.0 aims to enhance and upgrade the current manufacturing processes, decision-making system, data acquisition system and bringing of intelligence with the help of technological interventions such as the Internet of Things (IoT), Internet of Services (IoS), 3D technology, robotics and simulation tools to leverage the real-time data and mirror real world into the virtual model, big data analytics and augmented reality (AR) [1]. The Internet of Things (IoT) is a rapidly growing technology connecting emerging smart devices like mobile, machines and sensors. As per [2], 22 billion IoT devices were in use around the world, and based on the advancement of the hardware and software in the consumer electronics domain, it forecasts that 38.6 billion devices by 2025 and 50 billion devices by 2030 will be in use in the world. In the last decade, the average computational and processing requirements of the end user are rapidly increasing. To meet the growing demands, researchers have moved towards offloading the services to a centralized location, which is known as a cloud environment. The cloud computing environment extends permanent big storage necessities along with high computing power to meet the increasing requirement, but considering the wide acceptance of the IoT/IIoT scenario in the era of industry 4.0, latency is a desired quality in today's real-time applications. The challenges of cloud computing are:

---

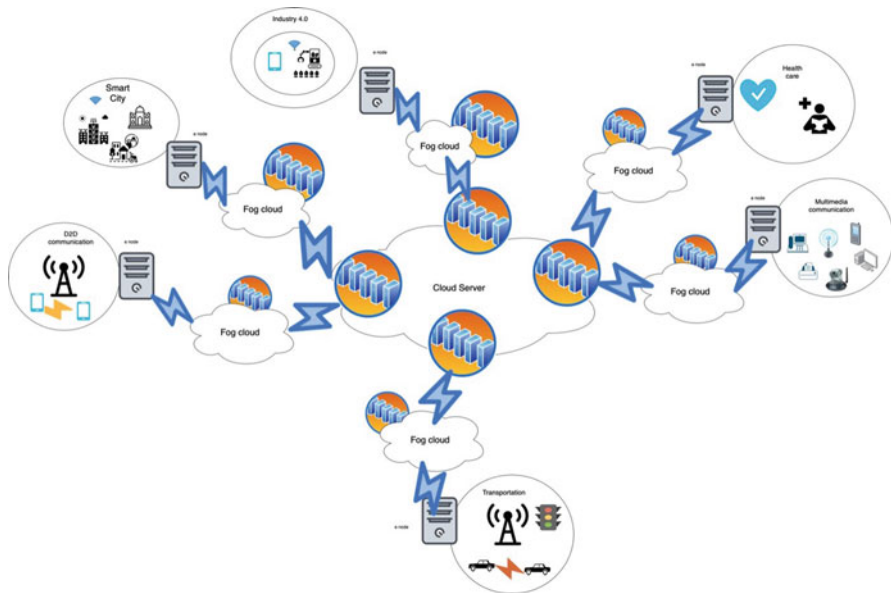
S. B. Joshi (✉) · S. D. Panchal  
Graduate School of Engineering and Technology, Gujarat Technological University, Ahmedabad,  
Gujarat, India  
e-mail: [ap\\_seema@gtu.edu.in](mailto:ap_seema@gtu.edu.in); [sdpanchal@gtu.edu.in](mailto:sdpanchal@gtu.edu.in)

- Remoteness between edge devices and cloud servers results in increased transmission and propagation delay.
- Speedily increases of IoT/IIoT-enabled devices, resulting in more computing power requirement, cause overstrain on cloud server consequently processing, and queuing delay will increase.
- Bandwidth requirement is increasing along with the increment of IoT/IIoT-enabled devices.
- Heterogeneity of the smart devices brings access, and configuration challenges in the cloud environment sometimes become a potential threat in the system which results in security challenges for the whole system.
- Offloading the computing requirement in the cloud causes delay, energy loss and reducing battery lifetime.

To mitigate the challenges, the need for a local computing environment has emerged, which can able to process the data locally in IoT/IIoT domains. Fog computing is assumed as an intermediate layer between cloud computing and IoT devices which extends the cloud services at the edge of the organization's network and doing the local processing [3]. Fog computing is a new emerging decentralized approach that can be deployed anywhere within the network edge, and fog devices can be any device that has computing power, storage and networking connectivity. The fog computing infrastructure not only addresses the issue of transmission and propagation delay of cloud but also resolving the availability, processing and queuing delay. At the same time, the advancement in communication fields enables to deal with bandwidth requirement issue in IoT and IIoT environment. With the emergence of 6G technology, the IoT/IIoT era is reinstated by the Internet of Everything (IoE) era, where ubiquitous connectivity would be possible to handle the bandwidth-related challenges [4]. The rest of the chapter are organized as follows: Sect. 1.2 includes a generalized view of fog computing and the Industrial Internet of Things. Section 1.3 stated fog computing-enabled IIoT applications. In Sect. 1.4, the security challenges of fog-enabled IoT and IIoT are discussed. The requirements of zero trust security in fog-enabled IoT and IIoT environments are discussed in Sect. 1.5. Finally, in Sect. 1.6, the chapter is concluded with several findings.

## **1.2 A Generalized View of Fog Computing and Industrial Internet of Things**

Due to the tremendous increase of IoT/IIoT-enabled smart devices, voluminous data is generating daily. Fog computing is an extended form of cloud computing, in respect of the industrial revolution giving applications and services at low latency, high processing at the edge of the network. The fog computing infrastructure was deployed at the edge of networking, which brings storage, maintenance and intelligence control to the proximity of the data devices. Figure 1.1 demonstrates the generalized view of cloud computing, fog computing and IIoT computing



**Fig. 1.1** A generalized view of IIoT, fog and cloud computing environment

environment. In a fog computing environment, data is processed locally with a single server which not only improves transmission, propagation delay and security but also helps in achieving instantaneous trustworthy communication. As data is processed locally, it keeps the overall system security within the premises. For each premise, smart devices within the periphery connect with the own fog server locally, and due to which fog computing is a cost-effective solution compared to a cloud computing environment.

As shown in Fig. 1.1, the device named e-node is near the data-generating smart devices, which has more computing power and equipped with intelligent controllers. Within premises, the heterogeneous smart devices are connected wired or wirelessly and uses the computing power of e-node to improve latency, reliability, privacy and security issues. As per Fig. 1.1, fog computing infrastructure is serving as a middle layer between end-user smart devices and the cloud computing environment where e-node play a vital role to provide an interface to data-generating devices and cloud servers. For end user, the whole architecture works as a ubiquitous computing environment with a seamless experience. In many countries, 5G technology standardization is completed, and it is underused for various applications using fog-enabled environment. In the future, the emergence of 6G technology which is capable of providing more bandwidth will surely prove added advantages to real-time applications running in fog-enabled environment to mitigate bandwidth-related issues of current technology.



### 1.3 Fog Computing-Enabled IIoT Applications

Fog computing is a relatively new and emerging computing environment and can be considered as an extension of a cloud computing environment. It is expected that it will gradually become popularized to satisfy the need for Industry 4.0 requirement. There are many real-time IoT/IIoT-based applications that get the benefits of the potential and affordable solution provided with the help of fog computing. The M2M communication is the key requirement of Industry 4.0, where machines are communicating with each other using the network infrastructure. The heterogeneity and interoperability among the connected IoT/IIoT devices bring challenges to the designers. In IoT/IIoT applications, the most challenging design parameters include energy constraint, latency issue, device deployments, throughput, integration, device maintenance, scalability, mobility, security, safety and privacy. IoT/IIoT is considered a rapidly growing innovative technology that attracts Industry 4.0 requirements. Exponential growth in IoT/IIoT applications in various fields results in high-speed Internet connectivity with a reducing latency period. The term fog computing is the extension of cloud computing architecture. The fog computing infrastructure offers the following advantages in comparison with cloud computing:

- Real-time connectivities with minimal latency
- Availability of computing resources at the edge of network periphery
- Enhanced security due to proximity and robust encryption algorithm
- Data storage on the network edge nodes eliminating transmission delay
- Higher data processing and analysing capabilities

The advancement from 5G to 6G communication technology will also be the added advantages for the IoT/IIoT applications, and it helps to make the fog computing technology for acting as an enabler of Industry 4.0 requirements. The smart cities, smart industry, smart products, healthcare sector, etc. are the main IoT/IIoT beneficial examples. Figure 1.2 shows the various sectors like transportation, infrastructure development and developing smart cities and building, in healthcare fields, to deal with environmental and climate changes issues, where the potential of IoT/IIoT applications is found which makes human life easier and comfortable.

### 1.4 Security Challenges of Fog-Enabled IoT and IIoT

Security challenges in fog computing are a key issue. In this section, the existing security threats and solutions of different layers of fog computing hierarchy are highlighted as shown in Fig. 1.3. As per the literature concerned, there is a sort of attacks that are related to IoT, IIoT and fog paradigm. The threat is the possibility of an unexpected malicious attempt with the clear intention of damaging the network

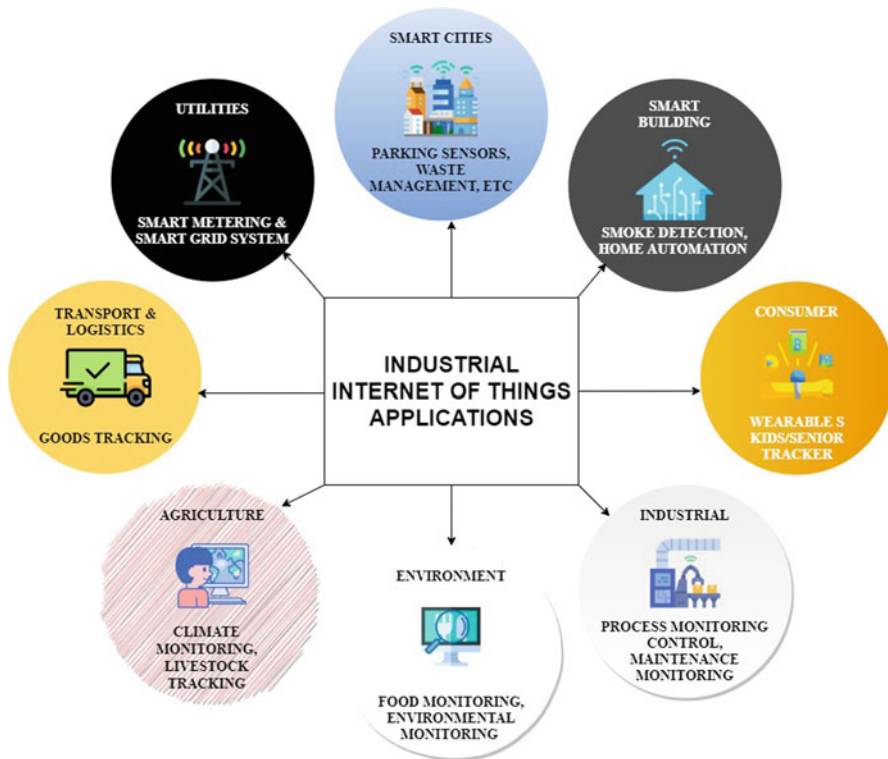


Fig. 1.2 Industrial IoT sectorial applications

system. It is something which might not have occurred but has the potential to directly affect the essential pillars of safety (CIA). The actualization of threat attack, when a threat turns into reality, it turned into an attack. In other terms, the known possibility of attack may be a threat. Therefore, it’s evident that threats and attacks go side by side. A list of the potential security threats of the fog environment is taken into consideration with existing solutions as shown in Fig. 1.3. The classification is done based on the top-down layered approach of fog computing.

### 1.4.1 Application Layer (Fog Server)

The application layer and business layer can be considered as a fog server. IoT and IIoT application deployment platforms are used to differentiate between various applications such as health, transportation, banking and SCADA for industrial automation. The protocols involved in the application layer are MQTT, AMQP, CoPA and XMPP which face the risk of threats. The possible security threats of

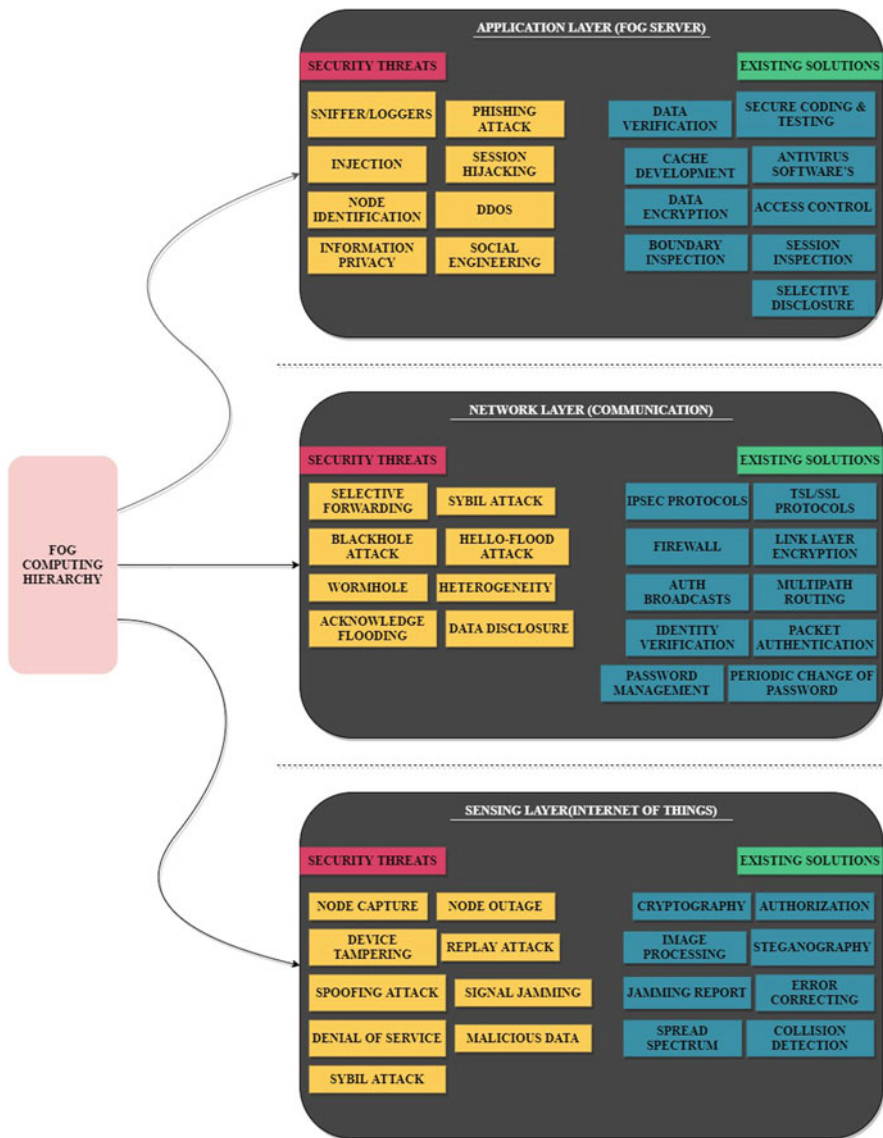


Fig. 1.3 Fog computing layered classification with security threats and existing solutions

application layer (fog server) of fog-enabled IoT and IIoT environment are shown in Table 1.1.

The existing solutions to overcome security threats of fog-enabled IoT and IIoT application layer include secure coding/programming and testing, use of updated antivirus software, cache development, data verification, intrusion detection system

**Table 1.1** Application layer security threats of fog-enabled IoT and IIoT environment

Security threats	Techniques
Sniffing attack/packet loggers	The attackers use packet sniffing for data extraction by capturing the network traffic analysis
Phishing attack	This is a kind of social engineering attack which is used to steal the user-sensitive information such as login credentials, credit card information, etc.
Injection	It is one of the most common attacks by injecting infected codes into the application executed on the server. This attack can result in loss of data and damage the application integrity
Session hijacking	The attacker hijacks someone else's identity and further gains access to personal identities
Distributed denial of service (DDoS) attack	Distributed denial of service (DDoS) attack occurs when multiple infected systems are used to damage a single system
Information privacy	The data loss and long-term damage of the system happened due to the vulnerable data protection techniques
Application-specific vulnerabilities	The vulnerabilities left during the application development due to ignorance of secure coding practice can be later exploited by attackers
Social engineering	Attackers gain sensitive information by befriending the users or by gaining user's trust and later misusing their information

(IDS), intrusion prevention system (IPS), firewall, session inspection, boundary inspection, cryptography and risk assessment. However, the fog server is a front end of the fog hierarchy, and therefore different security standards are needed as per the specific applications. More research is required in developing protocols as well as in cryptographic techniques to secure the fog-enabled IoT and IIoT environment.

### ***1.4.2 Network Layer (Communication)/Middleware***

This is the combination of the network layer and transport layer to work as a communication medium of fog computing. This layer is also considered middleware. The data received from the sensing layer are processed at middleware and transmitted to the application layer/fog server for further evaluation. The potential security threats of the network layer/middleware of fog-enabled IoT and IIoT environment are shown in Table 1.2.

The existing solutions to overcome security threats of fog-enabled IoT and IIoT network layer include TLS/SSL protocols (secure transport layer), IPsec protocols (secure network layer), IPS, PPSK, firewall, identity verification and packet authentication, multipath routing, link-layer encryption, password management and authentication policies. However, the primary and essential challenge of the

**Table 1.2** Network layer security threats of fog-enabled IoT and IIoT environment

Security threats	Techniques
Selective forwarding	This attack is performed by a malicious node to drop the data packets, and infected nodes randomly skip the routing data packets
Sybil attack	In this attack, one device plays the role of multiple identities to reduce the efficacy of fault-tolerant schemes
Black hole attack	A piece of unfaithful routing information is created in this attack to divert all the data packets to the sinkhole. This may cause packet drop and network congestion
Hello-Flood attack	To create network congestion, the attacker floods the channel with false data packets to create network congestion. Also, every malicious node persuades its neighbour to participate in packet transmission
Acknowledge flooding	This is similar to the denial of service (DoS) attack where the attacker sends the fake information to neighbouring nodes using acknowledgement
Scalability	The congestion and depletion of resources and lack of authentication mechanism through the untraceable number of connecting/disconnecting devices
Data disclosure	The attacker uses data retrieval techniques to extract sensitive information from nodes, which can lead to data privacy risks

network layer is designing IoT and IIoT middleware compatible for cloud and edge computing environments to support various IoT and IIoT applications.

### 1.4.3 Sensing Layer (*Internet of Things*)

The sensing layer is the bottom layer in the three-layered architecture. This is the combination of a physical layer and the data link layer for the communications stack. The potential security threats of the sensing layer of fog-enabled IoT and IIoT environment are shown in Table 1.3.

The existing solutions to overcome security threats of fog-enabled IoT and IIoT sensing layers include cryptography, steganography, authentication, authorization, spread spectrum communication, image processing, jamming report, error-correcting codes and collision detection. However, many open security challenges associated with sensing layers where the IoT devices deployed are required to focus. For example, IoT is an emergent platform where the integration of millions of computing devices and massive real-time data is sensed from these devices. These devices are powerful, compact, costly and globally connected. So, it is essential to monitor each object adding to the IoT network to detect and prevent malicious object risk. Moreover, limitations of sensing layer security associated with network protocol, hardware devices and 5G- to 6G-oriented communication channels are required to focus to secure the data routing and processing.

**Table 1.3** Sensing layer security threats of fog-enabled IoT and IIoT environment

Security threats	Techniques
Node capture or device tampering	Attackers gain unauthenticated access through weakened IoT gateway
Spoofing attack	To get full access to the systems, the attacker masquerades the data and sends fake data to the network
Signal jamming	Through signal jamming, interference is generated in communication between network devices with the radio frequencies
Malicious data	A malicious node infects the whole system by spreading malicious data
Denial of service attack/path-based DoS	Denial of service attack floods sensor nodes by injecting replayed and false packets. This attack results in exhaustion of batteries, network resources and cut down of the system service availability
Node outage	Node outage leads to loss of connectivity through the cut down of most of the devices in the network
Replay attack	In a replay attack, the original data packets are replaced by the false data packets. In this way, attackers put the network trust and authentication at risk
Sybil attack	In the Sybil attack, the aggregate message is changed to a false message. Due to this, negative reinforcements are created by malicious nodes

There are certain attacks such as code injection, data leakage, denial of service (DoS) and man-in-the-middle attack that are addressed by most of the researchers. Considering the information thefts that occur within the network, the approach is proposed by Stolfo et al. [4] to monitor the information access patterns through user behaviour profiling and keeping track of activities of malicious insiders. The user behaviour profiling was used in [5], as a key to intact safety by using the hybrid protocol that supported selective encryption and data cleaning. Li et al. [6] have devised a non-cooperative differential game-theoretic framework that estimates the changing behaviour of malicious nodes. They also quantify the value associated with the danger generated and analyse the strategy to scale back energy consumption and ensure QoS of the network. Butun et al. [7] have discussed the inference of using fog computing as a basic architecture of IoT keeping cyberspace in point of view. Diro et al. [8] have given a distributed deep learning-based IoT-fog network attack detection system. They also performed an experiment during which they found that their proposed attack detection system is performing well compared to a centralized detection system using deep learning.

Sohal et al. [9] have proposed a cyber-security framework that performs early prediction of the malicious edge devices using a two-state Markov model. The results generated after this framework support the effectiveness of the framework. Wang et al. [10] have given a fog-based scheme that divides the info into two parts: The larger one is shipped to the cloud, and therefore the smaller one is kept within the fog. In this way, the scheme claims to make sure the supply, confidentiality, and

integrity of the information. Shankarwar and Pawar [11] have listed various security threats that occur in cloud computing environments about user's sensitive data. The researchers have given various techniques to tackle issues by applying different approaches. They also discussed the pros and cons of the prevailing methods.

Khan et al. [12] have analysed the danger penetration and therefore the precautionary measures for the scholars using the Internet. They need also given a mind map of varied issues associated with security that need to be taken care of while considering cybercrime. Maimo et al. [13] have given a MEC-oriented architecture for network anomaly detection with the assistance of policies. They had also deployed the proposed detection technique using a deep learning approach to review the flow of the network and detect the anomalies occurring within the network. It was also experimentally proven that their proposed approach works effectively for anomaly detection and also adapts to the created detection module in a real-time and automatic way. Gandhi et al. [14] have given a name as HIoT POT that secures the IoT environment that also helps white hats to get new methodologies employed by black hats. Further, Ziegeldorf et al. [15] have firstly given a brief review of pertinent security issues; then, they categorized the prevailing issues in seven broad categories. Later, they need to discuss the challenges that occur, thanks to the above-stated threats, and have addressed the necessity to beat these threats. Zhang et al. [16], in their editorial paper, have discussed that a bulk of research is taking interest in working with threat management by managing intrusion detection. Gai et al. [17] have acknowledged different intrusion detection techniques and analysed the challenges existing in these schemes. They also presented a high-level security framework that uses IDS techniques for providing security to mobile cloud-based solutions during a 5G network. Yaseen et al. [18] have proposed a model for the detection of selective forward attack that's quite common in mobile WSNs. The model provides global IDs which will detect MWSN in their trace and refute the malicious nodes. They have tested their mechanism over the CloudExp simulator. The results came out better in terms of routing overhead and power consumption. The model proposed by them gives secure, low-cost and on-demand access to the infrastructure within the IoT network. Alrawais et al. [19] have addressed security and privacy issues that are present in fog computing and IoT scenarios. Authentication, trust, and rouge node detection are few of the threats that are addressed by the researchers. Later, the proposed scheme was improvised for the security enhancement of IoT devices. Lin et al. [20] proposed an intrusion detection system for fog computing including the demand of resource allocation inspection. Further, the single-layer dominant and max-min fair allocation scheme is proposed for single-layer multi-resource fair allocation systems, whereas multilayer dominant and max-min fairies is proposed for multilayer resource allocation issues.

Impact layer's dimensions of possible security threats can be considered as theft of operational information, damage to property, loss of availability, loss of control, loss of view, loss of productivity and revenue, manipulation of view, denial of view, loss of safety, denial of control and manipulation of control.



## 1.5 Zero Trust Architecture Concept and Its Requirement in Fog-Enabled IoT and IIoT

According to the chosen literature, trust plays a serious role in promoting interaction between the entities within the network [21, 22]. It is an important aspect as cyber-physical systems need to depend upon the services and resources that are under the ownership of edge, fog and cloud computing [23]. The emergence of fog computing has led to varied advances within the technological era [24]. Although this also introduces certain complexities within the existing scenario, the benefits of fog have a foothold over them [25–27]. With the existence of such an intermediate layer and therefore the progressive growth of technology, security is the biggest setback. Fog computing has certain similarities with cloud computing, but it is very distinct in other ways. It deals with various privacy and security issues aside from ones that are carried forwards from the cloud.

According to John Kindervag, Field CTO at Palo Alto Networks ‘Trust is always a vulnerability in a digital system’ [28]. The traditional data centre security practices involve trusted and untrusted domains of network segmentation. The zero trust architecture eliminates the idea of trusted/untrusted devices, network and users [29]. In the zero trust architecture, all network traffic is untrusted irrespective of the source. Zero trust covers security rules for seven stacks – networks, devices, data, people, workloads, automation and orchestration and visibility and analytics. It applies security protocols and policies on all network entities to secure all resources, limits and enforces access control and inspects and logs network traffic.

Blockchain is often applied to the implementation of IoT security. IoT device identity and network attributes are often stored on the blockchain-based distributed ledger to secure them from Sybil and spoofing attacks [30]. Machine learning plays an essential role, while blockchain facilitates information collection under the premise of knowledge regulation rules such as privacy protection. With a comprehensive understanding of machine learning and blockchain usage in industrial sectors, practical aspects of diversified services can be possible [31]. IoT device transactions are stored on the blockchain to guard their integrity and for centralized management and governance. Smart contracts can implement security policies at each node. Blockchain-based key management and distribution can eliminate the complex computation and high memory requirements involved in implementing security in an IoT network [32].

Zero trust is characterized by segmented, parallelized and centralized network based on three key concepts that empower secure networking:

- *Ease of segmented network management:* Zero trust recommends new ways of segmenting network hierarchy.
- *Multiple parallel switching core development:* Zero trust recommends the development of multiple smaller and less expensive cores by breaking the core switch. Zero trust segregates network traffic into smaller network segments by using the concept of distributed processing.



- **Single console central management:** Zero trust recommends a platform to manage all networking elements centrally and segment network traffic.

The holistic framework of zero trust and blockchain is proposed for IoT security, which may be helpful to spot the security solutions of fog-enabled IoT and IIoT environments. This proposed architecture is designed with the concepts of layered security and a defence-in-depth approach to eliminate the single points of failure and security compromise [33]. The following are the key concepts and components:

- *Segmentation gateway (SG)* forms the nucleus of the network in zero trust. It provides all security functions such as firewall, network access control, data loss prevention, intrusion prevention, intrusion detection, VPN gateway and others. Segmentation gateway implements all global network and security policies. It provides secure and parallel network segment management with network traffic segregation with the ability to early detection and containment of security incidents.
- *Microcore and perimeter (MCAP)* proposes the fine-grained parallel segmentation and isolation of critical network resources. MCAP is connected with SG using a microcore switch. The same set of functionalities and network policy attributes are shared by network resources in each microcore.
- *MCAP's centralized unified and transparent management* is a key feature of zero trust. In effect, zero trust shifts the paradigm of network management from an individual network component management to a centralized network management system.

The zero trust architecture concept can be essential to the fog-enabled IoT and IIoT platform in the form of risk assessment, secure access control mechanism and reporting management. The researchers can be inspired to improve the trust in the virtually connected smart world of Industry 4.0 with the combination of zero trust architecture and blockchain technology.

## 1.6 Conclusion

In this chapter, the fog computing architecture is briefed along with its benefits over the cloud computing environment. It includes the challenges faced by the IoT/IIoT applications using cloud computing and how they are overcome up to a certain extent with the help of a fog computing environment by providing intermediate computing infrastructural facilities. A systematic view is presented with a major specialization of fog computing architecture and therefore the threats that are pervasive in fog-enabled IoT and IIoT paradigm. The possibility of 5G to 6G advancement in the communication domain and its direct benefits to fog-enabled environment are also presented. It is concluded that many architectures which addressing this area and applicable as per the suitability of the smart devices to which it's being deployed, majority of them are still conceptual on the grounds of

fog computing being in its infancy. This takes us to the existence of various security and privacy issues that need to be addressed because various threats which can exist within the considered scenario are surveyed in literature. It has been found that the emergence of this layer and the increased surface have exponentially augmented the possibilities of attacks. Certain threats like the man-in-the-middle attack, Sybil, etc. are addressed by various researchers that directly affect the network established by the fog. Further, it has been observed that trust may be a crucial component of any communication that happens online. Various researchers have also supported this fact and stated that the consideration of trust even at the fog level is of utmost importance. Further research paths are often aimed towards the event of security and zero trust establishing schemes with blockchain technology to measure the reliability of the fog-enabled IoT and IIoT environment presented in the chapter.

## References

1. Lampropoulos, G., et al. (2019). Internet of things in the context of Industry 4.0: An overview. *Scienco-International Journal of Entrepreneurial Knowledge*, 7(2), 4–19.
2. Statista Research Department. (2021). *Number of internet of things (IoT) connected devices worldwide in 2018, 2025 and 2030*. Accessed 27 February, 2021. <https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/>
3. Aleisa, M. A., Abuhussein, A., & Sheldon, F. T. (2020). Access control in fog computing: Challenges and research agenda. *IEEE Access*, 8, 83986–83999.
4. Stolfo, S. J., Salem, M. B., & Keromytis, A. D. (2012). Fog computing: Mitigating insider data theft attacks in the cloud. In *2012 IEEE symposium on security and privacy workshops*. IEEE.
5. Sriram, M., et al. (2014). A hybrid protocol to secure the cloud from insider threats. In *2014 IEEE international conference on cloud computing in emerging markets (CCEM)*. IEEE.
6. Li, Z., et al. (2017). A non-cooperative differential game-based security model in fog computing. *China Communications*, 14(1), 180–189.
7. Butun, I., Sari, A., & Österberg, P. (2019). Security implications of fog computing on the internet of things. In *2019 IEEE international conference on consumer electronics (ICCE)*. IEEE.
8. Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, 82, 761–768.
9. Sohal, A. S., et al. (2018). A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. *Computers & Security*, 74, 340–354.
10. Wang, T., et al. (2018). Fog-based storage technology to fight with cyber threat. *Future Generation Computer Systems*, 83, 208–218.
11. Shankarwar, M. U., & Pawar, A. V. (2015). Security and privacy in cloud computing: A survey. In *Proceedings of the 3rd international conference on Frontiers of intelligent computing: Theory and applications (FICTA) 2014*. Springer.
12. Khan, N. S., Chishti, M. A., & Saleem, M. (2019). Identifying various risks in cyber-security and providing a mind-map of network security issues to mitigate cyber-crimes. In *Proceedings of 2<sup>nd</sup> International conference on communication, computing and networking*. Springer.
13. Maimó, L. F., et al. (2018). Dynamic management of a deep learning-based anomaly detection system for 5G networks. *Journal of Ambient Intelligence and Humanized Computing*, 10(8), 3083–3097.
14. Gandhi, U. D., et al. (2018). HIoTPOT: Surveillance on IoT devices against recent threats. *Wireless Personal Communications*, 103(2), 1179–1194.

15. Ziegeldorf, J. H., Morchon, O. G., & KlausWehrle. (2014). Privacy in the internet of things: Threats and challenges. *Security and Communication Networks*, 7(12), 2728–2742.
16. Zhang, X., et al. (2019). Intrusion detection and prevention in cloud, fog, and internet of things. *Security and Communication Networks*, 2019, 4529757.
17. Gai, K., et al. (2016). Intrusion detection techniques for mobile cloud computing in heterogeneous 5G. *Security and Communication Networks*, 9(16), 3049–3058.
18. Yaseen, Q., et al. (2018). Leveraging fog computing and software defined systems for selective forwarding attacks detection in mobile wireless sensor networks. *Transactions on Emerging Telecommunications Technologies*, 29(4), e3183.
19. Alrawais, A., et al. (2017). Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, 21(2), 34–42.
20. Lin, F., et al. (2018). Fair resource allocation in an intrusion detection system for edge computing: Ensuring the security of Internet of Things devices. *IEEE Consumer Electronics Magazine*, 7(6), 45–50.
21. Zhang, P. Y., Zhou, M. C., & Fortino, G. (2018). Security and trust issues in Fog computing: A survey. *Future Generation Computer Systems*, 88, 16–27.
22. Liu, Y., Fieldsend, J. E., & Min, G. (2017). A framework of fog computing: Architecture, challenges, and optimization. *IEEE Access*, 5, 25445–25454.
23. Soleymani, S. A., et al. (2017). A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing. *IEEE Access*, 5, 15619–15629.
24. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30–39.
25. Byers, C. C. (2017). Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled iot networks. *IEEE Communications Magazine*, 55(8), 14–20.
26. Wang, Y., Uehara, T., & Sasaki, R. (2015). Fog computing: Issues and challenges in security and forensics. In *2015 IEEE 39th annual computer software and applications conference* (Vol. 3). IEEE.
27. Kumari, A., et al. (2019). Fog data analytics: A taxonomy and process model. *Journal of Network and Computer Applications*, 128, 90–104.
28. Sam Greengard. (2018). *SRT Interview: John Kindervag Says 'Put Your Trust in Zero Trust'*. Accessed 24 February, 2021. <https://www.securityroundtable.org/john-kindervag-put-trust-zero-trust/>
29. Kindervag, J. (2010). *Build security into your network's DNA: The Zero Trust network architecture*. Forrester Research Inc. Accessed 24 February, 2021. [http://www.virtualstarmedia.com/downloads/Forrester\\_zero\\_trust\\_DNA.pdf](http://www.virtualstarmedia.com/downloads/Forrester_zero_trust_DNA.pdf)
30. Wu, Y., Dai, H.-N., & Wang, H. (2021). Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. *IEEE Internet of Things Journal*, 8(4), 2300–2317. <https://doi.org/10.1109/JIOT.2020.3025916>
31. Yulei, W., Wang, Z., Ma, Y., & Leung, V. C. M. (2021). Deep reinforcement learning for blockchain in industrial IoT: A survey. *Computer Networks*, 191, 108004., ISSN 1389-1286. <https://doi.org/10.1016/j.comnet.2021.108004>
32. Wu, Y. (2020). Cloud-edge orchestration for the internet-of-things: Architecture and AI-powered data processing. *IEEE Internet of Things Journal*, 8(16), 12792–12805. <https://doi.org/10.1109/JIOT.2020.3014845>
33. Dhar, S., & Bose, I. (2020). Securing IoT devices using zero trust and blockchain. *Journal of Organizational Computing and Electronic Commerce*, 31(1), 18–34.

# Chapter 2

## Evolutionary Algorithms for Enhancing Mobile Ad Hoc Network Security



G. M. Jinarajadasa and S. R. Liyanage

### 2.1 Fog Computing and Mobile Ad-hoc Networks (MANETs)

It is expected that the increased utilization of smart devices will reach around 10–12 billion by 2021 [1]. To satisfy this huge demand of future IoT applications, faster communication and higher computational capacities are needed for processing and sharing of data. Therefore, the existing computing paradigms should be updated to adapt to the situation and to fulfill the requirement of fast communication and high computation capacity to enable the emerging utilization of smart and mobile devices. Fog computing along with the fifth-generation (5G) networks is expected to be one of the most effective solutions for fulfilling this requirement. A complete utilization of 5G network technologies has not materialized mainly due to problems related to security, privacy control, and network traffic [2, 3]. Therefore, the emerging 6G technologies that direct fog computing toward intelligent edge computing are considered as the future of communication to facilitate faster and higher capacity networks that can overcome the current limitations [4].

Wireless networks have been one of the most progressive technologies in the last 50 years in the field of information technology. Wireless connectivity plays an important role in the implementation of IoT applications. Mobile ad hoc networks (MANETs) are one of the popular applications of wireless networks that is integrated with modern fog computing and 5G technologies. Cloud-based MANETs or C-MANETs are one of the most popular applications that are implemented along with the emerging 5G and 6G technologies that enable applications such as IoT Cloud-MANETs and intelligent vehicular ad hoc networks [5]. Therefore, a review on the mobile ad hoc networks and their applications along with fog computing technologies is presented in this section.

---

G. M. Jinarajadasa (✉) · S. R. Liyanage  
Faculty of Computing and Technology, University of Kelaniya, Kelaniya, Sri Lanka

### ***2.1.1 Preface to MANETs***

Mobile ad hoc networks (MANETs) are one of the in-demand categories of wireless ad hoc networks, which is a decentralized wireless network category. The key features of a MANET are:

- Self-configuration ability
- Being wireless
- Having an infrastructure-less or decentralized environment
- Having a dynamic topology that is changing continuously from time to time [6]

Hence, a MANET is declared as a consistently self-configuring, less-structured network that consists of wirelessly connected autonomous mobile equipment instead of using wires, where the mobile devices act as network nodes. Each node in a MANET has the capability of moving independently in any direction. Therefore, frequent changing of the links to other nodes in the network can happen where it creates a dynamic topology that changes periodically. Each node in a MANET can behave as both a host and a router. This behavior of individual nodes leads to autonomously acting nodes. The nodes of a MANET can join or leave the network anytime as they are mobile. This leads to the rapidly changing network topology of MANETs. The mobile nodes of MANETs are configured with less power, less energy, less memory, and other lightweight features. All the nodes of a MANET environment have identical features with similar capabilities and responsibilities, leading to a symmetric network environment. MANETs are capable of multi-hop routing that is helpful when a packet-sending node and a packet-receiving node are out of the transmission range [7].

### ***2.1.2 Types and Applications of MANETs***

Diverse types of MANETs are available other than the traditional ones, such as vehicular ad hoc network (VANET)s [8], smartphone ad hoc network (SPAN)s, Internet-based mobile ad hoc network (iMANET)s, etc. VANETs are the specific type of MANETs utilized to communicate among vehicles and roadside equipment which provides comfort and safety to drivers in vehicular environments. The concept of VANET has been evolved into the emerging artificial intelligence techniques. An extension to the VANETs named intelligence vehicular ad hoc network (InVANET) has been introduced in the recent decade. InVANET technologies have been applied for intelligent navigations during emergencies and intelligent behaviors to avoid vehicular collisions and other accidents [9].

SPANs are wireless ad hoc networks that utilize the available technologies in mobile phones such as Bluetooth and Wi-Fi. It creates multi-hop network transmissions so that any node has the capability of joining and exiting the network without affecting the steadiness of the network [10]. The iMANETs connect the

mobile nodes with fixed gateway nodes in such a way that they can create a geographically distributed set of MANETs [11]. Other applications of MANETs can be seen in various fields, including information networks in the military sector, ship-to-ship and marine communications in the commercial sector, home networks, and data networks [12].

### 2.1.3 Routing Protocols of MANETs

Routing is a crucial factor in MANETs because of the dynamic changes in the network structure. Therefore, it is important to have a routing protocol that can match the dynamicity of the network structure and enable the capability of each node equipped for going about as a router. Further, because of the restricted data transfer capacities of nodes, the routing path among source and destination may consist of multiple hops. Due to this limitation, the data has to be communicated to the intermediate nodes. MANET routing has been a popular research area for years. Some of the major research thrusts pertaining to MANETs have been to overcome the MANETs’ asymmetric connections, high routing overhead, interference, and routing misbehaviors occurring due to dynamic topology. The existing routing approaches can be classified into nine classes as follows:

- Reactive/source-initiated/on-demand routing protocols
- Proactive/table-driven routing protocols
- Hybrid routing protocols
- Hierarchical routing protocols
- Multipath routing protocols
- Multicast routing protocols
- Location-aware protocols
- Geographical multicast protocols
- Power-aware protocols [13]

Figure 2.1 displays the different groups of the MANET routing protocols with some example routing protocols under each group. Among them, DSR, AODV,

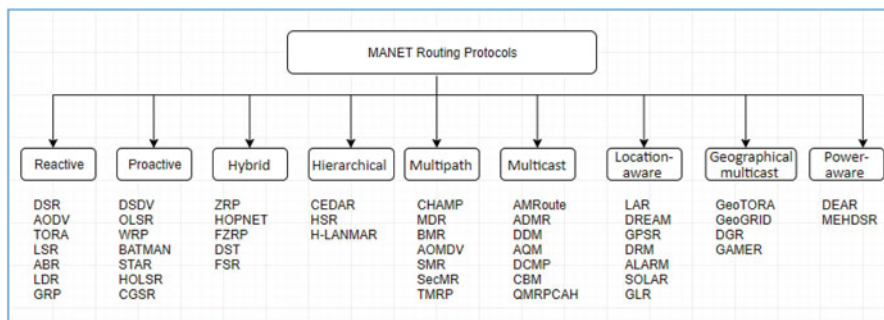


Fig. 2.1 MANET routing approach classification

and TORA are the most popular reactive routing protocols. DSDV, OLSR, and BATMAN are popular proactive mobile ad hoc routing protocols [14].

### ***2.1.4 Future Trends in MANETs Toward FOG Computing***

The exponential growth of ubiquitous wireless devices has eliminated the demand for many wired devices and has triggered an increasing demand for novel wireless devices. Before the third millennium, mobile ad hoc networks were not very popular. Though, throughout the past decade, the need for research and development on MANETs as academic research grew, its application in commercial ventures also gained momentum. This growth can be attributed to the emerging applications in vehicular networks, military communication applications, etc.

With the pervasive growth of mobile computing, MANETs have possessed the ability to seamlessly integrate with heterogeneous environments, including different types of networks and devices. The marriage of MANETs with cloud computing concepts has created a new research area into embedding FOG computing [15] to the MANET concepts, where it already has led to innovations and field experiments [16]. Kai et al. present a critical survey on merging concepts of fog/cloud computing into VANETs in different aspects, including various paradigms, scenarios, and issues [16].

## **2.2 Security Problems in MANETs**

Though MANETs have become popular recently, there is still a gap to be filled in the aspect of security that would help to establish and improve reliable communication links in them. MANETs have a set of characteristics that may lead to vulnerabilities of malicious attacks and unreliable and insecure communications among the mobile nodes. Some of them occur due to:

- Mobile nodes
- Dynamic topology
- Wireless links and limited physical security
- Cooperativeness – threats from compromised nodes inside the network
- Resource constraints
- Scalability and decentralized management
- Different requirements for different applications
- Lack of clear line of defense [17, 18]

Because of the generated vulnerabilities from these characteristics, MANETs are having more exposure to various attacks that include both passive and active attacks. The following are some common passive and active attacks that can occur in a MANET environment [19, 20]:

- Passive attacks – Eavesdropping, traffic analysis, and snooping
- Active attacks – Flooding attack, blackhole attack, wormhole attack, gray-hole attack, denial of service attack, and selfish misbehavior of nodes

Since MANETs are more prone to malicious attacks, ensuring “security” is a major issue. Plenty of research work and experiments have been carried out to ensure the security level of MANETS. These approaches have considered diverse aspects such as the context of trust, mitigating security attacks, trust/secure routing protocols, optimization approaches to find the optimal route, and avoiding malicious attacks. Recent developments in this area have attempted various machine learning approaches to address the security in MANETS. Section 2.3 provides an overview of existing approaches [21–41].

### 2.3 Different Approaches for Enhancing Security in MANETs and IoT

Various types of early research solutions can be found in establishing security in MANETs and IoT. The existing solutions found in literature can be categorized into few major groups such as:

- Machine learning – Supervised learning, unsupervised learning, and reinforcement learning
- Swarm intelligence
- Evolutionary algorithms
- Mobile agents
- Real-time heuristics
- Genetic algorithms
- Neural networks
- Decision trees
- Probabilistic models
- Other theoretical methods – Fuzzy logic, watchdog method, and scalable maturity models

Q-Learning [21] is a reward-based reinforcement learning algorithm that is widely used. Application of Q-learning in wireless networking has been common due to easy implementation and its good balance of memory and energy requirements, where it adapts to the resource constraints (power/energy and memory) of the network [22–24]. Various types of reinforcement learning mechanisms have been presented to enhance the MANET security with the means of generating secure routing protocols such as DRQ routing with dual RL[25], TPOT reinforcement learning [26, 27], and collaborative RL [28].

Many approaches that are inspired by the swarm intelligence methods like ant colony optimization (ACO) [29] and ant-based control (ABC) have been tested in the field of MANET security. AntNet [30] and AntHocNet [31] are popular



**Table 2.1** Comparison of different distributed approaches on MANETs

Property	Machine learning	Swarm intelligence	Mobile agents	Heuristics
Memory requirement	Medium	Medium	Low	Medium
Computational requirement	Medium	Medium	Low	Low
Flexibility to topology changes	High	High	Medium	Medium
Accuracy of results	High	High	N/A	Medium
Initial cost	High/medium	High	Low	High
Additional cost	Low	Medium	Medium	Low

**Table 2.2** Comparison of machine learning techniques on MANETs

Machine learning technique	Capturing dynamicity	Additional costs	Optimality of results
SVM	Low	High	High
Q- Learning	High	High	High
TPOT-RL	High	Medium	High
Dual RL	High	Low	High
Collaborative RL	High	Low	High

solutions generated for MANETs and wireless ad hoc networks that have utilized the ACO and ABC optimization techniques. SmartAgents [32] and Ant-AODV [33] are applications of mobile agents in MANETs, for designing the optimal routing approaches by agents finding the new paths, updating the routes, and collecting the next-hop node's information in the network.

Other than these approaches, real-time heuristic search mechanisms [34] have been applied for the route optimization of the MANETs [35, 36]. Real-time heuristic search mechanisms require minimal computational power so they are well suited to the ad hoc environments. Tables 2.1 and 2.2 show simple comparisons on characteristics of different distributed approaches and characteristics of machine learning approaches applied to MANETs, respectively.

Security enhancement of IoT services is essential with the promising IoT technologies. DQSP is a QoS-aware routing protocol for the novel IoT concept called SDN-IoT that is implemented by leveraging the deep reinforcement learning (DRL) methods [37]. DRL-based methods have been found to produce proactive, efficient, and intelligent routing that adapts to dynamic traffic changes when applied to SDNs [38].

In [39], Kore et al. have provided a sound analysis on the security of wireless sensor networks joined with IoT focusing on energy efficiency and the security measures of IoT. A novel routing protocol for Intelligent-IoT networks is proposed in [40]. It uses artificial intelligence as the underlying technology and has gained remarkable results in the network life span and reduced delays compared to other mechanisms. A novel routing protocol named MTISS-IoT derived based on ad hoc on-demand distance vector (AODV) routing protocol using cryptographic authentication is proposed by Mabodi et al. in [41]. It has shown remarkable detection rates for gray-hole attacks through NS-3.

## 2.4 Evolutionary Algorithms

Evolutionary algorithm (EA)s are optimization algorithms that search for optimal solutions by evolving multiset of candidate solutions. These are population-based metaheuristics that demonstrate a biological process like mutation, recombination, and natural selection to determine the optimal solution by being within the specified constraints [42]. A general evolutionary algorithm consists of several key components that are:

- Initial population via random generation
- Fitness function
- Evolution – Selection, crossover/generation of the offspring and evaluation the fitness, and mutation
- Termination by reporting the optimal solution generated by the best-fit individual [43]

When considering the existing optimization methods, population-based search methods are split into two classes as evolutionary algorithms and swarm intelligence (Fig. 2.2). The popular classes of EAs include:

- Genetic algorithm (GA)s
- Evolution strategies (ES)
- Differential evolution (DE)
- Estimation of distribution algorithm (EDA)s
- Multi-objective evolutionary algorithm (MOEA)s
- Memetic algorithm (MA)s
- Genetic programming (GP)
- Learning classifier systems (LCS) [44]

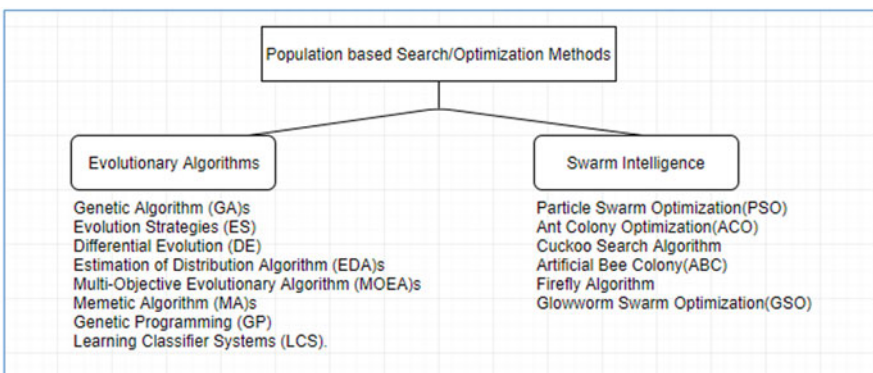


Fig. 2.2 Classification of Population-based Search Methods

Other than these EAs, a wide use of swarm optimization methods such as PSO and ant colony optimization (ACO) combined with EAs can be seen in the scientific literature.

### **2.4.1 Genetic Algorithm (GA)s**

The GAs are the most popular and commonly used type of EAs and also the earliest derived type of evolutionary algorithms. The process of GA includes random initialization of the population to begin the interaction. Then, a set of steps are conducted that are objective function evaluation, parent selection, application of genetic operations, creation of offspring via recombination, and mutation to generate a new population. These means are reshaped until an end condition is fulfilled [43, 45].

### **2.4.2 Differential Evolution (DE)**

The DE is a variation of EAs that accomplish the strength in enhancement technique and fast merging to an ideal answer for enhancement in a numeric value. In contrast to EAs, distinct feature that can be seen in the DE occurs in the evolution process wherein the generation of new solutions, mutation act as the primary operating factor and crossover as the secondary operating factor [43, 46].

### **2.4.3 Evolution Strategies (ES)**

The evolution is modeled by the ES as an interaction of the versatile conduct of the individual that keeps up the social linkage among parents and their offspring, separately, in the individual state. ES focuses on numerical optimizations by utilizing real variables, and it relies majorly on the mutation [43, 47].

## **2.5 Applications of Evolutionary Algorithms in MANETs and IoT**

Evolutionary algorithms can be utilized to overcome different types of problems related to MANETs and IoT networks. Evolutionary algorithms have been applied in MANETs to solve diverse optimization problems in multiple aspects, such as:

- Topology maintenance
- Broadcasting programs

- Routing protocols and protocol optimization
- Security, malicious attacks, or selfish behavior of nodes
- Mobility and mobility models
- Clustering approaches [48]

In this section, we focus on evolutionary algorithmic approaches applied for MANETs and IoT networks in terms of enhancing security. The enhancement of the security of a MANET can be achieved in different ways, including intrusion detection, avoiding or mitigating malicious attacks, maintaining the stable network topology, efficient broadcast of the messages, route optimization, trustworthy clustering, and tracking the mobility of the nodes. The following subsections illustrate the research approaches that use evolutionary algorithms in the areas related to mobile ad hoc network security.

### ***2.5.1 Enhancing MANET and IoT Security by Identifying Malicious Attacks with EA***

Intrusion detection, identifying different types of attacks on a MANET, and using risk-avoidance or risk-mitigating solutions would help to ensure the security of a MANET. Several studies have attempted evolutionary algorithmic techniques to identify malicious attacks [49–56].

Incorporation of neutrosophic rules into the GAs for upgrading an efficient malicious behavior detection system has been found to increase the threat identification capacity and decrease the false warning rate in MANETs in [49]. Three major facts have been considered to detect the attack: membership, nonmembership, and neutrosophic indeterminacy degree. A hybrid attack inference system for MANETs has been proposed by combining self-organizing feature maps (SOFM) and GAs [49].

A novel technique to analyze and identify abnormal behaviors in a MANET environment that utilizes the ad hoc on-demand distance vector (AODV) routing mechanism is proposed in [50]. The suggested GA-based solution monitors the node behavior of all nodes. It provides the details about potential attacks and also shows good results in detecting the common malicious attacks, which are similar to blackhole attacks.

To simulate the IDS (intrusion detection systems) that generates a set of best trade-offs between the criteria of the security and the power consumption, genetic programming (GP) can be blend together with a multi-objective evolutionary algorithm (MOEA) [51]. A novel method named hybridization of particle swarm optimization with genetic algorithm (HPSO-GA) routing system is proposed by Thanuja et al. in [52], to detect blackhole attack with the utilization of the AODV approach. The accuracy measure of the suggested mechanism is measured considering the set of filtered parameters in network environment.

In [53], researchers are designing a solution to generate security through the factor of “trust” and proposing an algorithm along with the differential evolution named trusted-differential evolution algorithm where it manages adversary nodes and represses them to turn into an individual member of the data transmission route. The proposed approach consists of two major components that are to locate the most fitting route and to manage the rising trustworthiness of nodes via reliability.

In [54], Sen et al. have investigated the utilization of evolutionary computation strategies, especially genetic programming and grammatical evolution, to advance interruption monitoring mechanisms for demanding networks similar to MANETs. Aware of the specific significance of energy, they have examined the power utilization of developed projects and utilize a multi-objective evolutionary solution to identify ideal compromises between interruption discovery capacity and energy consumption.

A survey consists of critical analysis on evolutionary computational (EC) methods for cybersecurity of MANETs conducted in [55] to discuss basic defense mechanisms in detail that can be followed to detecting vulnerabilities, deterrence of attacks, prevention and recovery, and risk mitigation.

An ensemble method-based novel routing protocol capable of identifying jamming attacks that can occur in IoT-based cognitive radio networks is presented in [56]. The results of the conducted experiments show that the proposed method improves the CRN performance against proactive jamming attacks.

### ***2.5.2 Enhancing MANET and IoT Security by Secure Routing and Protocol Optimization***

Evolutionary algorithmic calculations have been broadly utilized for the boundary setup of routing conventions. The goal is to locate the ideal parameters for efficient routing. In [57], the researchers are experimenting with a few multi-objective optimization methods to advance a straightforward route path discovery convention that discovers routing paths between two nodes in the MANET. The non-dominated sorting-based genetic algorithm II (NSGA-II) and the multi-objective differential evolution (MODE) are taken into account in the proposed approach to advance the cost of energy and the performance metrics of the mean jitter. As per the outcomes in [44], MODE calculation is fit for discovering routing paths closer to the set of optimal solutions and, in general, unites quicker than NSGA-II.

In [58], researchers have focused on a differential evolution algorithm that is put together with an ad hoc on-demand multipath distance vector (DE\_AOMDV) approach for MANETs. Suggested DE\_AOMDV routing convention has sound execution and increases the node connectivity in MANETs. The principal goal of the proposed solution is to locate the ideal path from various routes which are accessible between the source and destination nodes to be utilized in the process of recovering the routing.

In the investigation explained in [59], an objective algorithmic structure is planned to utilize a hybrid optimization algorithm, named M-LionWhale, for reliable routing. It is an optimization model that consolidates the lion algorithm (LA) into whale optimization algorithm (WOA) for the ideal determination of the route in MANETs. Several specific quality of service (QoS) parameters are considered in creating the multi-objective optimization model. Similarly considering QoS requirements for the information flow of the various IoT services, an intelligent routing protocol is introduced in [60]. The proposed protocol can identify the category of the information flow and the relevant QoS requirements beforehand. In [61], another fog-enabled QoS-aware intelligent resource management approach is proposed for an IoT-based home automation system by leveraging particle swarm optimization. It has demonstrated a remarkable reduction in network bandwidth, latency, response time, and energy consumption.

The paper [62] is suggesting an ACO and P-coding-based routing protocol to MANETs considering security and energy, which decreases the usage of energy in nodes by cutting the security cost.

In designing a security model for MANETs, an approach has been illustrated in [63] where it acquires the elective route path or reinforcement route to keep away from rerouting disclosure on account of connection or node failure. A genetic algorithm that focuses on the goal of designating near best route from packet-forwarding node to packet-receiving node dependent on schedule is suggested in [63]. The proposed algorithm acts as a MANET route optimization mechanism. The cluster heads amplify the usage of the algorithm and make the packet delay into the least [63].

### ***2.5.3 Enhancing MANET Security by Topology Management***

Maintaining the topology of a MANET affects in different ways to the security of the network since the topology of MANET can continuously vary within time. In [64], the researchers utilize a solo optimization method to get the ideal states and speed of several helper nodes in a rail route-station situation. The goal is to expand the warning distance at which a train moving toward the station gets the data. A PSO algorithm is utilized in [65] to send portable nodes, named agents, to increase MANET availability. Those agents additionally foresee the future behavior of nodes dependent on the nodes' states and their speed rates. Subsequently, the enhancement approach discovers optimal future situations for the agent nodes as indicated by the present and future conditions of the MANET. The network connectivity capacity is measured as the mean connectivity ratio of the nodes.

Focusing on game theory and brute-force GA, an evolutionary game called NSEG has been proposed in [66]. In this method, the objective of every network node is to disseminate itself over an obscure topographical territory to get a high wrapping level of the region by the network nodes. This will allow to accomplish

a uniform distribution of nodes while maintaining the associativity of the network [66].

### ***2.5.4 Enhancing MANET Security by Broadcasting Algorithms***

Broadcasting can be defined as one of the fundamental all-to-all communication mechanisms that are utilized commonly within mobile ad hoc networks [67]. The principal goal of a broadcasting program is to proficiently spread a data packet through the entire MANET, which will help establish security. In [68], the creators utilize the NSGA-II multi-objective algorithm [69] to enhance the plan of a broadcasting model dependent on closeness/difference coefficients.

The article [70] targets deciding the best correspondence methodologies for every node as per its density of neighborhood nodes. It depicts an apparatus joining network simulator 2 and an evolutionary algorithm (EA). Abdou et al. have considered five facts. For each of them, by monitoring the behavior of each node, the best and proper input parameters are determined. The proposed novel EA is compared to the existing three popular EAs that are NSGA-II, SPEA2, and DECHEMA-SQP, and with the evaluated results, the new EA is applied in the MANET broadcasting process.

### ***2.5.5 Enhancing MANET Security by Node Clustering***

Creating trustworthy clusters and identifying the trusted cluster heads play a vital role in achieving security by establishing and disseminating the trust among the MANET nodes. A novel node clustering mechanism is proposed in [71] considering the utilization of energy-efficient routing protocol in MANETs. It is implemented by keeping the focus on improving the reliability of information transmission with a high-security measure that uses an optimization algorithm. A modern discrete PSO algorithm is utilized to determine the most trustworthy cluster head. An encryption technique is used to guarantee the transmission security for establishing reliable links in a trusted MANET [71].

## **2.6 Conclusion**

Enhancing security in MANETs is a complex problem which has to consider various aspects of a MANET's environmental behavior. These include secure and optimal routing, malicious attack and other risk mitigation, topology maintenance,

node clustering, broadcasting, mobility tracking, etc. The utilization of evolutionary algorithms in solving security issues in MANETs has become popular in the past decade. Though MANETs still contain plenty of problems that can be researched in depth, EA approaches can be utilized to find the solutions. The chapter itself has introduced the primary highlights and limitations that ought to be thought about the utilization of evolutionary algorithms in MANETs. There are certain challenges occurring in applying evolutionary algorithms due to the energy and other resource constraints of the MANETs and the distributed nature. But hopefully, with the ubiquitous growth of mobile computing, that would increase the computational power of wireless devices. Therefore, the improvement of the EAs, by applying completely distributed powerful evolutionary algorithms to enhance security in MANETs, would be possible in the near future. Furthermore, the improvements of security and privacy in MANETs would support the future implementations of IoT Cloud MANETs. These advances can be utilized to enhance the security and privacy of 5G and 6G networks to overcome their existing challenges.

## References

1. Akpakwu, G. A., Silva, B. J., Hancke, G. P., & Abu-Mahfouz, A. M. (2017). A survey on 5G networks for the Internet of Things: Communication technologies and challenges. *IEEE Access*, 6, 3619–3647.
2. Meng, Y., Naeem, M. A., Almagrabi, A. O., Ali, R., & Kim, H. S. (2020). Advancing the state of the fog computing to enable 5g network technologies. *Sensors*, 20(6), 1754.
3. Al-Ansi, A., Al-Ansi, A. M., Muthanna, A., Elgendy, I. A., & Koucheryavy, A. (2021). Survey on intelligence edge computing in 6G: Characteristics, challenges, potential use cases, and market drivers. *Future Internet*, 13(5), 118.
4. Wang, M., Zhu, T., Zhang, T., Zhang, J., Yu, S., & Zhou, W. (2020). Security and privacy in 6G networks: New areas and new challenges. *Digital Communications and Networks*, 6(3), 281–291.
5. Alam, T. (2020). Cloud-MANET and its role in software-defined networking. *Transactions on Science and Technology*, 7(1), 1–7.
6. Basagni S, Conti M, Giordano S, Stojmenovic I (eds) (2004) Mobile ad hoc networking.
7. Basagni, S. (2013). *Mobile ad hoc networking*. Wiley.
8. Sharma, B., Sharma, M., & Tomar, R. (2019). A survey: Issues and challenges of vehicular ad hoc networks (VANETs). *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3363555>
9. Zeadally, S., Hunt, R., Chen, Y., et al. (2010). Vehicular ad hoc networks (VANETS): Status, results, and challenges. *Telecommunication Systems*, 50, 217–241. <https://doi.org/10.1007/s11235-010-9400-5>
10. Mitra, P., & Poellabauer, C. (2012). Emergency response in smartphone-based mobile ad-hoc networks. In *IEEE international conference on communications (ICC)*.
11. Corson, M., Macker, J., & Cirincione, G. (1999). Internet-based mobile ad hoc networking. *IEEE Internet Computing*, 3, 63–70. <https://doi.org/10.1109/4236.780962>
12. Lakhtaria, K. I. (2012). *Technological advancements and applications in Mobile ad-hoc networks: Research trends: Research trends*. IGI Global.
13. Rajeswari, A. R. (2020). A mobile ad hoc network routing protocols: A comparative study. In *Recent trends in communication networks*. IntechOpen.
14. Abolhasan, M., Wysocki, T., & Dutkiewicz, E. (2004). A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, 2, 1–22. [https://doi.org/10.1016/s1570-8705\(03\)00043-x](https://doi.org/10.1016/s1570-8705(03)00043-x)



15. Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37–42).
16. Kai, K., Cong, W., & Tao, L. (2016). Fog computing for vehicular ad-hoc networks: Paradigms, scenarios, and issues. *The Journal of China Universities of Posts and Telecommunications*, 23, 56–96. [https://doi.org/10.1016/s1005-8885\(16\)60021-3](https://doi.org/10.1016/s1005-8885(16)60021-3)
17. Sarika, S., Pravin, A., Vijayakumar, A., & Selvamani, K. (2016). Security issues in mobile ad hoc networks. *Procedia Computer Science.*, 92, 329–335.
18. Mokhtar, B., & Azab, M. (2015). Survey on security issues in vehicular ad hoc networks. *Alexandria Engineering Journal*, 54, 1115–1126. <https://doi.org/10.1016/j.aej.2015.07.011>
19. Soni, M. R., Dahiya, A. K., & Verma, M. S. (2016). Security issues and attacks in mobile ad hoc networks. *International Journal of Engineering Research and Technology*. <https://doi.org/10.17577/ijertv5is120189>
20. Singh, M., Singh, A., Tanwar, R., & Chauhan, R. (2011). Security attacks in mobile ad hoc networks. In *IJCA Proceedings on National Workshop-Cum-Conference on Recent Trends in Mathematics and Computing*.
21. Watkins, C. (1989). *Learning from delayed rewards*.
22. Beyens, P., Peeters, M., Steenhaut, K., & Nowe, A. (2005). Routing with compression in WSNs: A Q-learning approach. In *Proceedings of the 5th Eur. Wksp on adaptive agents and multi-agent systems (AAMAS)*.
23. Boyan, J. A., & Littman, M. L. (1994). Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in Neural Information Processing Systems, 1994*, 671–678.
24. Sun, R., Tatsumi, S., & Zhao, G. (2002). Q-map: A novel multicast routing method in wireless ad hoc networks with multiagent reinforcement learning. In *2002 IEEE region 10 conference on computers, communications, control and power engineering. TENC'02. Proceedings-2002 Oct 28 (Vol. 1, pp. 667–670)*. IEEE.
25. Kumar, S., & Miikkulainen, R. (1997). Dual reinforcement Q-routing: An on-line adaptive routing algorithm. *Proceedings of the artificial neural networks in engineering Conference, 1997*, 231–238.
26. Stone, P., & Veloso, M. (1999). Team-partitioned, opaque-transition reinforcement learning. In *Proceedings of the third annual conference on Autonomous Agents* (pp. 206–212).
27. Stone, P. (2000). TPOT-RL applied to network routing. In *ICML* (pp. 935–942).
28. Dowling, J., Curran, E., Cunningham, R., & Cahill, V. (2005). Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(3), 360–372. <https://doi.org/10.4249/scholarpedia.1461>
29. Dorigo, M. (2007). Ant colony optimization. *Scholarpedia*, 2, 1461. <https://doi.org/10.4249/scholarpedia.1461>
30. Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9, 317–365. <https://doi.org/10.1613/jair.530>
31. Di Caro, G., Ducatelle, F., & Gambardella, L. (2005). AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16, 443–455. <https://doi.org/10.1002/ett.1062>
32. Bonabeau, E., Henaux, F., Guérin, S., Snyers, D., Kuntz, P., & Theraulaz, G. (1998). Routing in telecommunication networks with ant-like agents. In *International workshop on intelligent agents for telecommunication applications* (pp. 60–71). Springer.
33. Marwaha, S., Tham, C. K., & Srinivasan, D. (2002). Mobile agents based routing protocol for mobile ad hoc networks. In *Global Telecommunications Conference, 2002. GLOBECOM; 02. IEEE - 17 Nov 2002 (Vol. 1, pp. 163–167)*. IEEE.
34. Koenig, S. (2001). *Agent-centered search*. AI Magazine.
35. Forster, A. (2007). Machine learning techniques applied to wireless ad-hoc networks: Guide and survey. In *2007 3rd international conference on intelligent sensors, sensor networks and information* (pp. 365–370). IEEE.

36. Rossi, M., Zorzi, M., & Rao, R. (2006). Statistically assisted routing algorithms (SARA) for hop count based forwarding in wireless sensor networks. *Wireless Networks*, 14, 55–70. <https://doi.org/10.1007/s11276-006-7791-8>
37. Guo, X., Lin, H., Li, Z., & Peng, M. (2019). Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT. *IEEE Internet of Things Journal*, 7(7), 6242–6251.
38. Casas Velasco, D., Caicedo Rendon, O.M. and da Fonseca, N.L.S., 2021. DRSIR: A deep reinforcement learning approach for routing in software-defined networking.
39. Kore, A., & Mishra, M. R. (2020). A review on joint IoT and WSN security for achieving the less energy consumption. *International Journal of Scientific & Technology Research*, 9(2).
40. Al-Janabi, T. A., & Al-Rawashidy, H. S. (2018). A centralized routing protocol with a scheduled mobile sink-based AI for large scale I-IoT. *IEEE Sensors Journal*, 18(24), 10248–10261.
41. Mabodi, K., Yusefi, M., Zandiyani, S., Irankehah, L., & Fotohi, R. (2020). Multi-level trust-based intelligence schema for securing of internet of things (IoT) against security threats using cryptographic authentication. *The Journal of Supercomputing*, 76(9), 7081–7106.
42. Davis, L. (1999). *Evolutionary algorithms*. Springer.
43. Janga Reddy, M., & Nagesh Kumar, D. (2021). Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: A state-of-the-art review. *H2Open Journal*, 3(1), 135–188.
44. Corne, D. W., & Lones, M. A. (2018). Evolutionary algorithms. *arXiv*.
45. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press.
46. Reddy, M. J., & Kumar, D. N. (2012). Computational algorithms inspired by biological processes and evolution. *Current Science*, 103(4), 1–11.
47. Fogel, D. B. (1994). An introduction to simulated evolutionary optimization. *IEEE Transactions on Neural Networks*, 5(1), 3–14.
48. Dorronsoro, B., Ruiz, P., Danoy, G., Pigné, Y., & Bouvry, P. (2014). *Evolutionary algorithms for mobile ad hoc networks*. Wiley.
49. Elwahsh, H., Gamal, M., Salama, A., & El-Henawy, I. (2018). A novel approach for classifying MANETs attacks with a Neutrosophic intelligent system based on genetic algorithm. *Security and Communication Networks*, 2018, 1–10. <https://doi.org/10.1155/2018/5828517>
50. Sujatha, K. S., Dharmar, V., & Bhuvaneshwaran, R. S. (2012). Design of genetic algorithm based IDS for MANET. In *2012 international conference on recent trends in information technology* (pp. 28–33). IEEE.
51. Şen, S., Clark, J. A., & Tapiador, J. E. (2009). Power-aware intrusion detection in mobile ad hoc networks. In *International conference on ad hoc networks* (pp. 224–239). Springer.
52. Thanuja, R., & Umamakeswari, A. (2018). Black hole detection using evolutionary algorithm for IDS/IPS in MANETs. *Cluster Computing*, 22, 3131–3143. <https://doi.org/10.1007/s10586-018-2006-5>
53. Prabha, S., & Yadav, R. (2019). Trusted-differential evolution algorithm for mobile ad hoc networks. In *Recent trends in communication, computing, and electronics 2019* (pp. 181–193). Springer.
54. Sen, S., & Clark, J. (2011). Evolutionary computation techniques for intrusion detection in mobile ad hoc networks. *Computer Networks*, 55, 3441–3457. <https://doi.org/10.1016/j.comnet.2011.07.001>
55. Kusyk, J., Uyar, M., & Sahin, C. (2018). Survey on evolutionary computation methods for cybersecurity of mobile ad hoc networks. *Evolutionary Intelligence*, 10, 95–117. <https://doi.org/10.1007/s12065-018-0154-4>
56. Salameh, H. B., Otoum, S., Aloqaily, M., Derbas, R., Al Ridhawi, I., & Jararweh, Y. (2020). Intelligent jamming-aware routing in multi-hop IoT-based opportunistic cognitive radio networks. *Ad Hoc Networks*, 98, 102035.
57. Yetgin, H., Cheung, K. T., & Hanzo, L. (2012). Multi-objective routing optimization using evolutionary algorithms. In *2012 IEEE wireless communications and networking conference (WCNC) 2012 Apr 1* (pp. 3030–3034). IEEE.

58. Sharma, A., & Sinha, M. (2019). A differential evolution-based routing algorithm for multi-path environment in mobile ad hoc network. *International Journal of Hybrid Intelligence*, 1, 23. <https://doi.org/10.1504/ijhi.2019.10021294>
59. Chintalapalli, R., & Ananthula, V. (2018). M-LionWhale: Multi-objective optimisation model for secure routing in mobile ad-hoc network. *IET Communications*, 12, 1406–1415. <https://doi.org/10.1049/iet-com.2017.1279>
60. Sun, W., Wang, Z., & Zhang, G. (2021). A QoS-guaranteed intelligent routing mechanism in software-defined networks. *Computer Networks*, 185, 107709.
61. Gill, S. S., Garraghan, P., & Buyya, R. (2019). ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices. *Journal of Systems and Software*, 154, 125–138.
62. Dwivedi, N., & Shukla, R. (2017). *Evolutionary algorithm based optimized encryption scheme for mobile Ad-hoc network*.
63. Nikhil, K., Agarwal, S., & Sharma, P. (2012). *Application of genetic algorithm in designing a security model for mobile ad hoc network*. Departement of IT, ABES Engineering College.
64. Gutiérrez-Reina, D., Toral Marín, S., Johnson, P., & Barrero, F. (2012). An evolutionary computation approach for designing mobile ad hoc networks. *Expert Systems with Applications*, 39, 6838–6845. <https://doi.org/10.1016/j.eswa.2012.01.012>
65. Dengiz, O., Konak, A., & Smith, A. (2011). Connectivity management in mobile ad hoc networks using particle swarm optimization. *Ad Hoc Networks*, 9, 1312–1326. <https://doi.org/10.1016/j.adhoc.2011.01.010>
66. Kusyk, J., Sahin, C., Umit Uyar, M., et al. (2011). Self-organization of nodes in mobile ad hoc networks using evolutionary games and genetic algorithms. *Journal of Advanced Research*, 2, 253–264. <https://doi.org/10.1016/j.jare.2011.04.006>
67. Reina, D., Toral, S., Johnson, P., & Barrero, F. (2015). A survey on probabilistic broadcast schemes for wireless ad hoc networks. *Ad Hoc Networks*, 25, 263–292. <https://doi.org/10.1016/j.adhoc.2014.10.001>
68. Reina, D., León-Coca, J., Toral, S., et al. (2013). Multi-objective performance optimization of a probabilistic similarity/dissimilarity-based broadcasting scheme for mobile ad hoc networks in disaster response scenarios. *Soft Computing*, 18, 1745–1756. <https://doi.org/10.1007/s00500-013-1207-3>
69. Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6, 182–197. <https://doi.org/10.1109/4235.996017>
70. Abdou, W., Henriet, A., Bloch, C., et al. (2011). Using an evolutionary algorithm to optimize the broadcasting methods in mobile ad hoc networks. *Journal of Network and Computer Applications*, 34, 1794–1804. <https://doi.org/10.1016/j.jnca.2011.01.004>
71. Elhoseny, M., & Shankar, K. (2020). Reliable data transmission model for Mobile ad hoc network using Signcryption technique. *IEEE Transactions on Reliability*, 69, 1077–1086. <https://doi.org/10.1109/tr.2019.2915800>

# Chapter 3

## Blockchain-Based Fog Computing



Anusha Vangala and Ashok Kumar Das

### 3.1 Introduction

Fog computing is a distributed computing application consisting of a number of servers that can perform computation, networking and provide storage similar to the servers in a cloud data center. It essentially aims to bring server resources closer to the devices involved in the generation of data. It increases the intelligence of local area network by allowing computation of the data to be performed using the resource capabilities available inside the network where the data gathering devices exist. This helps to reduce latency in response times that is encountered in cloud computing where data was needed to be transmitted to servers placed in different geographical locations before any processing could begin. Fog computing has also allowed increased security of data by allowing highly sensitive data to be processed at fog servers and only low-sensitive data to be forwarded to the cloud server. It also promotes better management of huge volumes of data by distributing the data among multiple nodes in the local network.

Fog computing can be used in conjunction with cloud computing and edge computing. Cloud computing consists of multiple high-resource servers placed inside a data center owned by a service provider. Any user needing resources will associate with the provider and pay for the amount of resources used, without the need for delving into the details of managing these resources. Fog computing allows the processing to be performed at the local network of the data gathering devices. On the other side, edge computing allows the processing to be performed at either the devices that hold the sensors or a gateway node placed in close physical proximity to these sensor devices.

---

A. Vangala (✉) · A. K. Das

Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, India

e-mail: [anusha.vangala@research.iiit.ac.in](mailto:anusha.vangala@research.iiit.ac.in); [ashok.das@iiit.ac.in](mailto:ashok.das@iiit.ac.in)

Internet of Things (IoT) is a collection of highly diverse devices with the ability to read the physical parameters of their surroundings, process the data in a distributed manner, and perform collective actions based on the processed data, using the Internet and with minimal human intervention. Fog computing has major applications in the IoT world where a huge amount of data sensed by the IoT smart sensor devices are regularly sent to the cloud servers for processing. The working of IoT is highly reliant on real-time processing of sensor data as the user is in continuous interaction with the smart devices. In such a scenario, latency due to processing and network transmission may be highly deterrent to the smooth functioning of many IoT applications. Also, data in IoT applications are sensitive to the user and require protection from any unprecedented misuse. This shows that fog computing is supremely relevant in the context of IoT applications.

Blockchain is compatible with fog computing as it allows the devices (nodes) to be used as blockchain nodes and fog nodes to be used as miner nodes. The idle resources available with the nodes in a fog network can be used for blockchain maintenance. The nodes that allow their idle resources to use for blockchain processing can be rewarded in accordance with the amount of resources provided, by using an appropriate consensus mechanism. In recent years, the blockchain technology has been adapted in many other potential applications in order to enhance the security of a system, such as smart farming [70, 71], IoT and industrial IoT [7, 58], Internet of Everything (IoE) [12], Internet of Drones (IoD) [11, 13, 14], smart grids [15], healthcare applications [36, 59, 63, 74], Internet of Vehicles (IoV) [6], Intelligent Transportation Systems (ITS) [69], Internet of Intelligent Things (IoIT) [80], Software-Defined Networks (SDN) [23], supply chains [43], and military applications [82].

### ***3.1.1 Application Areas of Fog Computing***

Nikouei et al. [52] proposed an authentication scheme on a smart surveillance system that is based on generating pattern indexes of identified interesting objects and timestamps on a live streaming video. The resulting indexes can be stored on the cloud to be used for further heavy processing. This event-oriented processing of the live video surveillance is done at three levels: a) object detection and tracking; b) extraction of low-level features at network edge; and c) data aggregation at fog nodes and processing and cloud centers. This requires event-oriented surveillance video query, real-time indexing, and secure data transferring, and blockchain-enabled authentication.

In the first level of object detection and tracking, the video live video is captured and sent to the edge or fog nodes in real time. The edge nodes then detect anomalies by extracting low-level features in the objects and behavior with minimal false alarm rate considering the limited resources available. This may require running a person, object, vehicle (POV) algorithm that is resource-heavy and hence avoided on edge devices. More resource-efficient tracker algorithms with the pre-trained

convolutional neural network (CNN)–Deep Learning models [3] are used. At the second level, certain relevant descriptive metrics are defined for the objects identified at the first level. Based on the defined metrics, further processing such as contextualization, classification, and saving are performed.

In general, the metric definition is done at the edge node, and the metric processing is outsourced to fog nodes or cloud servers. In such a case of outsourcing, the metric data needs to be transferred from the edge nodes and fog nodes and requires two-level encryption with symmetric encryption, such as Advanced Encryption Standard (AES) [2] and RSA public key encryption [57], with shared key encrypted with the fog node's public key, to prevent network sniffing attacks. This is initiated by the edge node that sends a handshake request to the fog node that obtains its public key certificate in response. The edge node sends the encrypted shared key. The fog nodes decrypt the shared key using its private key and send the hashed shared key. Once the edge node verifies the hashed shared key, data exchange can commence. The shared key is discarded at the end of every data exchange session. The features extracted at the edge node will be encrypted and forwarded to the fog node. The fog nodes then place a spatio-temporal context to the received features for contextualization. These frame-wise data with the location, time, sequence, the number of objects, and gestures are stored as key–value pair in fog nodes with sufficient storage that allows fast retrieval. This level of indexing speeds up the process of querying the video and replaces the slow process of observing the full video to identify the moments of interest.

The fog layer then shares the indexing data with the cloud layer for higher level processing tasks. To ensure a more secure and decentralized sharing with support to scalability, a blockchain-enabled authentication service is used. Every entity in the network has an account identified by its public key, called the virtual identity (VID) that is used in the identity authentication and management in the cloud server. When a fog node sends registration request to the cloud server, a profile is created after verifying the fog node's identity credentials. The hashed index table data is managed by a smart contract, which is deployed in the blockchain network allowing all the nodes to transparently access the transactions on the chain. Once the registration information of the fog node is verified, its access request is evaluated according to the authorization policies. If the request is granted, a transaction is executed by the cloud to update the list of entities authorized in the smart contract. Once the transaction itself is approved, the fog node receives the address of smart contract and the recording function. To authenticate the video query data stored on the fog, a cloud operator checks the current state of the smart contract and obtains the hashed key–value index record. Thus, the sensor devices can detect events in a video, which are indexed based on extracted features and stored in a table that is hashed to prevent malicious modifications.

Fernandez-Carames and Fraga-Lamas [34] provided a detailed study of introducing IoT with fog computing using blockchain in the field of education to propel educational sector toward smart universities and campuses. They defined the essential characteristics needed for such smart education system, compared different architectures provided for such a system, studied the effect of introducing

blockchain, analyzed the existing applications in smart education, and then proposed new challenges that should be researched in smart education. Chaiyarak et al. [20] also proposed an architecture for smart management of education even during unprecedented disastrous situations, such as the Coronavirus Disease-2019 (COVID-19) pandemic. In current situation, COVID-19 becomes a very serious health concern to the human life throughout the world [22]. One prominent solution is the use of the Internet of Medical Things (IoMT) that allows to deploy several wearable Internet of Things (IoT)-enabled smart devices in a patient's body [31, 36, 39]. The deployed smart devices should then securely communicate to nearby mobile device installed in a smart home, which then securely communicate with the associated Fog server for information processing. The processed information in terms of transactions is formed as blocks and put into a private blockchain consisting of cloud servers. Since the patient's vital signs are very confidential and private, the private blockchain is best suited for such kind of applications.

A number of smart IoT applications are engulfed in the concept of a smart city, such as smart lighting, smart transportation, smart healthcare, and smart buildings. Singh et al. [61] proposed an overview of such a smart city model and derived a blockchain and fog-based architecture with detailed characteristics of requirements along with a model diagram. It studies the average power consumption, based on the number of smart devices, and provides a latency comparison of fog and cloud systems in a smart city environment.

Islam et al. [42] proposed an architectural framework based on human activity recognition (HAR) directed toward monitoring patients with mental illnesses remotely. The activities of the patient captured through video are analyzed based on multi-class categorization using support vector machines. The accuracy of this classification is improved with the addition of blockchain-based fog architecture.

Gul et al. [38] proposed a reward-based business model based on blockchain that predicts medical status about a patient. Fernandez-Carames and Fraga-Lamas [35] proposed a communication architecture to remotely monitor the glucose levels of patients continuously and warn the patient to take appropriate preventive measures. This architecture makes use of crowdsourcing in mobile health for distributed problem solving, and federated blockchains are used to decentralize the system against single point of failure and increase the transaction privacy as the transaction data include highly sensitive medical data about patients. Fog computing is used in order to collect sample data from the patients using distributed mobile smart phone systems.

Baniata et al. [9] proposed a task scheduling system that can be used to efficiently automate the scheduling of tasks in complex applications such as smart city where task scheduling is considered as NP-hard problem, which is a computationally infeasible task. This system uses an ant colony optimization (ACO) on fog computing assisted with blockchain technology that is highly privacy-aware and takes very less execution time along with tackling high network load.



### ***3.1.2 Main Contributions***

In this chapter, we provide the following main contributions:

- We first discuss the necessity of security in fog computing environment. It is needed mainly due to the fact that the data is required to be protected from several potential attacks against passive as well as active adversaries.
- We then discuss the evolution of blockchain in fog computing context.
- Various security and functionality requirements in fog computing environment are discussed.
- Next, we discuss a taxonomy of various security protocols in fog computing. Design of security protocols for communication in fog computing may fall into one or more security protocols.
- We also discuss the network and threat models that are useful in discussing the existing security protocols for blockchain-enabled fog computing environment.
- Finally, a comparative study among the discussed existing security protocols for blockchain-enabled fog computing environment has been conducted to measure effectiveness of the protocols.

### ***3.1.3 Chapter Organization***

The remainder of this chapter is organized as follows. The security vulnerabilities of fog computing are discussed in Sect. 3.2. Section 3.3 studies in detail how blockchain can be integrated with fog computing and the evolution of this process since the inception of the ideas of blockchain and fog computing. The security requirements essential in fog computing are analyzed in Sect. 3.4. Section 3.5 is dedicated to the study of the types of security protocols needed to be designed keeping in view the security vulnerabilities and requirements of fog computing. Section 3.6 enhances the generalized architecture for fog computing by incorporating blockchain technology. It also studies the different threat models that apply to such a blockchain-based architecture that can help us analyze the existing security schemes. Section 3.7 studies a plethora of security schemes in detail that have been developed for fog computing using blockchain technology. Section 3.8 examines the security strength of studied schemes. Section 3.9 concludes the chapter by summarizing the blockchain-based fog computing solutions.

## **3.2 Need for Security in Fog Computing**

Delegation of tasks to fog servers may cause some of the data to be available to these servers. Such data need to be protected against many attacks as defined in Sect. 3.4. Since fog servers are closer to the end users in the terminal layer, the surveillance



of the devices is relatively weak. This presents the requirement for better protection as the fog devices are much more prone to malicious attacks.

Stojmenovic et al. [65] presented a case study of how man-in-the-middle (MiTM) attack affects the system security in fog servers as the authors believe that this attack can potentially become the most common attack in fog computing. An experimental study is conducted by launching MiTM attack in four chronological steps to hijack communication in a fog-based system. They also studied the effect of intrusion detection based on anomaly detection by observing the memory consumption and CPU utilization of gateway node during the launched MiTM attack.

Ali et al. [4] studied that trust can be achieved only after the security goals of authentication, authorization, and privacy are achieved for each component in each level of the fog environment. Once trust is achieved, some dynamic method of identification is applied to each of the components in the environment. Butun et al. [18] mentioned that IoT as an environment is naturally prone to violation of user privacy due to the deep commingling of the user devices with other devices in the network. When such an environment is coalesced with a fog environment, the privacy violation is exacerbated due to the escalated complexity of determining the ownership of the huge amount of data circulated in the network.

Kaur et al. [44] identified that most of the security issues in fog computing correspond to the handing over of pre-processed data from the fog layer to the cloud layer address the need for lightweight security schemes compared to heavyweight schemes due to the significant resource limitations in the fog servers in the fog layer compared to cloud servers in the cloud layer. Mukherjee et al. [51] studied the comparison of cloud, edge, and fog computing along with the fog–cloud interface that allows the cloud layer to distribute the services to the fog servers in the fog layer in a resource-efficient manner.

From the above studies, it is clear that the security plays a very important role for protecting data in fog computing setting. The security of fog computing can be further enhanced by using the blockchain technology.

### 3.3 Blockchain and Its Evolution in Fog Computing

Baniata et al. [8] provided a detailed study on the integration of blockchains with fog computing and made observations that a majority of the applications of blockchains in fog computing were targeted toward maintaining data, followed by identity management, payment/trading, and reputation systems in IoT-related systems and used proof-based consensus algorithms with most of the applications using the Proof-of-Work (PoW) consensus protocol [10]. Integrating blockchain with fog computing requires a trade-off decision between the need for security, reliability, and decentralization with the cost of money, energy, and latency of using blockchains.

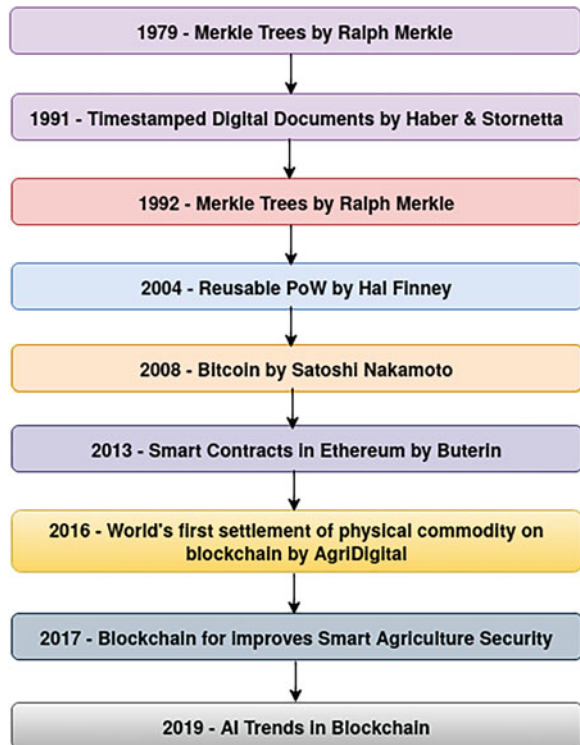
Uriarte et al. [68] studied the three blockchain-based fog solutions, a decentralized supercomputer, named Golem Network, a decentralized cloud named iExec

and a decentralized fog computer, named SONM, and identified that none of these solutions provide smart contract-based quality of service (QoS) and that privacy of consumer data is at stake since it is managed by their parties. The need for decentralized races was also identified.

Bouachir et al. [17] studied the challenges presented in a cyber-physical system useful for IoT and industrial Internet of Things (IIoT) and the usage of blockchains to overcome these challenges. They identify that the limited computational, communication, and storage resources of small devices are not naturally compatible with the blockchain infrastructure, which usually require high-compute-intensive machines. Also, blockchains are designed to use homogeneous nodes with equal capabilities and responsibilities, whereas the cyber-physical system environment has heterogeneous devices interconnected. Thus, the centralized network architecture is shifted to a distributed architecture with fog computing to overcome the resource limitations and heterogeneity challenges.

Wu et al. [83] proposed a strategy to integrate blockchains with fog computing by partitioning the fog server nodes into clusters such that every cluster has an associated access control list (ACL) stored on a customized compute-efficient blockchain to monitor and restrict access to resources between clusters. Figure 3.1 shows the evolution of blockchain in fog computing, which shows how the blockchain

**Fig. 3.1** Evolution timeline of blockchain in fog computing



technology started in 1979 and Artificial Intelligence (AI) trends came recently in blockchain for Big Data analytics purpose for accurate and better predictions on the data that are stored into the blockchains.

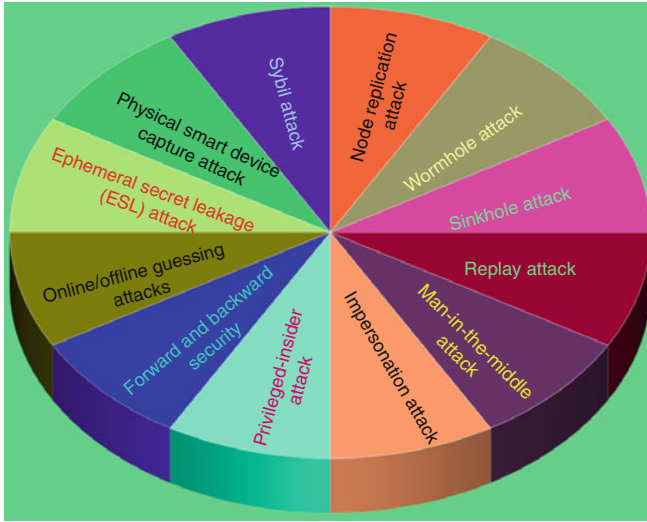
### 3.4 Security and Functionality Requirements in Fog Computing

The security requirements in fog computing environment are given below [64]:

- *Confidentiality*: It must be ensured that the data flowing in the network is understood by intended recipients only.
- *Data integrity*: Any message from an authorized sender to an intended recipient must not be altered during the transit.
- *Authentication*: It is required to validate a communicating node who it claims to be. All parties involved in fog computing environment, such as user, IoT smart device, fog node, and cloud server, must establish bi-directional trust through mutual authentication before granting access to restricted resources or any sensitive information.
- *Authorization*: In fog computing, where access control mechanisms are employed, it is required to authorize an authenticated entity to check if he/she has required privileges to access the requested resource. This, unauthorized access leads to an under-privileged user accessing an elevated resource.
- *Availability*: This requirement ensures that the services of fog computing are always available and must not be hampered by internal/external attacks or by resource starvation due to complex operations. In other words, denial-of-service (DoS) attacks must be prevented.
- *Data freshness*: Freshness is a very serious security feature in fog computing to ensure that the received data is freshly generated by the authentic participant and is not a replay message by an adversary.
- *Anonymity and privacy*: Identities of the entities must not be exposed to any eavesdropping adversaries.
- *Non-repudiability*: Every session must be uniquely associated with a valid communicating entity such that in case of misuse, the guilty can be held responsible for his/her actions.

Apart from the above security requirements, the following security properties should be fulfilled:

- *Forward secrecy*: If a node or an entity leaves the network, it must be blocked from reading any communication flowing in the network after its departure.
- *Backward secrecy*: If a new node (entity) joins the network, it must also be blocked from reading/decrypting the communication that is flowed before its introduction.



**Fig. 3.2** Various possible potential attacks in fog computing environment

Additionally, the following attacks must be prevented in fog computing environment (see Fig. 3.2):

- *Node replication attack*: The attacker can deploy a malicious node that can simulate the identity and working of an existing node. The malicious node may generate fake messages in the network causing the other nodes to receive multiple conflicting messages.
- *Wormhole attack*: The adversary directs the messages between two nodes in the network such that these messages are tunneled through a set of nodes that are under the attacker's control. It forces the end nodes to misinterpret the route as the more efficient route by deceiving the end nodes into construing their distance between them as minimal. This allows for the network traffic to be shaped according to the attacker's needs. Such an attack can make provision for other attacks on the network traffic such as sniffing, modification, and dropping.
- *Sinkhole attack*: In this attack, an attacker compromises a node and modifies all routes to be directed through it so that all the traffic can be captured. This is done by publishing a less hop distance to misguide the neighbor nodes. Once the malicious node receives the traffic, it can misuse the re-directed traffic to eavesdrop, capture, modify, delete, or add messages to the traffic.
- *Replay attack*: The attacker monitors traffic between two communicating entities and copies certain message packets from sender to the receiver. These copied packets can then be sent to the receiver node multiple times to obtain undue advantage in terms of financial gain.
- *Man-in-the-middle attack*: This is a very specific attack in which the adversary first captures and blocks messages from the message sender to the message

receiver. Then the attacker creates counterfeit messages to be sent to the receiver. Similar action is repeated during the response from the receiver to the sender. This results on the counterfeit messages to be exchanged between sender and receiver instead of the real messages and allows the attacker to manipulate the two parties into thinking that they have exchanged the data with each other when in reality they have exchanged the data with the adversary. The two parties may not even be aware of the existence of the adversary.

- *Impersonation attack*: In this attack, the adversary illegitimately obtains the credentials of a legitimate entity. These credentials are then used by the attacker to communicate with other entities, misleading them into thinking that they communicate with the real entity.
- *Privileged-insider attack*: This attack is different from other attacks in that the adversary is not an outsider, but a legitimate user who has misuses his/her access privileges to obtain illegal information.
- *Online/offline guessing attacks*: Offline guessing attack refers to the act of speculating the correct login credentials of an entity. Online guessing attack is similar except that the adversary also attempts to login to the server. Offline guessing attack is considered to be more dangerous as the only limitation is the speed of the computer that is used to crack the password, whereas the speed of the network is an additional limiting factor in an online password guessing attack.
- *Ephemeral secret leakage (ESL) attack*: Any secret that is produced during the key establishment phase is called an ephemeral secret. Such secrets lead usually to play an important part in formulating the secret key and hence can present a major vulnerability in leakage of secret key information. ESL attacks are directed toward extraction of such ephemeral secrets used in the key agreement/establishment process.
- *Physical smart device capture attack*: This attack is possible in small-sized devices that may be mobile or immobile. The adversary seizes a device and extracts information from its memory. Such devices usually store secret information in their memory. If the device memory is insecure, this attack may lead to loss of a lot of secret information. Recovery from this attack involves replacement of such a device that may affect the cost involved.
- *Sybil attack*: In this attack, a malicious node maintains multiple active pseudonymous identities to itself in the network. Other nodes identify each of the identities to be unique nodes.

The following are the functionality requirements that are needed in fog computing deployment:

- A designed security protocol must be efficient in terms of storage, computation, and communication.
- Various entities registration/enrollment process should be executed in offline mode by the registration authority in order to reduce huge communication and computational overheads as the registration process is typically one-time procedure.

- The designed security protocol must support dynamic addition of entities in fog computing environment because some resource-constrained devices, such as IoT smart devices deployed in the network, may be physically captured by an adversary or they may be drained out of their battery power.
- A legal registered user must be permitted to change his/her password locally without contacting the registration authority in the designed security protocol.
- The designed security protocol must be scalable for supporting a huge number of nodes in a target network.

### 3.5 Taxonomy of Security Protocols in Fog Computing

Design of security protocols using the blockchain technology for communication in fog computing may fall into one or more of the categories as shown in Fig. 3.3.

#### 3.5.1 Authentication

Authentication of an entity verifies the identity of that entity by comparing the given credentials associated with the entity with the existing credentials that are allowed to access the system. The entity under consideration may be a device, a host, or a user. Authentication of a message ensures that the origin of the message is the intended source entity. On the other side, authentication of an entity may be single factor or multi-factor. Single-factor authentication uses one set of credentials, whereas multi-factor authentication uses multiple sets of credentials to verify an identity. In general, up to three-factor authentication is commonly used. The classification of authentication is provided below:

- *User authentication:* Under this category, a user is typically registered with a trusted registration authority (RA) and obtains the secret credentials from the

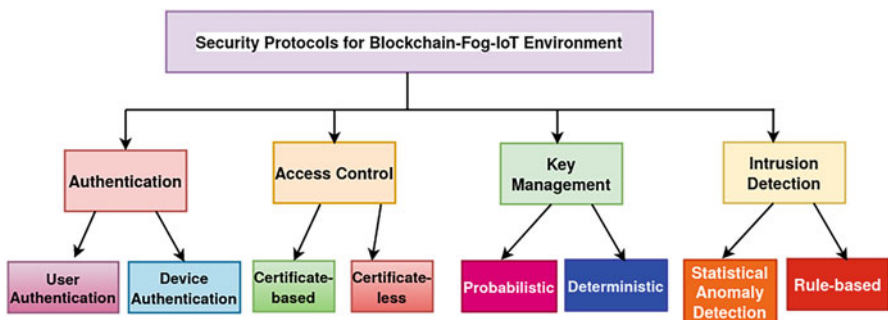


Fig. 3.3 Taxonomy of security protocols in fog computing environment

*RA* that are stored in a smart card or a mobile device. Later using the registration credentials stored in the smart card or mobile device, a user authentication with an accessed entity in fog computing environment and after successful mutual authentication, they establish a session key that is further used in secure communications [21, 30, 50, 66, 73, 77, 78].

- *Device authentication*: In device authentication, after registration with the *RA*, two devices need to mutually authenticate each other prior to establishment of session (secret) key using their pre-loaded registration credentials [81]. Next, using the establishment secret key, they can securely communicate each other for accessing the services in fog computing environment.

### 3.5.2 Access Control

Access control is the process of defining the operations that are allowed by an authorized entity in a given system and verifying that the entity is performing only the allowed operations on the system. Such access control schemes may use a certificate or they be also certificate-less. The access control mechanism primarily comprises the following two tasks [24, 25, 40, 45]:

- *Node authentication*: The newly deployed node must authenticate itself to the neighbor nodes in order to prove that it is a legal registered node and can access the network.
- *Key establishment*: It is essential for the newly deployed node in order to establish secret keys with the neighbor nodes to assure secure communication while transmitting the data only after mutual authentication.

### 3.5.3 Key Management

Two or more entities that wish to communicate securely with each other in such a way that the exchanged data is not visible to another external party must encrypt the data with a secret key common to all the entities involved in the communication. Such a key is to be agreed upon by all the involved entities and distributed securely among them [26–29]. This key also needs to be protected against compromise from different attacks. If it is compromised, the copy of the key at every entity must be replaced with a new agreed key. There are two types of key management that are possible:

- *Probabilistic key management*: Let the probability of establishing a secret key shared between any two neighbor nodes in the network be denoted by  $p_{key}$ . If  $0 < p_{key} < 1$ , a key management scheme is said to be probabilistic or randomized key management scheme.

- *Deterministic key management*: If  $p_{key} = 1$ , a key management scheme is termed as a deterministic key management scheme.

A node in fog computing environment (for example, an IoT smart device) can be physically captured by an adversary. By compromising the secret credentials stored in the compromised nodes, the attacker may be able to decrypt secure communication among other two non-compromised nodes in the network. Let  $P_e(n_c)$  denote the fraction of secure communication links that are compromised when  $n_c$  nodes are already compromised in the network excluding the communication links that are directly involved due to compromise of  $n_c$  nodes. If  $p_e(n_c) = 0$ , we say a key management scheme is *unconditional secure* or *perfectly resilience* against physical node capture attack.

### 3.5.4 Intrusion Detection

Intrusion detection system (IDS) is a regular monitoring system that can be either a hardware device or a software to identify any activity that can be considered as malicious according to pre-defined rules or policies. The techniques to detect intrusion in a system can be statistical- or anomaly-based and rule- or signature-based [55, 56, 75, 76, 79]. In statistical techniques, behavior of the system under normal circumstance is defined and stored in the IDS. This is done by collecting relevant data of regular users who are allowed by the system as legitimate. While the system is monitored, its behavior is analyzed against the stored conditions to categorize the current condition as normal or abnormal, if it falls outside the scope for the defined behavior for normal working of the system. In the rule-based techniques, the behavior of the system under potential attack is defined. While the system or network is under surveillance, any activity that concurs with the attack pattern is categorized to be an intrusion.

## 3.6 System Models

In this section, we elaborate the network and threat models related to blockchain-enabled fog computing environment.

### 3.6.1 Network Model

The network model exhibited in Fig. 3.4 consists of three layers: (1) terminal layer, (2) fog layer, and (3) cloud layer. These layers have the following functionalities and characteristics:



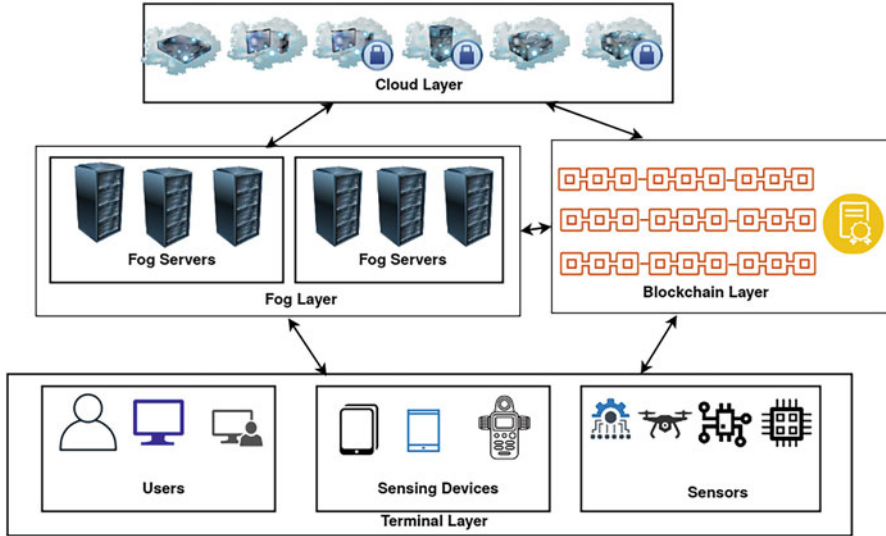


Fig. 3.4 Network model for a blockchain-enabled fog computing environment

- The *terminal layer* is the layer that consists of the end users, sensors, and other sensing and actuator devices that have the capability to sense the environmental readings of various physical parameters and send these parameters to the nearest fog server in the fog layer. It may also consist of actuator devices that can receive signals to perform a certain action that can affect the physical parameters.
- The *fog layer* consists of fog servers that are placed in groups nearer to the location of the actual sensor devices environment. These fog servers receive the sensor data consisting of readings of physical parameters and can perform certain pre-processing operations on this data. The blockchain is accessible to the terminal layer, fog layer, and also the cloud layer. Meta-data about this pre-processed data may be stored on the blockchain with the help of pre-defined operations in smart contracts. Once pre-processing is completed, the sensor data is stored at the fog servers until sufficient data is collected to be passed onto the cloud layer.
- The *cloud layer* consists of an assemblage of different types of cloud centers, such as private cloud center, public cloud center, and hybrid cloud center, which consist of high-end servers with access to private, public, and hybrid blockchains in the blockchain center. Depending on the application at hand, either private or public or hybrid blockchain may be used. The servers in the cloud layer have the capability to completely process the sensor data. Based on this processing, the smart contract may trigger a particular action to be performed at the terminal layer so the physical parameters can be controlled as required. This data may also be further forwarded to an *Artificial Intelligence (AI) and Big data center* that has the capability to utilize the data for further analysis and apply prediction

techniques to forecast any consequences of current actions at the terminal layer. Such consequences may trigger a new cycle of terminal to fog layer and fog to cloud layer data exchange and processing. Thus, the network model presents a system that is continuously active for the purpose of keeping the environmental surroundings in required limits.

### 3.6.2 Threat Model

The proposed network model is designed to be resistant against attacks defined in the threat models of Dolev–Yao model (DY model) [33] and Canetti–Krawczyk adversary model (CK model) [19].

- The DY model considers an adversary with the capability to monitor the messages communicated between two parties in order to modify or delete the messages. It also allows the adversary to add malicious content into the message on transit that may lead to misinterpretation of messages between the two parties.
- An adversary in the CK adversary model is similar to an adversary in the DY model. Apart from that, it can extract the information regarding the secret credentials, secret keys, and also any information about the states of the current session if all this information is stored insecurely in the memory of the communicating entities by launching session hijacking attack.

In addition, the end-point entities involved in communication are not in general trustworthy. The adversary may launch power analysis attack [49] or timing attack [32] to extract sensitive information from the physical captured devices' insecure memory and use the extracted data to impersonate the compromised entity. The cloud and fog servers can typically be treated as semi-trusted entities in the network. Finally, the registration authority involved during the registration process is considered as a fully trusted entity in the network.

## 3.7 Security Solutions for Blockchain-Based Fog Computing Environment

The user authentication system proposed by Almadhoun et al. [5] uses smart contracts to map fog nodes with IoT devices [47]. The access control permissions of the users and their permitted operations are handled by the administrator. The end users access the devices with their unique Ethereum addresses indirectly via smart contract or directly through application. Their scheme supports confidentiality, integrity, and non-repudiation, and it is also resilient against denial-of-service (DoS) attacks [62]. This scheme is a traditional authentication scheme where the entity is to be declared legitimate prior to any exchange of communication data. Once

declared so, the entity is trusted to be authentic. This may lead to vulnerabilities toward attacks aiming on active sessions.

Al-Naji and Zagrouba [41] builds upon the scheme by Almadhoun et al. [5] by adding continuous authentication, in which the entities involved in a session are continuously authenticated for the duration of an active session. This scheme is developed as a user-to-device model for mutual authentication between an end user and fog nodes with a smart contract issuing the access to avoid the involvement of trusted third party. A machine learning model for face recognition is used at the fog layer that is continuously updated according to the data collected by the IoT smart devices. The smart contract applies face similarity score and a similarity threshold on the face recognition model to obtain the trust model based on the comparison results, which then yields the access decision model with the decision to continue or lockout that is fed back into the fog layer.

Wang et al. [72] proposed a mutual authentication scheme between an end user  $EU_i$  and an edge server  $ES_j$  with the key materials table  $KMST$  deployed on a smart contract over a blockchain system based on Ethereum or Hyperledger fabric. The deployed smart contract contains algorithms to perform initialize, update, query, and revoke on the  $KMST$ . A trusted registration authority ( $RA$ ) registers the end user  $EU_i$  and the edge server  $ES_j$  via separate private and secure communication channels between them. When an  $EU$  decides to join the network, it sends a request with  $ID_i$  to  $RA$ . The  $RA$  chooses its own private scalar  $r_i \in Z_q^*$  and multiplies the base point  $P$ ,  $r_i$  times, to obtain a point on a non-singular elliptic curve as  $R_i = r_i \cdot P$ , where  $k \cdot P = P + P + \dots + P$  ( $k$  times) denotes the elliptic curve point (scalar) multiplication [46],  $q$  is a large prime such that elliptic curve discrete logarithm problem (ECDLP) becomes intractable,  $Z_q = \{0, 1, \dots, q - 1\}$ , and  $Z_q^* = \{1, \dots, q - 1\}$ .  $RA$  computes another private scalar  $x_i \in Z_q^*$  for  $EU_i$  using its own private scalar  $r_i$  and its own master key  $s$ , and also computes the corresponding public key for  $EU_i$  as  $PK_i$ .  $RA$  generates  $PID_i$  as the hash of  $EU_i$ 's public key  $PK_i$  and encrypts  $EU_i$ 's identity  $ID_i$ .  $x_i$  is stored securely at  $EU_i$ . Similar procedure is applied at  $ES_j$  to obtain, verify, and store  $x_j$ .

The authentication process in Wang et al.'s scheme [72] between  $EU_i$  and  $ES_j$  starts with  $EU_i$  generating a private scalar  $a \in Z_q^*$  and computing the corresponding elliptic curve point  $A = a \cdot P$ . It then computes parameter  $pid_i$  as the bitwise exclusive-OR (XOR) of  $PK_i$  and the hash of  $A$  concatenated with the point  $a \cdot PK_j$ . It also computes the parameter  $k$  as the sum of  $a$  and the product of  $x_i$  and the hash of  $PK_i$ ,  $pid_i$ ,  $A$  and the timestamp  $t_i$ .  $EU_i$  sends the parameters  $A$ ,  $pid_i$ ,  $k$ , and  $t_i$  to  $ES_j$ .  $ES_j$  verifies timestamp  $t_i$ , extracts  $PK_i$  from  $pid_i$ , and hashes it to obtain  $PID_i$  that is sent as an argument to query the  $KMST$  and check the validity. If the result is true,  $ET_i$  has not expired and  $ES_j$  verifies the parameter  $k$  using  $A$  and  $PK_i$  extracted above.  $ES_j$  then computes the parameters  $K_1$  as the sum of the points obtained from point multiplication of  $x_j$ ,  $A$  and  $b$ ,  $PK_i$ , where  $b$  is a private scalar chosen by  $ES_j$  and its equivalent ECC point computed as  $B$ . The other parameter  $K_2$  that is the last component of the session key is obtained as the product of private scalar  $b$  and the point  $A$  received from  $EU_i$ . The point  $B$ ,

the session key verifier  $w$ , and timestamp  $t_j$  are sent to the  $EU_i$ .  $EU_i$  computes the session key similar to  $ES_j$  and verifies it using the session key verifier.

Pallavi and Kumar [53] proposed an authentication scheme based on smart contracts that allows the data owners to verify the entities requesting for the data without the involvement of any third party. The proposed scheme uses a system model consisting of an administrator, end users, fog nodes, IoT smart devices, and cloud. Administrator registers fog nodes and IoT smart devices and handles access control through attribute permission using smart contracts. The end users are the requesters of the data from specific IoT smart devices. Fog nodes provide some storage and computation ability to reduce the processing and storage latency at the IoT devices. The cloud consists of the complete collection of data that can be used for heavy processing. A smart contract consists of a mapping of which IoT smart devices send their data to specific fog nodes and also a mapping of which end users are allowed to access which IoT smart devices. To register a device, the admin creates a smart contract that generates a device password, based on the device ID, and Ethereum address also uploaded to the IoT smart device. The device password is also recomputed and stored at the smart device. To access the device, an end user needs to specify this device password correctly. To map the fog nodes and IoT smart devices, admin creates a message using device Ethereum address and another message using Ethereum address of fog node. The pair of fog node and the device to be mapped are then stored in both fog nodes and IoT smart devices. Similarly, another message is created from user ID and password that is stored in both end user and smart device to map them together. The authentication phase authenticates the end user to the admin by sending a request to access a specific device with its Ethereum address. The request may be rejected by the admin if the user is not authorized to access the requested device. The token generation phase uses the hash of timestamp, device Ethereum address, and public key to create: (a) an access user token with the device Ethereum address and identity and (b) a user token with the user identity and Ethereum address. As a response to the authentication request sent in the authentication phase, the admin sends an access token to the user followed by the creation of user token by the user. The tokens are verified by the smart contract by computing a message that encrypts the product of user password XORed with the timestamp and the hash of device identity concatenated with the random number used in the first message of device–fog mapping. After the end user enters the user password for access, the verification message is also computed at the end user. If the verification messages generated at the smart contract and the end user are identical, then the user token is sent to the fog node and an access token is sent to the end user. The received tokens are verified at both ends. From the received user token, the fog node computes the user private key and the session key as the encryption of the device identity concatenated with device password. The fog nodes send both the private key and the session key to the end user. The end user digitally signs its user token with the received private key and sends to the fog node. The fog node then verifies the digital signature on the user token and generates the first signed message with encryption of user token and user private key concatenated, and a second signed message as the hash of the Chebyshev polynomial [37] XORed with

the user token. The end user sends the two signed messages to the fog node. The fog node re-computes the two signed messages and verifies if the received signed messages match the computed messages. If so, the end user is granted access to the IoT smart device. In the data exchange phase, the end user and IoT device can directly communicate over an established secure sockets layer (SSL) connection. The fog node and the end user generate a parameter by encrypting the concatenation of the first signed message and the session key and adding it to the user private key. If this computed parameter matches both the fog node and the end user, the data exchange can be successfully initiated. The drawback of this scheme is that the fog nodes send both the private key and the session key to the end user. This scheme requires the channel to be a private secure channel (via SSL) between the fog nodes and the device.

Abdalah et al. [1] proposed a system where a controller registers and manages the gateway fog nodes and registers the IoT smart devices. Each controller manages one gateway fog node, and each fog node manages multiple IoT smart devices. The cloud server is a centralized system that has the capability to register the devices, create users, deploy smart contract, and register the controller to the blockchain. The cloud server first registers the devices by executing the add device function in the smart contract and issues a private key to the device. The device itself generates the corresponding public key from the private key received. The device now registers with the gateway fog node by sending a request with its public key and receives the gateway public key in response. To prevent replay attack, the device sends its identity, its device information in JavaScript Object Notation for Linked Data (JSON-LD) format [67], a nonce, and a timestamp encrypted with the gateway's public key to the gateway node. After verifying the timestamp, the gateway forwards this request to the controller that ensures that the device exists of the blockchain and is registered by the user before adding the device identity and the device information onto the blockchain. Once it is done, the controller sends the device identity and a new nonce encrypted with the gateway public key to the gateway. If the nonce is valid, the gateway replies to the device with the device identity and user-device key. The device is now registered. The gateway sends its identity and public key to the controller, which invokes the smart contract to add the gateway and responds with its own public key if the gateway registration is successful. During device authentication process, the device sends its identity and request nonce to the gateway, which invokes the get device function in smart contract to extract the device information, if it is legitimate. The gateway responds to the device with the received request nonce and adds a new response nonce encrypted with the user-device key. The device decrypts the response nonce with the user-device key and sends this nonce to the gateway encrypted with the gateway's public key. If this is verified correctly, the gateway responds with the request nonce and a timestamp to declare that the device is now authenticated to communicate with the device.

Patonico et al. [54] proposed an authentication scheme designed to provide anonymity and data integrity along with the generation of a secret session key by computing separate parameters for each of these security functionalities. Each device is pre-loaded with an identity, a certificate, a pair of public and private keys,

and the public key of the cloud server. The sum of a random variable and the private key of the computing entity is generated. This sum is multiplied with the base point of an elliptic curve to obtain the parameters for session key generation. The sum is further multiplied with the public key of the cloud server to obtain a symmetric key. The first anonymity parameter is generated by encrypting the entity identity and certificate using the symmetric key. A second anonymity parameter is generated by multiplying the sum with public key of the entity. The data integrity parameter is generated by hashing the concatenation of session key parameter, anonymity parameters, and data integrity parameter. All these parameters are sent to the fog device, and the sensor device is said to be initialized at this stage. The fog server follows a similar procedure to obtain its own session key parameters, anonymity parameters, and data integrity parameters with the first anonymity parameters generated using session key parameters and second anonymity parameters of both sensor node and fog node. All the parameters from fog node, the second sensor anonymity parameter, and the sensor session key parameter are all forwarded to the central server. The central server follows the same procedure as the fog node and generates an anonymity parameter for the sensor device that is passed to the sensor node via the fog node. The session key consists of the hash of the session key parameter between sensor-to-fog, fog-to-central server, and central server-to-sensor node.

Yang et al. [84] proposed a framework for access control with a cloud service provider, a data owner that uploads the data to the cloud, and the associated access rights for each resource to the blockchain and a data user that accesses resources from the cloud if verified for the access rights requested. The need for blockchain arises from the fact that the cloud is assumed to be only semi-trusted. When the data user requests for a resource from the cloud, it queries the blockchain for the access rights of the user for the requested resource. Depending on the result obtained from the blockchain, the final access permission is determined. For the smooth flow in this system, the cloud, the data owner, and the data user register with the blockchain in the initialization phase by sending a request message along with the start and end timestamps of the time period during which the data requested is to be synchronized. The blockchain generates the public and private keys for the cloud using a smart contract function. Using the public key of the cloud, the associated address for the cloud is determined. The symmetric key between the cloud and the blockchain is used to encrypt the cloud keys pair, and the cloud address with the private cloud key encrypted again with the symmetric key. The cloud uses the symmetric key to decrypt its address and key pair. The data user and data owner are also registered in a similar procedure. To publish the resource, it is uploaded to the cloud by the data owner, and the associated metadata returned by the cloud is uploaded to the blockchain using a smart contract function. To access a resource, the user sends the encrypted resource information and its own address to the cloud. The cloud decrypts the resource information and the user address and obtains the hashed resource information that is passed to a smart contract function on the blockchain that returns the appropriate result metadata information, which is passed on from the blockchain to the cloud. The cloud decrypts this result metadata information

and checks if the resulting metadata information in the actual data in the cloud has the same value of this result metadata returned from the blockchain. If it happens so, the cloud responds to the user to access the resource and updates the access log in the blockchain about this recent user access to the resource. Authorization of access to different users may be directly given to the data owner by allowing the blockchain to call the verification smart contract function or indirectly by a previous data user to other data users by allowing the data users to send an authorization notice proving that they are allowed to authorize other data users. This scheme uses the address of resources and users instead of usernames that gives improved performance. In addition, this scheme provides accountability, availability, authenticity, and integrity with multiple protection mechanisms.

Zhang [85] proposed a key management scheme, named as dynamic contributory broadcast encryption, that can be used to establish a secure channel among a group of fog nodes such that a common public key is generated for encryption and separate private keys for each fog node in the groups are generated for decryption, without the involvement of any third party. This allows any external end user to generate messages intended to be received by one of the fog nodes in the group and securely encrypt it with the group fog public key. Such a message can only be decrypted by the specific recipient fog node in the group. No other node in the group will be able to decrypt the correct message. This scheme allows any fog node to leave the group at any time and any node to join the group at any time. This scheme uses the bilinear pairing cryptographic primitive to generate a tuple for each group of fog nodes corresponding to the group size. A bilinear pairing is a mapping  $e: G_1 \times G_1 \rightarrow G_2$  with the following three properties [16, 48]. Here,  $G_1$  and  $G_2$  are the cyclic additive and multiplicative groups of a large prime order  $q$ , and  $G_2$  is called the target group.

- **Bilinearity:**  $e(P + Q, R) = e(P, R)e(Q, R)$  and  $e(P, Q + R) = e(P, Q)e(P, R)$ ,  $\forall P, Q, R \in G_1$ . In general, we have  $e(aP, bQ) = e(P, Q)^{ab}$ ,  $\forall a, b \in Z_q^* = \{1, 2, \dots, q - 1\}$ .
- **Non-degeneracy:** Let  $e_{G_1}$  be the identity in  $G_1$ . Then,  $e(P, P) \neq e_{G_1}$ ,  $\forall P \in G_1$ .
- **Computability:** There is an efficient polynomial-time algorithm to calculate  $e(P, Q)$ ,  $\forall P, Q \in G_1$ .

The group size dynamically determines based on the earlier groups and applications. Since the fog nodes can join and leave dynamically, the number of fog nodes may exceed the group size at some point that necessitates the need for creation of a new group. Every fog node is given a position in the system. A system position is set to 1 if occupied by a fog node. During initialization, every fog node computes its local parameter for itself and other fog nodes and publishes all these parameters. The rest of the fog nodes then computes the public encryption key from the received messages and derives its own private decryption key. When a new fog node is to join the system, it has to repeat the initialization process and set its position in the system to 1. When a fog node is to leave the group, it publishes its local parameters for itself and its group nodes to the entire group. The rest of the members then multiplies with the inverse of these parameters to nullify the existence of this fog node. To send a



message to multiple groups, an end user has to encrypt the same message using the public keys of the intended groups multiple times separately. This process has a large communication overhead, and to reduce this, a uniform session key for all the groups can be shared to every group encrypted with the group session key so that a broadcast message can be passed to all the fog nodes across all the groups at once. This trade-off reduces the complexity of communication to linear complexity.

Shabisha et al. [60] proposed an authentication system and key agreement system for a group of fog nodes where a fully trusted server registers and authorizes the devices and the fog nodes jointly perform the agreement of the key among themselves. The designed scheme, based on elliptic curve cryptography and Lagrange interpolation, considered several factors such as ensuring the privacy of the devices and keeping the connection among the nodes untrackable, with no necessity of pre-shared key variables. The scheme has been designed by considering two typologies: (a) static topology, where the devices have fixed locations and are statically assigned to a fixed fog node, and (b) dynamic topology, where the devices are assumed to be mobile leading dynamic mapping of fog nodes based on the changed location. The initialization phase ensures that the device's public session parameter is known to the fog node and the server with the device's identity hidden using hash and elliptic curve point multiplication with the server public key. The difference between the static and dynamic initialization is that the public device parameter is computed at the server for static initialization, whereas it is computed at the device node for dynamic initialization. The group authentication and key agreement phase runs in four stages: (a) request of update by fog node, (b) response by devices, (c) response by fog nodes, and (d) acknowledgment. In request of update stage, the fog node computes a signature on a public variable that is derived from a local private random variable and sends it to the server along with its fog identity. After the server verifies the signature, it computes its own signature from a local private random variable, two parameters for device and fog node, and a polynomial from the Lagrange interpolation. The hashed polynomial along with the server signature and fog signature are sent to the fog. The fog verifies the signature and forwards the message to the device. The device verifies the integrity of the message and the signature before storing the hashed polynomial. It encrypts the received parameters after hashing their concatenation and forwards them to the fog node. The fog node reconstructs the polynomial and extracts all the coefficients and forwards them to the device. Using this, the device derives the polynomial using the Lagrange interpolation and checks if it matches with the stored hashed polynomial. If it is so, the local private random variable is updated, and the hash of the polynomial, old private random value, and new private random value are taken and multiplied with the device private key and subtracted from the new private random value to obtain the difference as  $s$ . This difference  $s$  along with the old and new private random values are passed to the fog node. The fog node performs the same computation of difference, and after verification with the received difference, it updates the public random variable of device in its memory. The server performs the same verification and updation of device public random variable in its memory. For data exchange, the message may be sent in plaintext or encrypted with the symmetric key



between device and fog node. A timestamp is further generated. Three hashes of the message and timestamp, the symmetric key and timestamp, and the message, the symmetric key, and timestamp are generated, concatenated, and encrypted with the Lagrange polynomial as the key for non-encrypted communication. For encrypted communication, the ciphertext corresponding to the message is concatenated with the hash of the symmetric key and timestamp, along with the Lagrange polynomial and passed to the fog node. This scheme provides authentication of the entities, authenticity, integrity, anonymity, unlinkability, perfect forward secrecy, group forward secrecy, and backward confidentiality.

## 3.8 Comparative Analysis

In this section, we perform a detailed comparative study on the communication and computational costs and also security features among various state-of-the-art security protocols, such as the schemes designed by Almadhoun et al. [5], Al-Naji and Zagrouba [41], Wang et al. [72], Pallavi and Kumar [53], Abdalah et al. [1], Patonico et al. [54], Yang et al. [84], Zhang [85], and Shabisha et al. [60].

### 3.8.1 Comparative Analysis on Communication and Computational Costs

For comparative study on the communication and computational costs, we have computed the communication and computational costs for different schemes. Next, we have rearranged the schemes in descending order based on their communication and computational costs. If the computational/communication cost for a scheme is high/very high, we have marked it as **high**; if the computational/communication cost of a scheme is low, we have marked it as **low**; otherwise, if the computational/communication cost for a scheme is medium, we have then marked it as **medium**. Table 3.1 shows a comparative study on communication and computational costs for the existing schemes.

The studied security protocols have been compared on the basis of the number of operations required for expensive computations in the schemes and the amount of data to be transmitted as communication costs. The cost ranges for communication less than 3000 bits have been considered as low, and more than 4000 bits has been taken as high. With computation cost, the schemes that are based on bilinear pairings or involve many elliptic curve multiplication operations turn out to have very high computation costs.

**Table 3.1** Comparative study on communication and computational costs

Scheme	Communication cost	Computational cost
Almadhoun et al. [5]	Low	Medium
Al-Naji and Zagrouba [41]	Low	Medium
Wang et al. [72]	Medium	Medium
Pallavi and Kumar [53]	Low	Low
Abdalah et al. [1]	High	High
Patonico et al. [54]	High	High
Yang et al. [84]	Medium	High
Zhang [85]	High	High
Shabisha et al. [60]	Low	High

**Table 3.2** Comparative study on security features

Features	[5]	[41]	[72]	[53]	[1]	[54]	[84]	[85]	[60]
Confidentiality	✓	✓	✓	✓	✓	×	✓	✓	✓
Integrity	✓	✓	✓	✓	✓	✓	✓	✓	✓
Authenticity	✓	✓	✓	✓	✓	✓	✓	✓	✓
Non-repudiation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Anonymity	✓	✓	×	✓	✓	×	×	×	✓
Traceability or unlinkability	✓	✓	×	✓	✓	✓	✓	✓	✓
Mutual authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓
Key agreement	✓	✓	✓	✓	✓	✓	✓	✓	✓
Forward secrecy	×	×	×	×	×	×	✓	✓	✓
Backward secrecy	×	×	×	×	×	×	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Man-in-the-middle attack	✓	✓	✓	✓	✓	✓	✓	✓	✓
Privileged-insider attack	×	×	×	×	✓	✓	×	×	✓
ESL attack	×	✓	×	✓	✓	✓	×	✓	✓
Impersonation attack	×	✓	✓	✓	✓	✓	×	✓	✓
Physical node capture Attack	×	✓	✓	✓	✓	✓	×	✓	✓
DoS attack	✓	✓	×	×	✓	✓	×	✓	✓

Note: ✓: A scheme resists an attack or supports a feature; ×: a scheme does not resist an attack or it does not support a feature

### 3.8.2 Comparative Analysis on Security Features

The security features among the existing schemes are compared in Table 3.2. Several security features have been considered based on security requirements and threats in fog computing environment that are already discussed in Sect. 3.4. It is evident that the scheme [60] provides better security as compared to other existing schemes considered in Table 3.2.

### 3.9 Conclusion

In this chapter, we focused on studying fog computing in detail by analyzing its applications in various fields. The applied analysis allows to understand the need for security in fog computing. Once the security and functionality requirements of fog computing were identified, the evolution of the usage of blockchains to fulfill the security gaps in fog computing was studied. The literature was analyzed to understand the existing security schemes that apply blockchains in fog computing. Finally, we provided a detailed comparative analysis on the communication and computational costs and also security features among various state-of-the-art security protocols that are proposed in the line of blockchain-based fog computing environment.

### References

1. Abdalah, A. N., Mohamed, A., & Hefny, H.A. (2020). Proposed authentication protocol for IoT using blockchain and fog nodes. *International Journal of Advanced Computer Science and Applications*, 11(4)
2. Advanced Encryption Standard (AES). (2001). FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001. <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>. Accessed on February 2021.
3. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *International Conference on Engineering and Technology (ICET)* (pp. 1–6). <https://doi.org/10.1109/ICEngTechnol.2017.8308186>.
4. Ali, A., Ahmed, M., Imran, M., & Khattak, H.A. (2020). *Security and privacy issues in fog computing* (chap. 5, pp. 105–137). Hoboken: John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781119551713.ch5>.
5. Almadhoun, R., Kadadha, M., Alhemeiri, M., Alshehhi, M., & Salah, K. (2018). A user authentication scheme of IoT devices using blockchain-enabled fog nodes. In *15th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, Aqaba, Jordan (pp. 1–8).
6. Bagga, P., Sutrala, A. K., Das, A. K., & Vijayakumar, P. (2021). Blockchain-based batch authentication protocol for Internet of Vehicles. *Journal of Systems Architecture*, 113, 101877.
7. Banerjee, S., Bera, B., Das, A. K., Chattopadhyay, S., Khan, M.K., & Rodrigues, J. J. (2021). Private blockchain-envisioned multi-authority CP-ABE-based user access control scheme in IIoT. *Computer Communications*, 169, 99–113.
8. Baniata, H., & Kertesz, A. (2020). A survey on blockchain-fog integration approaches. *IEEE Access*, 8, 102657–102668. <https://doi.org/10.1109/ACCESS.2020.2999213>.
9. Baniata, H., Anaqreh, A., & Kertesz, A. (2021). PF-BTS: A privacy-aware fog-enhanced blockchain-assisted task scheduling. *Information Processing & Management*, 58(1), 102393. <https://doi.org/10.1016/j.ipm.2020.102393>.
10. Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending Bitcoin's proof of work via proof of stake. *SIGMETRICS Performance Evaluation Review*, 42(3), 34–37.
11. Bera, B., Chattaraj, D., & Das, A. K. (2020). Designing secure blockchain-based access control scheme in IoT-enabled Internet of Drones deployment. *Computer Communications*, 153, 229–249.
12. Bera, B., Das, A. K., Obaidat, M., Vijayakumar, P., Hsiao, K. F., & Park, Y.: AI-enabled blockchain-based access control for malicious attacks detection and mitigation in IoE. *IEEE Consumer Electronics Magazine*, 1–1 (2020). <https://doi.org/10.1109/MCE.2020.3040541>.

13. Bera, B., Das, A. K., & Sutrala, A. K. (2021). Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of Drones environment. *Computer Communications*, 166, 91–109.
14. Bera, B., Saha, S., Das, A. K., Kumar, N., Lorenz, P., & Alazab, M. (2020). Blockchain-envisioned secure data delivery and collection scheme for 5G-based IoT-enabled internet of drones environment. *IEEE Transactions on Vehicular Technology*, 69(8), 9097–9111.
15. Bera, B., Saha, S., Das, A. K., & Vasilakos, A. V. (2020). Designing blockchain-based access control protocol in IoT-enabled smart-grid system. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2020.3030308>.
16. Boneh, D. (2012). Pairing-based cryptography: Past, present, and future. In *International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'12)*, Beijing, China (pp. 1–1)
17. Bouachir, O., Aloqaily, M., Tseng, L., & Boukerche, A. (2020). Blockchain and fog computing for cyberphysical systems: The case of smart industry. *Computer*, 53(9), 36–45. <https://doi.org/10.1109/MC.2020.2996212>.
18. Butun, I., Sari, A., & Öhsterberg, P. (2020). Hardware security of fog end-devices for the Internet of Things. *Sensors*, 20(20). <https://doi.org/10.3390/s20205729>.
19. Canetti, R., & Krawczyk, H. (2002). Universally composable notions of key exchange and secure channels. In *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02)*, Amsterdam, The Netherlands (pp. 337–351).
20. Chaiyarak, S., Koednet, A., & Nilsook, P. (2020). Blockchain, IoT and fog computing for smart education management. *International Journal of Education and Information Technologies*, 14. <https://doi.org/10.46300/9109.2020.14.7>.
21. Challa, S., Das, A. K., Gope, P., Kumar, N., Wu, F., & Vasilakos, A. V. (2020). Design and analysis of authenticated key agreement scheme in cloud-assisted cyber-physical systems. *Future Generation Computer Systems*, 108, 1267–1286.
22. Chamola, V., Hassija, V., Gupta, V., & Guizani, M. (2020). A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact. *IEEE Access*, 8, 90225–90265.
23. Chattaraj, D., Saha, S., Bera, B., & Das, A. K. (2020). On the design of blockchain-based access control scheme for software defined networks. In *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada (pp. 237–242). <https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162669>.
24. Chatterjee, S., Das, A. K., & Sing., J. K. (2013). Analysis and formal security verification of access control schemes in wireless sensor networks: a critical survey. *Journal of Information Assurance and Security*, 8(1), 33–57.
25. Chatterjee, S., Das, A. K., & Sing., J. K. (2014). An enhanced access control scheme in wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 21(1–2), 121–149.
26. Das, A. K. (2008). An identity-based random key pre-distribution scheme for direct key establishment to prevent attacks in wireless sensor networks. *International Journal of Network Security*, 6(2), 134–144.
27. Das, A. K. (2008). An unconditionally secure location-aware key management scheme for static sensor networks. *Journal of Discrete Mathematical Sciences and Cryptography*, 11(3), 333–355.
28. Das, A. K. (2008). ECPKS: An improved location-aware key management scheme in static sensor networks. *International Journal of Network Security*, 7(3), 358–369.
29. Das, A. K. (2012). A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks. *International Journal of Information Security*, 11(3), 189–211.
30. Das, A. K., Sutrala, A. K., Kumari, S., Odelu, V., Wazid, M., & Li, X. (2016). An efficient multi-gateway-based three-factor user authentication and key agreement scheme in hierarchical wireless sensor networks. *Security and Communication Networks*, 9(13), 2070–2092.

31. Das, A. K., Wazid, M., Kumar, N., Khan, M. K., Choo, K. K. R., & Park, Y. (2018). Design of secure and lightweight authentication protocol for wearable devices environment. *IEEE Journal of Biomedical and Health Informatics*, 22(4), 1310–1322. <https://doi.org/10.1109/JBHI.2017.2753464>.
32. Dhem, J. F., Koene, F., Leroux, P. A., Mestre, P., Quisquater, J. J., & Willems, J. L. (1998). A practical implementation of the timing attack. In *International Conference on Smart Card Research and Advanced Applications* (pp. 167–182). Berlin: Springer.
33. Dolev, D., & Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2), 198–208.
34. Fernández-Caramés, T. M., & Fraga-Lamas, P. (2019). Towards next generation teaching, learning, and context-aware applications for higher education: A review on blockchain, IoT, fog and edge computing enabled smart campuses and universities. *Applied Sciences*, 9(21), 4479.
35. Fernandez-Carames, T. M., & Fraga-Lamas, P. (2019). Design of a fog computing, blockchain and IoT-based continuous glucose monitoring system for crowdsourcing mHealth. In *Proceedings of 5th International Electronic Conference on Sensors and Applications* (vol. 4). <https://doi.org/10.3390/ecsa-5-05757>.
36. Garg, N., Wazid, M., Das, A. K., Singh, D. P., Rodrigues, J. J. P. C., & Park, Y. (2020). BAKMP-IoMT: Design of blockchain enabled authenticated key management protocol for internet of medical things deployment. *IEEE Access*, 8, 95956–95977.
37. Gil, A., Segura, J., & Temme, N. M. (2007). *Numerical methods for special functions*. Philadelphia, USA: Society for Industrial and Applied Mathematics (SIAM). <https://epubs.siam.org/doi/abs/10.1137/1.9780898717822>.
38. Gul, M. J., Subramanian, B., Paul, A., & Kim, J. (2021). Blockchain for public health care in smart society. *Microprocessors and Microsystems*, 80, 103524.
39. Guo, R., Yang, G., Shi, H., Zhang, Y., & Zheng, D. (2021). O-R-CP-ABE: An efficient and revocable attribute-based encryption scheme in the cloud-assisted IoMT system. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2021.3055541>.
40. Huang, H. F. (2009). A novel access control protocol for secure sensor networks. *Computer Standards & Interfaces*, 31, 272–276.
41. Hussain Al-Naji, F., & Zagrouba, R. (2020). CAB-IoT: Continuous authentication architecture based on Blockchain for Internet of Things. *Journal of King Saud University – Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2020.11.023>.
42. Islam, N., Faheem, Y., Din, I. U., Talha, M., Guizani, M., & Khalil, M. (2019). A blockchain-based fog computing framework for activity recognition as an application to e-healthcare services. *Future Generation Computer Systems*, 100, 569–578.
43. Jangirala, S., Das, A. K., & Vasilakos, A. V. (2020). Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment. *IEEE Transactions on Industrial Informatics*, 16(11), 7081–7093.
44. Kaur, J., Agrawal, A., & Khan, R. A. (2020). Security issues in fog environment: A systematic literature review. *International Journal of Wireless Information Networks*, 27, 467–483.
45. Kim, H. S., & Lee, S. W. (2009). Enhanced novel access control protocol over wireless sensor networks. *IEEE Transactions on Consumer Electronics*, 55(2), 492–498.
46. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48, 203–209.
47. Kumari, A., Tanwar, S., Tyagi, S., & Kumar, N. (2018). Fog computing for Healthcare 4.0 environment: Opportunities and challenges. *Computers & Electrical Engineering*, 72, 1–13. <https://doi.org/10.1016/j.compeleceng.2018.08.015>.
48. Menezes, A. (2013). An introduction to pairing-based cryptography. <https://www.math.uwaterloo.ca/~ajmenez/publications/pairings.pdf>. Accessed on May 2020.
49. Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5), 541–552.
50. Mishra, D., Das, A. K., & Mukhopadhyay, S. (2016). A secure and efficient ECC-based user anonymity-preserving session initiation authentication protocol using smart card. *Peer-to-Peer Networking and Applications*, 9(1), 171–192.

51. Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: challenges. *IEEE Access*, 5, 19293–19304. <https://doi.org/10.1109/ACCESS.2017.2749422>.
52. Nikouei, S. Y., Xu, R., Nagothu, D., Chen, Y., Aved, & A., Blasch, E. (2018). Real-time index authentication for event-oriented surveillance video query using blockchain. In *2018 IEEE International Smart Cities Conference (ISC2)* (pp. 1–8). <https://doi.org/10.1109/ISC2.2018.8656668>.
53. Pallavi, K. N., & Kumar, V. R. (2020). Authentication-based access control and data exchanging mechanism of IoT devices in fog computing environment. *Wireless Personal Communications*, 1–22.
54. Patonico, S., Braeken, A., & Steenhaut, K. (2019). Identity-based and anonymous key agreement protocol for fog computing resistant in the Canetti–Krawczyk security model. *Wireless Networks*, 1–13.
55. Pundir, S., Wazid, M., Singh, D. P., Das, A. K., Rodrigues, J. P. C., & Park, Y. (2020). Designing efficient sinkhole attack detection mechanism in edge-based IoT deployment. *Sensors*, 20(5).
56. Pundir, S., Wazid, M., Singh, D. P., Das, A. K., Rodrigues, J. P. C., & Park, Y. (2020). Intrusion detection protocols in wireless sensor networks integrated to internet of things deployment: Survey and future challenges. *IEEE Access*, 8, 3343–3363.
57. Rivest, R. L., Shamir, A., & Adleman, L. (1983). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 26(1), 96–99.
58. Saha, S., Chattaraj, D., Bera, B., & Kumar Das, A. (2020). Consortium blockchain-enabled access control mechanism in edge computing based generic Internet of Things environment. *Transactions on Emerging Telecommunications Technologies*, e3995. <https://doi.org/10.1002/ett.3995>.
59. Saha, S., Sutrala, A. K., Das, A. K., Kumar, N., & Rodrigues, J. J. P. C. (2020). On the design of blockchain-based access control protocol for IoT-enabled healthcare applications. In *ICC 2020–2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland (pp. 1–6). <https://doi.org/10.1109/ICC40277.2020.9148915>.
60. Shabisha, P., Braeken, A., Kumar, P., & Steenhaut, K. (2019). Fog-orchestrated and server-controlled anonymous group authentication and key agreement. *IEEE Access*, 7, 150247–150261.
61. Singh, P., Nayyar, A., Kaur, A., & Ghosh, U. (2020). Blockchain and fog based architecture for internet of everything in smart cities. *Future Internet*, 12(4). <https://doi.org/10.3390/fi12040061>.
62. Singh, R., Tanwar, S., & Sharma, T. P. (2020). Utilization of blockchain for mitigating the distributed denial of service attacks. *Security and Privacy*, 3(3), e96. <https://doi.org/10.1002/spy2.96>.
63. Son, S., Lee, J., Kim, M., Yu, S., Das, A. K., & Park, Y. (2020). Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain. *IEEE Access*, 8, 192177–192191. <https://doi.org/10.1109/ACCESS.2020.3032680>.
64. Stallings, W. (2004). *Cryptography and network security: Principles and practices*, 3rd edn. India: Pearson Education.
65. Stojmenovic, I., Wen, S., Huang, X., & Luan, H. (2016). An overview of fog computing and its security issues. *Concurrency and Computation: Practice and Experience*, 28(10), 2991–3005. <https://doi.org/10.1002/cpe.3485>.
66. Sutrala, A. K., Obaidat, M. S., Saha, S., Das, A. K., Alazab, M., & Park, Y. (2021). Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial cyber-physical systems. *IEEE Transactions on Intelligent Transportation Systems*. <https://doi.org/10.1109/TITS.2021.3056704>.
67. Tutorial 3: Introduction to JSON-LD. (2017). <http://www.linkeddatatools.com/introduction-json-ld>. Accessed on February 2021.
68. Uriarte, R. B., & DeNicola, R. (2018). Blockchain-based decentralized cloud/fog solutions: Challenges, opportunities, and standards. *IEEE Communications Standards Magazine*, 2(3), 22–28. <https://doi.org/10.1109/MCOMSTD.2018.1800020>.

69. Vangala, A., Bera, B., Saha, S., Das, A. K., Kumar, N., & Park, Y. H. (2020). Blockchain-enabled certificate-based authentication for vehicle accident detection and notification in intelligent transportation systems. *IEEE Sensors Journal*. <https://doi.org/10.1109/JSEN.2020.3009382>.
70. Vangala, A., Das, A. K., Kumar, N., & Alazab, M. (2020). Smart secure sensing for IoT-based agriculture: Blockchain perspective. *IEEE Sensors Journal*. <https://doi.org/10.1109/JSEN.2020.3012294>.
71. Vangala, A., Sutrala, A. K., Das, A. K., & Jo, M. (2021). Smart contract-based blockchain-envisioned authentication scheme for smart farming. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2021.3050676>.
72. Wang, J., Wu, L., Choo, K. R., & He, D. (2020). Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. *IEEE Transactions on Industrial Informatics*, 16(3), 1984–1992. <https://doi.org/10.1109/TII.2019.2936278>.
73. Wazid, M., Bagga, P., Das, A. K., Shetty, S., Rodrigues, J. J. P. C., & Park, Y. (2019). AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment. *IEEE Internet of Things Journal*, 6(5), 8804–8817.
74. Wazid, M., Bera, B., Mitra, A., Das, A. K., & Ali, R. (2020). Private blockchain-envisioned security framework for AI-enabled IoT-based drone-aided healthcare services. In *Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond (DroneCom'20)*, London, United Kingdom (pp. 37–42).
75. Wazid, M., & Das, A. K. (2016). An efficient hybrid anomaly detection scheme using K-means clustering for wireless sensor networks. *Wireless Personal Communications*, 90(4), 1971–2000.
76. Wazid, M., & Das, A. K. (2017). A secure group-based blackhole node detection scheme for hierarchical wireless sensor networks. *Wireless Personal Communications*, 94(3), 1165–1191.
77. Wazid, M., Das, A. K., Khan, M. K., Al-Ghaiheb, A. A., Kumar, N., & Vasilakos, A. V. (2017). Secure authentication scheme for medicine anti-counterfeiting system in IoT environment. *IEEE Internet of Things Journal*, 4(5), 1634–1646.
78. Wazid, M., Das, A. K., Kumar, N., Vasilakos, A. V., & Rodrigues, J. J. P. C. (2019). Design and analysis of secure lightweight remote user authentication and key agreement scheme in internet of drones deployment. *IEEE Internet of Things Journal*, 6(2), 3572–3584.
79. Wazid, M., Das, A. K., Kumari, S., & Khan, M. K. (2016). Design of sinkhole node detection mechanism for hierarchical wireless sensor networks. *Security and Communication Networks*, 9(17), 4596–4614.
80. Wazid, M., Das, A. K., Shetty, S., & Jo, M. (2020). A tutorial and future research for building a blockchain-based secure communication scheme for internet of intelligent things. *IEEE Access*, 8, 88700–88716.
81. Wazid, M., Das, A. K., Shetty, S., Rodrigues, J. P. C., & Park, Y. (2019). LDKM-EIoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment. *Sensors*, 19(24). <https://doi.org/10.3390/s19245539>.
82. Wazid, M., Das, A. K., Shetty, S., & Rodrigues, J. J. P. C. (2020). On the design of secure communication framework for blockchain-based internet of intelligent battlefield things environment. In *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada (pp. 888–893). <https://doi.org/10.1109/INFOCOMWKSHP50562.2020.9163066>.
83. Wu, D., & Ansari, N. (2020). A cooperative computing strategy for blockchain-secured fog computing. *IEEE Internet of Things Journal*, 7(7), 6603–6609. <https://doi.org/10.1109/JIOT.2020.2974231>.
84. Yang, C., Tan, L., Shi, N., Xu, B., Cao, Y., & Yu, K. (2020). AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud. *IEEE Access*, 8, 70604–70615.
85. Zhang, L. (2019). Key management scheme for secure channel establishment in fog computing. *IEEE Transactions on Cloud Computing*. <https://doi.org/10.1109/TCC.2019.2903254>.



# Chapter 4

## Physical Layer Security Challenges and Solutions for Beyond 5G Fog Computing Networks



Alessandro Brighente, Mauro Conti, and Foroogh Mohammadnia

### 4.1 Introduction

**Physical Layer Security (PLS)** has attracted major attention in communication systems in the last decades [4]. The premise behind the emergence of **PLS** is to exploit the physical characteristics of the communication channel to increase the security via coding and signal processing techniques. The goal is to securely deliver a message from a transmitter to the intended receiver in the presence of malicious attackers and eavesdroppers [29]. While conventional cryptography-based security approaches can be exploited in upper layers, the openness of the wireless channel still presents a wide attack surface in largely distributed networks.

Conventional cryptographic solutions face multiple obstacles in providing secure communications in large networks. In fact, the computational complexity of key distribution of symmetric cryptographic methods or massive computational of asymmetric cryptography may result prohibitive. Moreover, with recent advancements in quantum computing and the consequent increase in computational capabilities, these solutions have become more vulnerable. Eavesdroppers and malicious users are hence supposed to have unlimited computational resource and network parameters awareness [26]. Guaranteeing security in **Fog Computing (FC)** is further complicated. In fact, considering the large amount of data generated by multiple devices, the rapid increase of the number of objects connected to the Internet, and the fact that most communications occur in wireless medium, security and reliability have become more critical. Furthermore, the quality of network service should not

---

A. Brighente (✉) · M. Conti

Department of Mathematics and HIT Research Center, University of Padova, Padova, Italy  
e-mail: [alessandro.brighente@unipd.it](mailto:alessandro.brighente@unipd.it)

F. Mohammadnia

Politecnico di Torino, Torino, Italy



be undermined by the complexity of key distribution schemes or cryptographic methods. PLS represents a suitable methodology to be applied at the physical layer to enhance the overall communication system security while maintaining low complexity and high scalability. These are among the major goals for beyond 5G networks [23].

In this chapter, we provide an overview of the FC paradigm and architecture, with a particular focus on its wireless links. We provide an overview of the related vulnerabilities and attacks and discuss how PLS techniques can be applied to guarantee the network security and confidentiality. We also discuss how physical layer technologies can guarantee security. We focus on the most recent advancements in the field of wireless communications related to the development of the beyond 5G and 6G communication networks. We then discuss how PLS represents a viable solution for FC security and how it can be exploited for secure-by-design network development.

## 4.2 Chapter Outline

The remainder of this chapter is organized as follows. In Sect. 4.3, we present the FC paradigm and architecture including the fundamentals of wireless communications in FC. Then in Sect. 4.4, we discuss the security and privacy issues and challenges in FC such as authentication discussion and trust concept. In Sect. 4.5, we present the basics of PLS compromising the physical layer authentication in different scenarios. After discussing the PLS basics, in Sect. 4.6, we discuss the technologies and solutions for PLS in FC, including non-orthogonal multiple access and massive MIMO. Ultimately, in Sect. 4.7, we conclude the discussion over PLS in FC.

## 4.3 Fog Computing Paradigm and Architecture

In this section, we first provide an overview of the FC paradigm and architecture. We then focus on the presence of wireless communication links and stress the distributed nature of FC.

### 4.3.1 Fog Computing Paradigm

The current uprising of the Internet of Things (IoT) paradigm creates a trend where more and more devices are connected to the network. In particular, IoT includes sensing devices that collectively generate a large amount of data. The previous generations of network architecture were based on cloud computing, i.e., the data generated by network edge devices were sent in the raw form to a

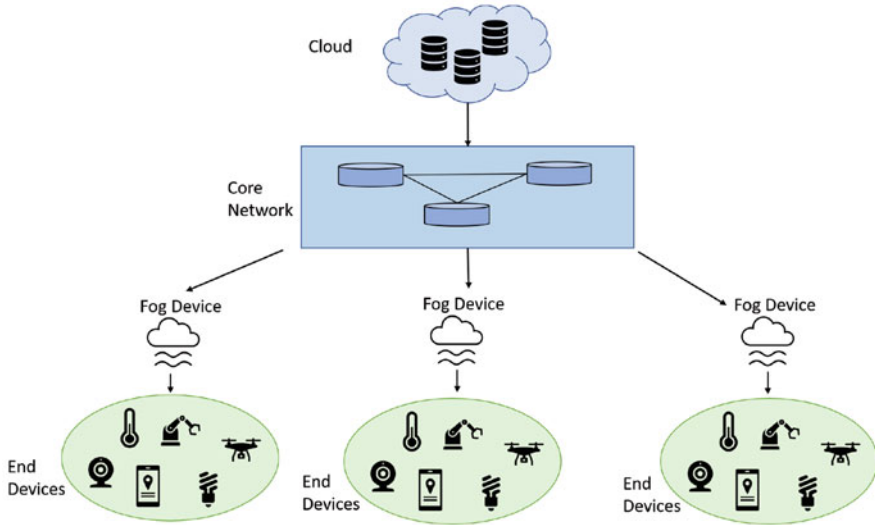
central cloud computing facility to be processed. However, the current market trend envisions the connection of 25.44 billion devices by 2030 [21]. Therefore, the cloud computing facilities will need to manage an enormous amount of data, risking to incur performance degradation due to the limited bandwidth.

To reduce the amount of data that needs to be delivered to the cloud, FC has been proposed as a novel network paradigm [5]. FC is a distributed computing paradigm where raw data generated at the network edge is pre-processed by dedicated edge devices before being passed to the cloud [13]. Therefore, thanks to the intermediate processing performed by fog devices, the processing needed by the cloud is drastically reduced. Users send requests and subscribe to fog services and are constantly updated on the service outcomes until the end of the subscription. Fog servers and devices perform the intermediate processing needed between the users' request and the service output. Therefore, thanks to the fog devices, the amount of data that needs to be delivered to the cloud is drastically reduced. This allows for reduced bandwidth requirements. A further advantage of FC is the reduction of the service latency. Considering that end devices are generally the party that needs access to the processed data, the end-to-end latency between the generation of the data and its use is reduced thanks to the proximity of fog devices to the generating devices. Therefore, compared to cloud computing, FC moves the computing power closer to the users, targeting the quality of service of the users and clients including network delay, reliability, throughput, and energy consumption.

### 4.3.2 Fog Computing Architecture

Figure 4.1 shows the architecture of a FC network. We can divide the whole network into four different layers: end layer, fog layer, core layer, and cloud layer. The end layer is represented by end devices that generate data. These devices may comprise IoT sensors, industrial devices, mobile devices, and drones. All generated data need to be processed before being used by the network or the devices themselves. The processed data are passed to the fog layer that includes the fog nodes. These nodes are responsible for data processing and for reducing the amount of data that will be delivered to the cloud. The fog layer is then connected to the core layer that is composed of the core network devices that enable the communication between the fog nodes and the cloud. The cloud layer is the upper layer, composed of the cloud infrastructure.

Fog devices can be implemented starting from a general device that has computing and storage capabilities, together with a network connectivity. Such devices include switches, routers, network controllers, servers, or video surveillance cameras [18]. The fog layer can be further divided into different layers according to the specific use case. In fact, multiple fog nodes can be connected and controlled via a fog server to act on a specific domain. Fog servers can be implemented starting from any generic server device. They are equipped with communications modules to connect to the cloud and retrieve information whenever the local processing requires



**Fig. 4.1** Architecture of a fog computing network

it. The **FC** paradigm does not specify any protocol or connection type (wired or wireless) among different deployed devices. Furthermore, it does not specify the protocols that need to be exploited. In this chapter, we focus on the scenarios where wireless technology is employed for the inter- and intra-layer communications.

### 4.3.3 *Wireless Communication in Fog Computing*

The fog computing architecture depicted in Fig. 4.1 includes several connections that can be implemented via wireless technology. This is particularly true in the edge layer, where sensing devices may be organized in vehicular or **IoT** networks. These devices are generally equipped with modules that enable their ubiquitous identification, sensing, actuating, and communication capabilities. These devices are deployed, based on their specific capabilities, in different domains including medical, agricultural, industrial, and smart home scenarios [18]. All these scenarios envision the deployment of wireless sensors connected among each other according to suitable topologies. Furthermore, they need to be wirelessly connected with the fog devices to report data and retrieve the service output. Wireless connectivity is also exploited to connect fog devices among each other and possibly with the core layer. In fact, mobility is a key feature both at the end layer and for computing and storage devices [18]. Therefore, wireless connectivity is a fundamental enabler also in the fog layer.

## 4.4 Security and Privacy Issues and Challenges in FC

In this section, we review the main security and privacy issues of the FC paradigm. In particular, we focus on the security and privacy aspects for which we envision PLS as a possible solution. We discuss the generic issue, without going into the details of the PLS solutions. We first provide an overview of the PLS threats in FC. We then discuss the legitimacy of communications, focusing on authentication and authorization. Afterward, we show how anomaly and intrusion detection can be exploited to provide a further security level. Last, we discuss the privacy issues in FC.

### 4.4.1 Physical Layer Threats in FC

In this section, we focus on the attacks targeting the physical layer. Physical layer attacks can be classified into two main groups: (i) active attacks and (ii) passive attacks. Passive attacks are represented by eavesdropping. Due to the absence of a protected communication medium, a malicious user can intercept the communication between two or more legitimate parties to obtain sensitive information. We discuss in Sect. 4.4.5 how this attack may affect the privacy and data protection of the FC network. In passive attacks, the attacker can act in a stealthy way and not be detected by the legitimate users or the network. Passive attacks are hard to detect since the attacker is not engaged in active communications.

Active attacks are easier to detect than passive ones, as the attacker actively affects the communication. Although easier to detect, active attacks still represent a serious threat as they are difficult to mitigate. Active attacks are either intended to modify the content of the messages or to undermine the availability of the network services. Regarding the first class of attacks, i.e., those jeopardizing the integrity of the communication, we can identify the pilot and feed back contamination. The attacker provides false physical layer signaling to the legitimate users so that basic operations such as synchronization or channel estimation obtain erroneous results. An attacker may also spoof the identity of a node, therefore impersonating a legitimate actor and obtaining access to data or controlling actions of the network. Regarding availability, an attacker may exploit different strategies. The first involves creating fake identities to perform an attack. In this case, an attacker may be able to send a large number of messages to the network exploiting false identities to lower the set of resources available for legitimate entities. A different strategy is jamming attack, where the attacker leverages interference to lower the channel quality of the legitimate user. Thanks to jamming, the attacker jeopardizes the network availability, making the channel inaccessible by legitimate nodes. Notice that thanks to the large number of wireless devices in FC, jamming has a strong impact. In fact, a single attacker may be able to undermine multiple communication links thanks to the wireless medium.

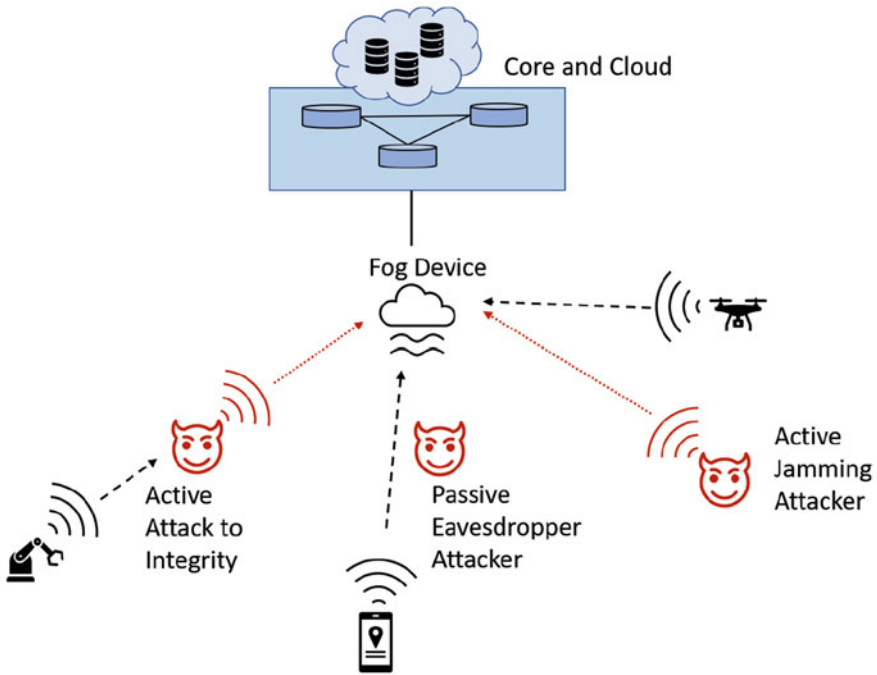


Fig. 4.2 Pictorial representation of the PLS and privacy threats in FC

Figure 4.2 shows a pictorial representation of the threats toward confidentiality, integrity, and availability due to physical layer attacks in a FC network. Notice that these types of attacks may also affect in case of wireless communications between fog nodes or from the fog node to the core IP network and cloud. In the next section, we will discuss possible generic approaches that may help in preventing the aforementioned attacks and highlight the challenges imposed by each of the proposed framework.

#### 4.4.2 Authentication in FC

Authentication of devices plays a fundamental role in guaranteeing the security of a FC network. Authentication provides a means to verify the identity and the legitimacy of a node delivering information. Nodes should authenticate to the fog network to join the network and be able to subscribe to services. The employed authentication protocols are supported by resource-constrained IoT devices. In fact, the use of public key cryptography may be hindered by the low storage and computing capabilities of IoT devices. To cope with this issue, authentication should

be delivered as a service in FC. In fact, although devices in the end layer may be resource-constrained, fog nodes and servers are equipped with sufficient storage and computing capabilities [17]. A further challenge is provided by the mobility of nodes. In fact, mobile IoT devices may frequently join and leave the network, behavior that might jeopardize the continuous availability of the service to end users. Therefore, registration and re-authentication will be delivered with low complexity to avoid huge impacts on the quality of service.

### ***4.4.3 Trust in FC***

Due to the existence of multiple mobile end devices generating different data types, a fundamental issue in FC is to understand which and to what degree end devices can be trusted. This issue arises both in the end layer communications and in the communication between the fog layers. Services' reliability highly depends on the trust and reliability of all the involved parties, in this case being end devices, fog nodes, and fog servers. Therefore, effective trust models should include the trust that each involved party poses to all the other parties. This issue is particularly present in FC due to its highly distributed nature. Authentication is a first step toward trust, as it provides, among the others, information on the identity of the involved party. However, authentication alone is not sufficient to provide trust. In fact, a node may be malfunctioning and, although authenticated, report incorrect information to the fog node or the end user. Therefore, trust includes the need for devices to validate the veracity of the received data [17]. The other application of trust is to verify whether the devices that are requesting a service are indeed genuine. However, there is no efficient mechanism that can be used to attain information on when and to which extent trust a node [2]. Therefore, the main scope is to identify metrics and attributes that can define trust in FC. Furthermore, due to the highly decentralized nature of FC, it is fundamental to identify the network components responsible for the trust model.

### ***4.4.4 Access Control, Intrusion, and Anomaly Detection***

The set of actions that can be performed by the network entities is under strict regulations in order to guarantee the network security. In particular, we need to ensure that only authorized entities have access to certain data or resources. Therefore, access control is a fundamental component of the overall security architecture. Access control faces additional challenges in fog computing compared to other more centralized architectures. In fact, the large number of end nodes and the huge amount of generated data represent a challenge in guaranteeing that only authorized users report or access data. Intrusion and anomaly detection could be helpful in guaranteeing access control in terms of regulation of the actors performing

certain actions. In fact, although an attacker may access the facility and perform some actions, intrusion and anomaly detection techniques should identify where the unauthorized user is and what action she has performed. However, anomaly detection techniques currently target a small subset of possible actions, making it difficult for the legitimate users to detect an attack not included in the set of those considered by the anomaly detection framework. Anomaly detection not only helps in identifying attacks, but also helps the network in taking decisions on which data can be considered as reliable and hence useful for further actions. In fact, malicious data can be reported also by authorized users due to either internal attacks or malfunctioning.

#### ***4.4.5 Privacy Issues and Data Protection***

Sensing and actuating devices measure and report a huge amount of data that a malicious user could exploit to infer information about end users. These data include usage and location information, both of which represent sensitive data. Several research contributions showed how the usage of such data can be exploited to infer different types of information. In fact, based on the energy consumption, a malicious user can infer the types of activities of the end user or profile users based on their specific physical channel features [6, 10]. Using a similar approach, a malicious user can infer information regarding the habits of the owner of a certain house, including how many people live there, when they are at home, or consume specific information [14]. A large number of IoT services are location-based, therefore providing information that can be exploited for users tracing purposes [17]. As previously discussed, IoT devices are resource-constrained and cannot therefore employ sufficiently strong privacy-preserving cryptographic techniques. The privacy challenge in FC is further complicated by the proximity of fog nodes to data generating devices and, therefore, to the end users. Moreover, due to the lack of a centralized control in highly distributed FC networks, a weak fog node can be used as an entry point for an attacker to steal users' private data. Location privacy is one of the FC-specific issues. In fact, end layer devices share data with the fog devices, and the location of the device can be linked to that of the owner, together with trajectory and mobility habits [12].

The privacy and data protection challenge in FC is further complicated by the increased number of devices compared to, e.g., the cloud computing framework. In fact, data will pass through an additional layer (fog devices) before being delivered to the cloud. Furthermore, the large number of deployed wireless connections makes the network vulnerable to eavesdropping attacks due to the absence of a closed communication medium. On the other hand, thanks to pre-processing at the edge nodes, suitable mechanism can be designed to minimize the amount of sensitive data flowing from the devices to the cloud.

## 4.5 Principles of Physical Layer Security

In this section, we provide an overview of the basic concepts of **PLS**. This section will serve as a reference theoretical point for further design of **PLS** solutions. We first describe how **PLS** can be exploited to guarantee the communication secrecy. We then discuss how physical layer features can be exploited for authentication.

### 4.5.1 Physical Layer Security Basics

As previously mentioned, **PLS** schemes exploit the channel state information (CSI) and transmitting precoding schemes to secure communications. Additionally, it is also possible to use a secret key to do the encryption. Therefore, the information-theoretic security will be the toughest form of security. **PLS** techniques are independent from upper layers security techniques, so **PLS** is used to increase the existing security. In fact, adding **PLS** to the network creates a multilayered security approach to augment the security of wired and wireless networks [16].

The primary work on **PLS** was presented by Shannon [1, 20]. Shannon proposed the foundation of cryptography theory according to information-theoretic methods. In [28], the hypothesis of a wiretap channel was introduced. The simple representation of a wiretap channel includes three terminals: a transmitter conventionally called Alice, a predestinate receiver Bob, and an eavesdropper. The purpose of the **PLS** in this context is that Alice sends a secure message to Bob preventing the eavesdropper from decoding any information from the forwarded signal.

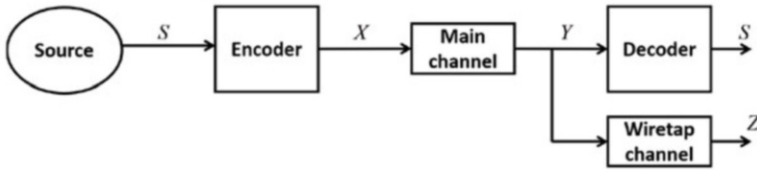
In the theory of wiretap channel, the most applicable metric is the secrecy capacity (SC). The secrecy capacity is the maximum rate achieved without letting the eavesdropper extracting any information from the transmission. Specifically, assuming the white Gaussian noise in the channel, the secrecy capacity of the channel can be computed as

$$C_s = [\log_2(1 + \gamma_M) - \log_2(1 + \gamma_E)]^+, \quad (4.1)$$

where  $[x]^* = \max(x, 0)$ , and  $\gamma_M$  and  $\gamma_E$  denote the signal-to-noise ratio (SNR) of the principal channel and the malicious user, respectively [1]. As interpreted from (4.1), the SC is the subtraction of the capacity of the link between Alice–Bob and capacity of the Alice–Eve link. SC plays a fundamental role in physical layer security. Simply speaking, the secrecy capacity determines the main limits of secure communications in noisy channels. The SC is innately associated with wiretap channel as a broadcast channel where there is at least one eavesdropper that should not receive information shared over the channel.

In the traditional approach of security, a secret key was shared between the transmitter and the receiver to encode and decode the message without a solid base of mathematics devoted to secrecy. In fact, a private secret key  $K$  is used





**Fig. 4.3** The wiretap channel of Wyner [28], where the eavesdropper’s channel is degraded relative to the main channel

for encryption of the message. The encrypted signal is then transmitted through a noiseless channel, introducing the perfect secrecy concept. As mentioned before, Wyner [28] opened a new era in information-theoretic security by introducing the wiretap channel concept. With his assumption, the intended signal  $X$  is forwarded to the predestinate receiver over the main channel, and the receiver receives  $Y$  that has passed through the wiretap channel without being detected by the eavesdropper as  $Z$ , depicted in Fig. 4.3. Wyner tried to maximize the secrecy rate (SR) in the main channel to minimize the leakage of information to the wiretapper. Wyner could prove via his formulas that security and secure communications can be achieved without the application of any secret key.

The information-theoretic security is divided into two major branches: secret key-based security and key-less security [16]. These two main fields of secrecy have evolved drastically during the last decades. By considering various features in the main channel and the wiretapper, different secrecy capacities have been achieved. In fact, many different transmission strategies based on CSI designation have been proposed, and channel coding is another strategy toward improving the physical layer security. A lot of research has been devoted to create secrecy-preserving channel codes. More recently, some strong and robust coding schemes have been proposed for distributed data systems and cloud-based systems. For instance in these types of systems, the data are distributed through various nodes, end users or data collecting devices should be able to regain the original data files from nodes, and these storage nodes are susceptible to attack and failures, so a novel concept was introduced as “regenerating codes” to satisfy the necessity of security in these types of systems.

#### 4.5.2 *Criteria of Physical Layer Authentication (PLA)*

Physical layer authentication and authorization plays the most fundamental part of the physical layer security (PLS) content. Indeed, the most substantial part is to make sure that messages are just received by the predestinate receiver. The receiver should be enabled to verify if the received message is intended for it or it has been modified by other users than source. The physical layer authentication methods are mostly focused on identifying different transmitters. Like traditional encryption-

based security approaches that are usually performed in upper layers, authentication was also applied in upper layers, but recently different research directions are appearing to consider the physical layer counterparts [4].

A proper authentication scheme should satisfy these three features: security, covertness, and robustness. Covertness in the authentication procedure implies that the authentication process must not influence the normal data transmission significantly and should not use much of computational power; also PLA does not affect the traditional upper layer authentication based on cryptography. The robustness is clear in PLA and means that PLA approach must be robust against fading effect and interference, and the third PLA properness factor called security obviously means to preserve the PLA process from malicious users and eavesdroppers but for sure meeting all these three properties, while designing a PLA scheme in a system is too complicated, so in most cases the focus is to achieve the security purposes in authentication. Different works have defined different authentication schemes, and different authentication evaluation metrics are defined to evaluate the authentication processes, such as detection rate and authentication rate.

The PLA schemes in general are basically categorized into two main categories: key-less and key-based authentication on the fact that the secret key is used for authentication or not. In order to clarify, let us present two simple ideas of key-based and key-less authentication. For instance, currently, physical layer features are used as authentication keys in many different approaches [4]. One of the approaches is to hide a pre-shared key in the modulation coding scheme that later is discovered by the receiver. There are also some key-less approaches for authentication in the physical layer. On the other hand, in a key-less procedure, the receiver will extract some transmission parameters, the parameters claim the source, and the authentication process is done by comparing the parameters with other authenticated messages.

## **Key-Based PLA**

In this section, we discuss how channel features can be exploited to generate keys for authentication purposes. We first discuss the one-way transmission approach. Then, we discuss the challenge–response method.

### **Authentication by One-Way Transmission**

In general, key-based physical layer authentication schemes are analogous to traditional symmetric cryptographic methods. The first studies on key-based authentication mechanisms commenced by the idea of generating secret keys with suitable coding schemes to provide authentication in the communication system. A general key-based authentication is depicted in Fig. 4.4.

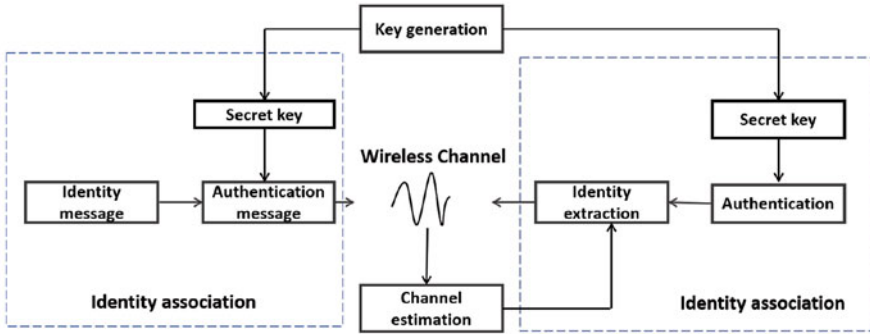


Fig. 4.4 A general representation of key-based authentication

Key-based PLA includes two principal phases:

- The first stage is the identification association, referring to the basic communication system of Alice, Bob, and eavesdropper, where Alice (the transmitter) generates keys and in the authentication tag includes the identification message information and forwards the message to Bob (the receiver).
- The second phase is the identification verification in which the identification verification is performed based on the received message and Bob's key.

### Authentication by Challenge–Response Transmission

PHY-CRAM represents the physical layer challenge–response authentication mechanism. The concept behind PHY-CRAM is analogous to the authentication mechanism that is applied in traditional security mechanism, but PHY-CRAM exploits the transmission channel randomness to secure the communication. In other words, it uses the channel characteristics to provide security. The short representation of this communication is illustrated in Fig. 4.5. The authentication process in this scenario is simplified as follows: Alice and Bob share the same secret key:

- First, Alice forwards a random challenge message to Bob.
- Bob applies a function and uses the received message as the input of that function to generate a response signal with the help of the shared secret key and sends back that response signal to Alice.
- Alice receives this response and employs the authentication procedure and her secret key to verify Bob's identification.

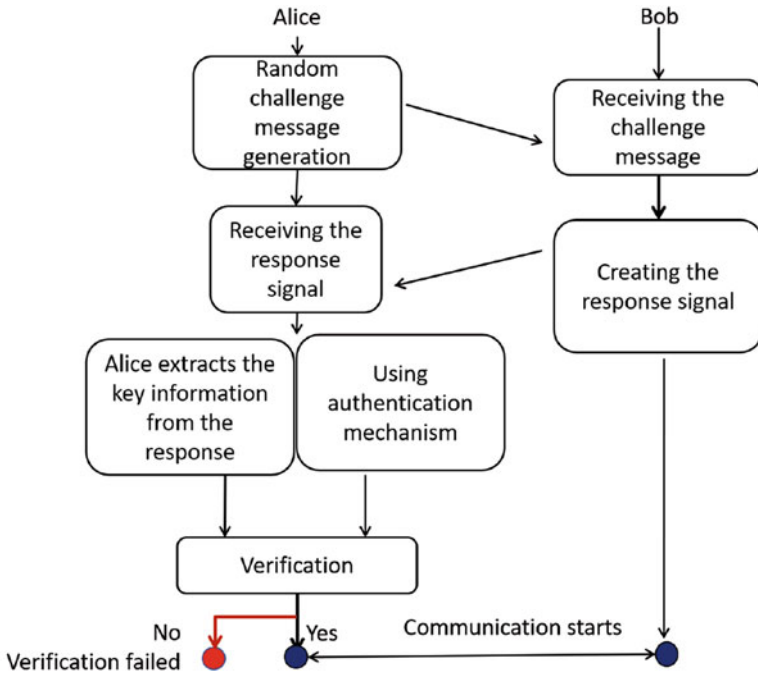


Fig. 4.5 A basic key-based PHY-CRAM system

## 4.6 Technologies and Solutions for Physical Layer Security in Fog Computing

Different wireless technologies can be deployed to cope with specific physical layer drawbacks. In this section, we review the main technologies that drive the transition from previous generations to 5G and beyond wireless networks. For each of these technologies, we then provide a connection with the aforementioned security and privacy threats. We first discuss the antenna technology with massive [Multiple-Input Multiple-Output \(MIMO\)](#). Successively, we will present how full-duplex transmissions may be used to provide security. Then, we discuss how interference can be exploited in the [Non-Orthogonal Multiple Access \(NOMA\)](#) framework. Last, we describe how physical layer attributes can be exploited to verify the location of users or devices.

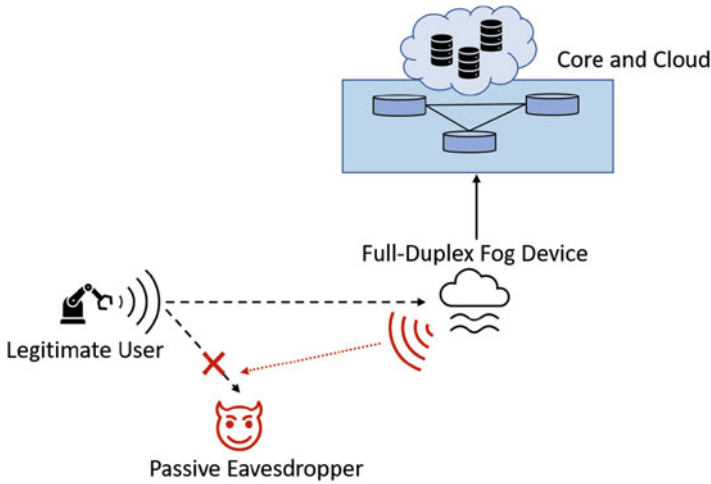
### 4.6.1 Massive MIMO

MIMO technology envisions the presence of multiple antennas at both the transmitter and receiver. In massive MIMO, the number of antennas is further increased. However, in the FC context, we should expect a larger number of antennas to be installed on fog devices rather than on the end devices. The reason behind this choice is the power and resource constraints of end nodes. This technology has also attracted a large research attention in the IoT field [3]. However, the presence of multiple antennas is well known to be a huge driver toward increased network capacity, especially in the transitions toward beyond 5G systems. Furthermore, this is particularly true when considering the path loss limitations imposed by the use of Millimeter Wave (mmWave) and THz frequencies. In fact, thanks to a large number of antennas, it is possible to increase the directivity of the signal to increase the directional signal power and overcome the drawbacks imposed by the higher attenuation at such frequencies. The technique used to convey directional signals is named beamforming and can be exploited for multiple purposes in PLS. Thanks to massive MIMO, it is possible to increase the network's privacy level. In fact, thanks to the use of multiple antennas, it is possible to convey the useful information to the legitimate receiver while at the same time jamming the eavesdropper [30]. Furthermore, it is possible to exploit the presence of multiple antennas to encode signals exploiting spatial modulation. In the FC context, it is however important to account for the limited resources of end nodes. Therefore, suitable solutions at the end layer should account for the energy consumption [25]. MIMO can also provide advantages toward jamming attacks [11]. In fact, suitable pilot transmission and retransmission schemes can prevent the damages caused by a jamming attack toward pilot sequences.

### 4.6.2 Full Duplex

The full-duplex paradigm envisions the simultaneous transmission and reception of signal over the same frequency bands. This allows for increased network capacity and reduced feedback and end-to-end delay. It is therefore an enabling technology for FC, as all these features are part of quality-of-service targets. Full duplex can be used to counter jamming attacks, where the receiver, while receiving the intended signal, also performs jamming [32]. Figure 4.6 shows the described scenario.

Another application of the simultaneous transmission and reception of signals is related to the detection of pilot contamination attacks [19]. Also in the case of full-duplex communications, the constraints in terms of available resources of end nodes are considered when designing secure network solutions. Although providing useful features, full duplex represents an additional tool also at the attackers' side. In fact, if the attacker holds a full-duplex device, she can simultaneously eavesdrop and jam a communication network.



**Fig. 4.6** Pictorial representation on how full duplex can be exploited at the receiver to increase the network privacy

### 4.6.3 Non-orthogonal Multiple Access

**NOMA** refers to the paradigm where multiple users are assigned the same set of time–frequency resources. Whereas in previous generations the main idea was to minimize the inter-user interference by exploiting a different set of resources for different users, recent advancements showed that **NOMA** provides higher spectral efficiency together with improved connectivity [9, 27]. This technology is particularly suitable for **FC**, where a large number of devices are connected to the fog devices. Successive interference cancellation will be implemented at the receiver to separate the multiple superimposed signals transmitted by the multiple devices. However, in case of pilot contamination, successive interference cancellation may lead to privacy and confidential data leakage. Anomaly detection techniques can be implemented to detect whether the pilot sequences underwent contamination. In this context, anomaly detection is implemented at the fog device, such that additional components are deployed in a smaller number of devices (i.e., only fog nodes) and do not undermine the life span of the edge devices by imposing additional operations. An example of an anomaly detection technique to tackle pilot contamination privacy leakage has been proposed by authors in [24]. Their proposed solution suits the **FC** scenario, as the additional operations are required only at the receiver’s side. The secrecy level that can be obtained in **NOMA** networks is accounted for as a fundamental metric when designing secure solutions. To enhance the secrecy level, cooperative jamming techniques can be exploited. In this case, multiple users agree on minimizing the signal to interference plus noise ratio at the eavesdropper’s side [22]. However, this assumes that users know the eavesdropper’s location, which might be a rather strong assumption. Therefore, secrecy rate models

such as [15] are included in the network design and optimization and should also account for random eavesdroppers' location.

#### 4.6.4 Context-Based PLS

The physical features of the channel can be exploited to regulate network and services access. In fact, the wireless channel is subject to the surrounding environment that causes transmitters located in different directions to face different reflections and attenuation values due to the presence of objects in the surrounding. An example is given by the presence of buildings in a certain area: two transmitters located at different sides of the building will face different channels for a given angle of transmission. Figure 4.7 depicts how physical layer context verification can be implemented. Upon collecting samples of the physical layer channel attributes in different directions, a fog device can detect whether a transmission comes from a legitimate user based on its channel features. This method is related to the presence of legitimate transmission areas or to the tracking of the legitimate user's channel and a suitable feature prediction method. An example of this application is in-region location verification [7], where physical channel features are used to detect whether a certain user is located in a pre-defined region of interest where access to the network facility can be granted. These types of applications should however account for the different types of attacks that may target the transmission power or weaknesses in the detection algorithm [8]. Region-based trust plays a fundamental role in guaranteeing trust among FC network components [31]. Thanks to the features of the physical channel, it is possible to verify whether a certain user is in

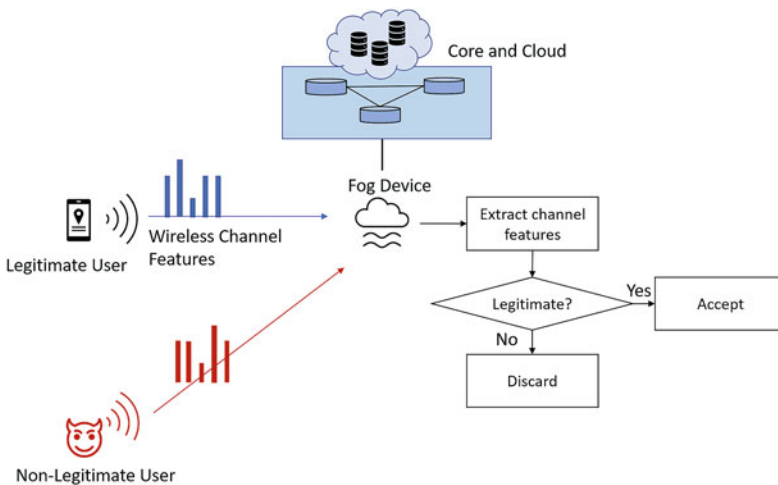


Fig. 4.7 Representation of physical layer context verification in FC

an intended area and shows certain expected physical channel features to augment the level of trust among devices. Thanks to the presence of edge devices (e.g., fog nodes), such techniques can achieve higher precision thanks to the definition of features pertaining to smaller physical areas. This represents a significant advantage compared to cloud computing, where, due to the absence of edge devices, a physical layer context includes a wider area lacking sufficient precision.

## 4.7 Conclusions

The widely distributed nature of FC and the resource constraints of its end devices demand for low-complexity solutions to guarantee the network security. In this chapter, we presented how PLS can be exploited to provide multiple security features while maintaining a low complexity. We analyzed how wireless connections are implemented in a FC network and discussed the main security features required by this particular network architecture. We reviewed the basic concepts of PLS. We then discussed how, thanks to beyond 5G technologies, PLS can be adapted to deliver the aforementioned security features. The integration of PLS in FC represents a fundamental step in delivering secure distributed next-generation networks.

## References

1. Abbas, M. A., & Hong, J.-P. (2017). Survey on physical layer security in downlink networks. *Journal of Information and Communication Convergence Engineering*, 15(1), 14–20.
2. Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the Internet of Things: Security and privacy issues. *IEEE Internet Computing*, 21(2), 34–42.
3. Araújo, D. C., Maksymyuk, T., de Almeida, A. L. F., Maciel, T., Mota, J. C. M., & Jo, M. (2016). Massive MIMO: Survey and future research topics. *IET Communications*, 10(15), 1938–1946.
4. Bai, L., Zhu, L., Liu, J., Choi, J., & Zhang, W. (2020). Physical layer authentication in wireless communication networks: A survey. *Journal of Communications and Information Networks*, 5(3), 237–264.
5. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (pp. 13–16).
6. Brighente, A., Conti, M., Donadel, D., & Turrin, F. (2021). EVScout2.0: Electric vehicle profiling through charging profile. arXiv preprint arXiv:2106.16016.
7. Brighente, A., Formaggio, F., Di Nunzio, G. M., & Tomasin, S. (2019). Machine learning for in-region location verification in wireless networks. *IEEE Journal on Selected Areas in Communications*, 37(11), 2490–2502.
8. Brighente, A., Formaggio, F., Ruvoletto, G., & Tomasin, S. (2019). Ranking-based attacks to in-region location verification systems. In *Proceedings of 2019 IEEE International Workshop on Information Forensics and Security (WIFS)* (pp. 1–6). Piscataway: IEEE.



9. Brighente, A., & Tomasin, S. (2017). Beamforming and scheduling for mmWave downlink sparse virtual channels with non-orthogonal and orthogonal multiple access. In *Proceedings of 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)* (pp. 1–6). Piscataway: IEEE.
10. Conti, M., Nati, M., Rotundo, E., & Spolaor, R. (2016). Mind the plug! laptop-user recognition through power consumption. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security* (pp. 37–44)
11. Do, T. T., Ngo, H. Q., Duong, T. Q., Oechtering, T. J., & Skoglund, M. (2016). Massive MIMO pilot retransmission strategies for robustification against jamming. *IEEE Wireless Communications Letters*, 6(1), 58–61.
12. Ferrag, M. A., Maglaras, L., & Ahmim, A. (2017). Privacy-preserving schemes for ad hoc social networks: A survey. *IEEE Communications Surveys & Tutorials*, 19(4), 3015–3045.
13. Habibi, P., Farhoudi, M., Kazemian, S., Khorsandi, S., & Leon-Garcia, A. (2020). Fog computing: a comprehensive architectural survey. *IEEE Access*, 8, 69105–69133.
14. Lin, H., & Bergmann, N. W. (2016). IoT privacy and security challenges for smart home environments. *Information*, 7(3):44.
15. Liu, Y., Qin, Z., El-kashlan, M., Gao, Y., & Hanzo, L. (2017). Enhancing the physical layer security of non-orthogonal multiple access in large-scale networks. *IEEE Transactions on Wireless Communications*, 16(3), 1656–1672.
16. Mukherjee, A., Fakoorian, S. A. A., Huang, J., & Swindlehurst, A. L. (2014). Principles of physical layer security in multiuser wireless networks: A survey. *IEEE Communications Surveys & Tutorials*, 16(3), 1550–1573.
17. Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: Challenges. *IEEE Access*, 5, 19293–19304.
18. Naha, R. K., Garg, S., Georgakopoulos, D., Jayaraman, P. P., Gao, L., Xiang, Y., & Ranjan, R., (2018). Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access*, 6, 47980–48009.
19. Prakash, J., Wang, J., & Lee, J. (2015). Detection of pilot contamination attack with full-duplex receiver. In *Proceedings of ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications* (pp. 54–55).
20. Shannon, C. E., (1949). Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4), 656–715.
21. Statista. (2021). Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030. <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>.
22. Su, B., Ni, Q., & He, B. (2018). Robust transmit designs for secrecy rate constrained MISO NOMA system. In *Proceedings of 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (pp. 1–5). Piscataway: IEEE.
23. Sullivan, S., Brighente, A., Kumar, S. A. P., & Conti, M. (2021). 5g security challenges and solutions: A review by OSI layers. *IEEE Access*, 9, 116294–116314. <https://doi.org/10.1109/ACCESS.2021.3105396>.
24. Wang, N., Jiao, L., & Zeng, K. (2018). Pilot contamination attack detection for NOMA in mm-wave and massive MIMO 5G communication. In *Proceedings of 2018 IEEE Conference on Communications and Network Security (CNS)* (pp. 1–9). Piscataway: IEEE.
25. Wang, S., Li, W., & Lei, J. (2018). Physical-layer encryption in massive MIMO systems with spatial modulation. *China Communications*, 15(10), 159–171.
26. Wu, Y., Khisti, A., Xiao, C., Caire, G., Wong, K.-K., & Gao, X. (2018). A survey of physical layer security techniques for 5g wireless networks and challenges ahead. *IEEE Journal on Selected Areas in Communications*, 36(4), 679–695.
27. Wu, Z., Lu, K., Jiang, C., & Shao, X. (2018). Comprehensive study and comparison on 5G NOMA schemes. *IEEE Access*, 6, 18511–18519.
28. Wyner, A. D. (1975). The wire-tap channel. *Bell System Technical Journal*, 54(8), 1355–1387.
29. (Sean) Zhou, X., Xianbin, W., & Matthieu, B. (2018). Best readings in physical-layer security. <https://www.comsoc.org/publications/best-readings/physical-layer-security>.

30. Yaacoub, E., & Al-Husseini, M. (2017). Achieving physical layer security with massive MIMO beamforming. In *Proceedings of 2017 11th European Conference on Antennas and Propagation (EUCAP)* (pp. 1753–1757). Piscataway: IEEE.
31. Zhang, P. Y., Zhou, M. C., & Fortino, G. (2018). Security and trust issues in fog computing: A survey. *Future Generation Computer Systems*, 88, 16–27.
32. Zheng, G., Krikidis, I., Li, J., Petropulu, A. P., & Ottersten, B. (2013). Improving physical layer secrecy using full-duplex jamming receivers. *IEEE Transactions on Signal Processing*, 61(20), 4962–4974.

# Chapter 5

## Blockchain for Secure Data Sharing in Fog-Based Smart City Systems



Elarbi Badidi and Essaid Sabir

### 5.1 Introduction

The phenomenal deployment of IoT solutions in various smart city systems results in massive amounts of data generated by sensors and IoT devices. Insight skills from data analytics will empower cities by offering smart services to citizens and businesses and reducing costs. They can improve traffic flow in real time, enhance waste management solutions by picking up bins only when they are full, detect water leaks, enhance car parking discovery, and offer smart street lighting. For this to happen, smart cities need to process data streams locally and share data securely between the various city stakeholders to create value-added services. Many applications need to efficiently process data streams in real time at the edge to respond to urgent situations quickly. Data processing and analytics at the fog level would enable getting more profound insights from the data and deploy latency-sensitive applications that edge devices cannot execute due to their limited resources.

In parallel with these benefits of fog computing, security is one of the essential concerns when designing new systems based on fog and cloud computing. Besides, privacy is another critical concern when storing and sharing data. Several works examined security and confidentiality issues in fog computing [12, 17, 27]. Fog

---

E. Badidi (✉)

Department of Computer Science and Software Engineering, College of Information Technology, UAE University, Abu Dhabi, UAE  
e-mail: [ebadidi@uaeu.ac.ae](mailto:ebadidi@uaeu.ac.ae)

E. Sabir

NEST Research Group, ENSEM, Hassan II University of Casablanca, Casablanca, Morocco  
Computer Science Department, University of Quebec at Montreal (UQAM), Montreal, QC, Canada  
e-mail: [e.sabir@ensem.ac.ma](mailto:e.sabir@ensem.ac.ma)

nodes usually belong to different organizations. Because of their proximity to users and their devices, sensitive and private information about users may be exposed, for example, when users are at home, when they visit new locations, when they use specific services such as healthcare services, and when they allow data sharing with other organizations. Robust encryption schemes should be used to store all user data securely. Besides, isolation is necessary for fog computing to ensure data privacy.

As IoT proliferates, concerns about data integrity and the protection of individual privacy also broaden. Many logical vulnerabilities in IoT devices could open doors for cybercriminals to take full advantage of any device. With an ever-increasing number of connected devices worldwide, the IoT industry needs to use a rigorous cybersecurity system. Blockchain might be the answer to this critical need [7]. One exciting feature of Blockchain, a distributed ledger (DLT) technology, is its ability to secure data sharing between organizations and thwart cyberattacks by relying entirely on encryption. The integration of Blockchain and IoT can enable the sharing of resources and services, leading to a market for services between IoT devices, build smart and independent systems that can use data with enhanced security, and ensure trusted distributed authentication of devices for IoT applications [4, 21, 23]. The principal security benefit of a distributed ledger is that if cybercriminals were somehow able to enter a chain, they could only access a tiny amount of data before the other nodes in the blockchain network realize that there has been a breach. The combination of a transparent ledger and this distributed security system gives Blockchain the upper hand in cybersecurity. Moreover, Blockchain can make IoT solutions secure and fast. For example, making payments and executing contracts just got easier with the peer-to-peer model of Blockchain. Smart contracts in the Blockchain eliminate the need to use the service of a trusted intermediary as transactions are approved or disapproved almost immediately, saving time in processing those transactions and saving millions of dollars for businesses [4].

This chapter proposes a Blockchain-based approach that allows secure data sharing among smart city stakeholders' fog nodes. Thus, fog computing ensures availability and low latency for applications, and Blockchain-based smart contracts and transaction handling guarantee the privacy required for data sharing. We consider a smart healthcare use case to illustrate the approach.

The contributions of this chapter are: (i) It addresses the data security issues in the context of fog computing and discusses the security techniques, which have evolved over the years to take many forms that can be used in fog nodes to protect organizational IoT data from external and internal threats. (ii) It describes a use case scenario concerning the secure sharing of patient data between healthcare providers who are permissioned Blockchain members. The paper describes the different steps for executing an "update of vital signs data of a patient" transaction using smart contracts.

The following sections of this chapter are organized as follows: Sect. 5.2 provides general information about fog computing and an overview of Blockchain. Section 5.3 describes various techniques to cope with the data security problem in fog computing. Section 5.4 presents the proposed architecture for secure data sharing

between fog nodes and details the process of handling a data sharing transaction in the context of smart healthcare. Finally, Sect. 5.5 concludes the chapter.

## 5.2 Background

### 5.2.1 Fog Computing

Many modern distributed computing architectures are organized into three or four layers, the IoT infrastructure layer, the edge layer, the fog layer, and the cloud layer, as shown in Fig. 5.1. In the three-layer architecture, the IoT infrastructure and the edge layers are combined to form one layer. The fog layer is at the middle level and is closer to edge gateways and IoT infrastructure than the cloud. Edge devices are connected to fog nodes through edge gateways, and each fog node is connected to the cloud. Additionally, fog nodes can be connected to share data and provide load balancing and fault tolerance [11, 13].

Fog nodes act as bridges between cloud servers and edge devices. They have more computing resources than edge devices and can process a large amount of data. However, when a data processing task is complex and time-consuming, the fog node must send the compute work to cloud servers. Fog nodes typically perform data management and processing operations such as rich and advanced data collection, aggregation, and analytics that involve machine learning and event processing. The main benefits of using fog computing are optimizing bandwidth, reducing traffic, reducing latency, and improving privacy and security.

In smart cities where multiple applications are time-sensitive, fog computing will play an essential role in implementing smart applications. For example, for real-time traffic and safety monitoring in public spaces to become a reality, operations personnel must react in real time to unexpected situations. These monitoring applications would not tolerate sending data to cloud servers and waiting for the results of data analysis. The system must have the capacity to process the sensed data and react instantly. Smart city facilities and systems are ideal for fog computing use. Indeed, sensors and actuators in the IoT infrastructure can receive commands based on decisions made locally without waiting for decisions made in remote locations. Smart city systems can use fog computing to get up-to-date information on the condition of facilities, roads, streets, and buildings to take corrective action before accidents or unwanted conditions occur. The processing of IoT data streams can be pushed from the cloud to the fog, reducing network traffic congestion and end-to-end latency.

### 5.2.2 *Blockchain for IoT*

Industry and the research community have considered Blockchain technology as a disruptive technology that is ready to play a significant role in managing, controlling, and, above all, securing IoT devices. A Blockchain is essentially a decentralized, distributed, shared, and immutable ledger database that stores the register of assets and transactions on a peer-to-peer (P2P) network. It chained blocks of data stamped and validated by miners. The Blockchain uses the elliptic curve (ECC) encryption and SHA-256 hash to provide substantial cryptographic evidence for authentication and data integrity [1]. The block data contains a list of all the transactions and a hash to the previous block. The Blockchain has a complete history of all transactions and provides cross-border overall distributed trust.

Bitcoin Blockchain has been the underlying platform and technology of many of the most popular cryptocurrencies today. However, with the advent of the Ethereum Blockchain, which implements smart contracts, the potential space for using the Blockchain has become endless. Similar smart contract Blockchain platforms have recently emerged. These include Hyperledger [15], Eris [5], Stellar [16], Ripple [2, 24], and Tendermint [14, 26]. Blockchain can solve IoT security challenges safely and effectively. It permits reliable and authorized identity registration, ownership tracking, and monitoring of products and assets. Approaches such as TrustChain [7] allow Blockchain-approved transactions while maintaining their integrity in a distributed environment. IoT devices are no exception. Blockchain can permit to register and give identity to connected IoT devices, with a set of attributes and relationships that can be uploaded and stored on the distributed ledger Blockchain.

Blockchain Smart Contracts provide decentralized authentication rules and logic to provide single, multi-party authentication for an IoT device. Also, smart contracts can provide more efficient authorization access rules for connected IoT devices, with much lower complexity than traditional authorization protocols such as Role-Based Access Management (RBAC) [25], OAuth 2.0 [9], OpenID [22], and LWM2M [19]. These protocols are widely used nowadays for authentication, authorization, and management of IoT devices. Furthermore, smart contracts can ensure data privacy by defining the access rules, conditions, and time required to enable specific users or groups of users or machines to own, control, or access data at rest or in transit. Smart contracts can also specify who has the right to update, upgrade, patch IoT software or hardware, reset the IoT device, provide new key pairs, initiate a service or repair request, or modify the ownership.

Permissioned Blockchains differ from their public counterparts in that they are ruled with permissions. Thus, not anyone with an Internet connection could access a permissioned Blockchain. These types of Blockchains could also be described as semi-decentralized. Control of a permissioned Blockchain is not assigned to a single entity but a group of authorized entities. With a permissioned Blockchain, the consensus process is different from that of a public Blockchain. Instead of allowing anyone to participate in the process, consensus participants in a permissioned

Blockchain are probably a group of pre-approved nodes on the network. Thus, permissioned Blockchains have the security features inherent to public Blockchains while allowing greater network control.

### 5.3 Data Security in Fog Computing

Edge devices and edge gateways typically send their data to fog nodes for storage and processing. As with cloud computing, there are mainly four kinds of data services in fog computing: data storage, data sharing, data processing, and data query and retrieval. These four data services have different unique data security and privacy requirements [8].

The primary goal of data security at the fog layer is to protect the data that each fog node collects from edge devices, stores, creates from received data, receives, or transmits to other fog nodes. No matter what device, technology, or process is used in fog nodes to collect, store, process, or manage data, it must be protected. Data breaches can lead to litigation and damage to the organization's reputation that owns or operates the fog node. The importance of protecting data stored, processed, or transmitted by fog nodes against security threats is becoming critical today with the proliferation of fog nodes, the increasing acceptance of fog and edge computing as a new form of computing, and the need for collaboration among fog nodes [17] [27].

Data security technology has evolved over the years to take many forms aimed at protecting organizational data from external and internal threats. Securing data at the fog level would typically require using the following techniques:

- **Data encryption:** Data encryption is a security method where information is encrypted by applying a code to every piece of it and can only be viewed or decrypted by a user or a process with the correct encryption key. Ciphertext, or encrypted data, would appear distorted or unreadable to a user or process accessing it without proper authorization.
- **Data masking:** Data masking is a technique to create bogus but realistic versions of organizational data to guard it against disclosure to external malicious sources, as well as the internal staff who could potentially use the data. The aim is to protect sensitive data while providing a functional alternative when actual data is not needed, for example, during user training, business demonstrations, or software testing. Several data masking techniques alter the data, including character shuffling, nulling out, data scrambling, word or character substitution, Pseudonymization, and encryption [20].
- **Data resilience:** With data resilience solutions, businesses create backup copies of their critical data, which could be recovered if it is accidentally corrupted or altered in a data breach. These solutions aim to protect data without disrupting operation, ensuring flawless business continuity during a system failure or a natural disaster.

When an entity collects any personal data, it immediately becomes identified as a data processor. This label comes with great responsibility. That is why many compliance regulations govern organizations that process personal data, regardless of type or volume. The rules that affect an organization will depend on factors, such as the industry in which the organization operates and the kind of data it stores. For example, if it stores data relating to the European Union (EU) citizens, it will need to comply with the General Data Protection Regulation (GDPR) [6]. Failure to comply with privacy regulations can result in hefty fines. Blockchain technology creates a data structure with built-in security qualities that rely on decentralization, consensus, and cryptography to guarantee trust in transactions. Data in this structure, called ledger, is arranged in blocks, and each block contains one or a set of transactions. Each new block connects to the blocks that precede it in a cryptochain, making it almost impossible to tamper. Each computer of the Blockchain network has a complete copy of the ledger. A consensus mechanism allows validating all transactions in blocks, ensuring that each one is true and correct.

## **5.4 Blockchain for Secure Data Sharing in the Fog**

### **5.4.1 Use Case Scenario**

One of the exciting scenarios of fog computing is smart healthcare, which is a typical case of IoT implementation. Healthcare data streams come from various sensors and IoT devices deployed in medical equipment and healthcare facilities and worn by patients. The huge amounts of data generated by these sensors need to be securely stored and shared among healthcare stakeholders. Thus, a fog-computing-based solution would be helpful in this case by creating an edge and fog tiers closer to the data sources to aggregate, integrate, and process generated data. Edge AI models will allow, for example, the classification of medical images locally without the need to transmit data to the cloud. For other time-sensitive healthcare applications, data streams could be analyzed immediately at the fog nodes. In contrast, for none of the time-sensitive applications, the data can be transferred to the cloud for deeper insights.

A patient may need to see her primary care physician for general health concerns, other specialists for minor health issues, and a dentist for dental care. These doctors can work with different health care providers. Necessary information, such as vital signs, as gathered by the primary care physician is vital to all other specialists. Other doctors would also like to know the drugs prescribed by each doctor. Other health problems or symptoms may be caused by or related to other conditions of the patient. By sharing health information between health care providers, an exchange of health information would greatly benefit physicians and the patient. However, health information sharing carries the risk of violating patients' personal health



information or being stolen by an unethical hacker. Health information is under serious threat wherever it is stored and whenever it is transferred.

### 5.4.2 Architecture

Figure 5.1 depicts our proposed fog and Blockchain-based architecture that consists of four layers: infrastructure layer, edge layer, fog–Blockchain layer, and cloud layer.

The edge layer consists of IoT gateways that are an essential component of any IoT implementation. They are responsible for aggregating the data, translating sensor protocols, and preprocessing the data before transmitting it to other layers (or to the cloud) for further processing. Aggregation and preprocessing of data received from sensors are necessary to cope with the massive amounts of data coming from sensors and devices. In the fog layer, the fog nodes at the edge of the network, which have more compute and storage resources, receive digitized and aggregated data from IoT gateways. Data that requires immediate feedback may undergo further processing before it is delivered to applications, shared with other nodes, or transferred to the cloud. These fog nodes, which can perform analysis at the edge, help alleviate the load on the IT infrastructure, as massive amounts of IoT data can easily overload data center resources and consume most of the network

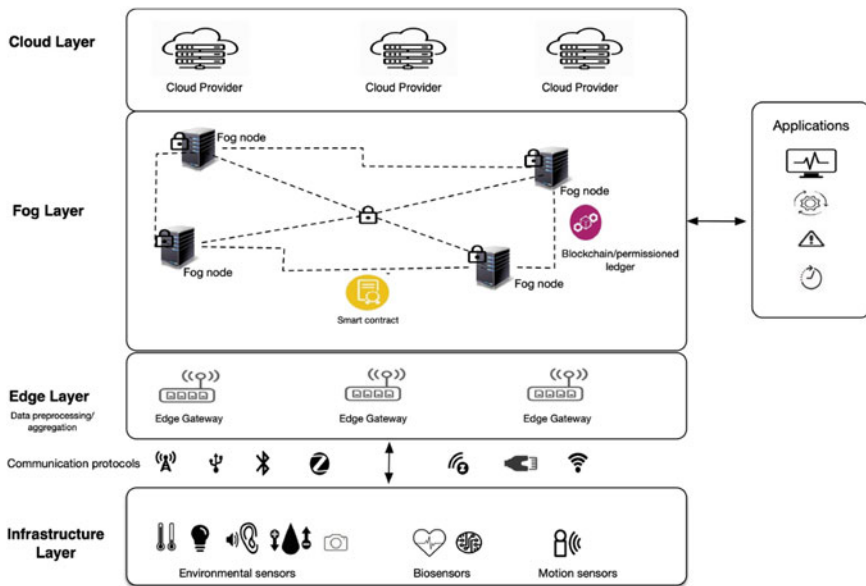


Fig. 5.1 Blockchain- and fog-based architecture

bandwidth. Data that does not require immediate feedback but does require further processing is transferred to cloud servers.

In the above use case scenario, the deployment of fog nodes at different sites will actively contribute to the design and implementation of distributed data storage and secure data sharing among healthcare stakeholders, using Blockchain and smart contracts technologies. Here, we consider data sharing and storage only at the fog layer. Fog nodes are associated with healthcare stakeholders. The consensus among different healthcare stakeholders must be achieved in a decentralized and distributed manner using consortium Blockchain to facilitate communications, trust, management, and coordination. We believe that a permissioned distributed ledger system, such as Hyperledger Fabric [3], would be the most suitable for healthcare in most health cases and processes as it guarantees a secure, private, and scalable platform that can connect all key stakeholders. The transacting parties can be healthcare institutions or individuals who want to enter into a contract that governs the exchange of their data or services. Clinics, insurers, laboratories, and pharmacies can get involved depending on the use case and act as asset protectors and transaction validators. Regulators can gain access to transaction records to monitor the system.

Several consensus algorithms were developed for Blockchain. The mechanics behind each one of these algorithms is not the focus of this chapter. Some famous consensus algorithms are proof-of-work (PoW), proof-of-stake (PoS), practical Byzantine fault tolerance (PBFT), and proof-of-authority (PoA) [18]. Cryptocurrencies typically use PoW and PoS algorithms. The consensus in Hyperledger Fabric includes three phases: endorsement, ordering, and validation:

- Endorsement phase is driven by a policy, such as  $n$  out of  $p$  signatures, upon which peers endorse a transaction.
- Ordering phase accepts endorsed transactions from clients and accepts that the order is committed to the ledger.
- Validation phase checks the correctness of the results of a proposed block of ordered transactions as well as the endorsement policy.

Hyperledger fabric provides support for using pluggable consensus services for the above three phases. Therefore, applications may use different pluggable consensus services for endorsement, ordering, and validation depending on their needs. In particular, the API of the ordering service allows plugging in Byzantine Fault Tolerance (BFT)-based agreement algorithms. The ordering service API provides two primary operations: broadcast and deliver [10].

The selective endorsement could be used to decide that transactions follow the healthcare business logic defined in the smart contract. Endorser peers are selected to agree on the transaction's validity by checking its satisfiability to the corresponding smart contract terms and conditions. Endorsed transactions are appended to the shared ledger in each participant's peer with appropriate confidentiality. When a healthcare transaction proposal is initiated, it is first sent to the endorsing peers for smart contract validation. The endorsing peers then decide whether the proposal is valid or not by simulating the transaction according to the healthcare business

terms and conditions. When the healthcare network's consensus is reached, the transaction information is hashed and included in a block that is appended to the shared ledger as an immutable record. The shared ledger stores the transactions in a secure and trusted way to prevent malicious nodes from corrupting the stored data. Since each peer node has its copy of the ledger, the whole network is updated with the new copy of the ledger. Finally, the transaction is executed, and the completion event notification is emitted. The following sub-section describes the process of handing a transaction for data sharing in the Blockchain network established by many healthcare stakeholders using the Hyperledger Blockchain.

### 5.4.3 *Transaction for Sharing New Data*

This transaction aims to update the patient's medical record with new readings of her vital signs, such as temperature, blood pressure, and ECG. These readings may be sent by the patient wearables through a mobile gateway to a health care provider's fog node, which can average them or store them in raw format. In other words, the transaction aims to share the new data of the patient in the health network, where each member of the network could access only authorized data. Seven steps are involved in executing a data sharing transaction from its proposal by one of the stakeholders' applications to its validation and execution as depicted in Fig. 5.2.

*Step 1: Proposing a data sharing transaction.* The client application of a stakeholder submits an *updatePatientVitalSigns* transaction to the endorser fog nodes of the Blockchain network. The endorsement policy specifies the need to have endorsements from specific Blockchain network members, such as members representing the patient primary care provider, health care regulator, and insurer. The other members of the network are not required to endorse the transaction.

*Step 2: Executing the transaction proposal.* When the endorser fog nodes receive the transaction proposal, they all execute the smart contract, depicted in Fig. 5.3, for the proposed transaction independently, and check all the rules defined by the smart contract (i.e., check whether the peer client is allowed to update patient data or not). Each endorser calculates a set of outputs for the transaction. Endorsements do not update the general ledger with the output of executed transactions.

*Step 3: Proposal response.* During the transaction execution, each endorser prepares a read-write (R/W) set for the transaction. The read set contains unique keys that the transaction reads during the simulation and their committed versions. The write set contains a set of unique keys and their new values that the transaction writes. An overlap between the two sets may occur. Each endorser responds to the client with the signed transaction and R/W set. The client then checks the consistency of the R/W sets received from all endorsers. Consistent responses mean that all endorser fog nodes read the same input and compute the same output. In other words, they all agree that the client's fog node can update the patient data. If there is any inconsistency between the response of the endorsers, the client ignores the transaction.

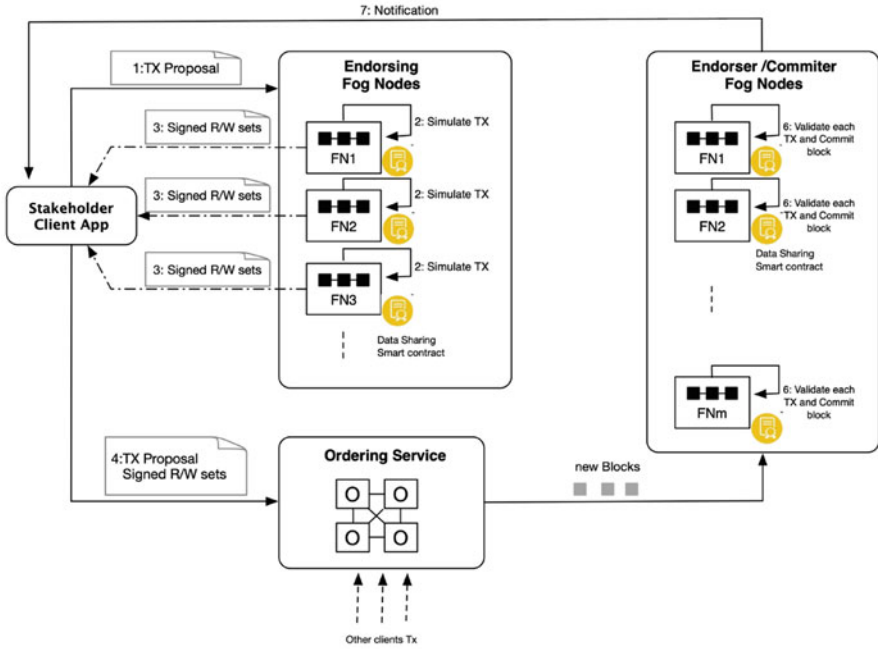


Fig. 5.2 Blockchain-based execution of the data sharing transaction

```
VitalSigns.ts x
1 @Transaction(false)
2 @Returns('VitalSigns')
3 public async queryPatientVitalSigns(ctx: Context, pid: string): Promise<VitalSigns> {
4     /*
5     The transaction returns a vitalsigns object of the patient with the same id as the pid parameter
6     */
7     // Check if the patient exists
8     if (!await ctx.getPatientList().exists(pid)) {
9         throw new Error('Patient with ID $(pid) doesn't exists');
10    }
11    // Return vitalsigns of the Patient from ledger
12    return await ctx.getPatientList().get(pid).getVitalSigns();
13 }
14
15
16 @Transaction(true)
17 public async updatePatientVitalSigns (ctx: Context, pid: string, newvitalsigns: VitalSigns) {
18     // Check if role == 'HealthProvider'
19     await this.hasRole(ctx, ['HealthProvider']);
20     // Check if the patient exists
21     if (!await ctx.getPatientList().exists(pid)) {
22         throw new Error('Patient with ID $(pid) doesn't exists');
23     }
24     // get the Patient instance and call its updateVitalSigns function
25     await ctx.getPatientList().get(pid).updateVitalSigns(newvitalsigns);
26 }
```

Fig. 5.3 Vital signs smart contract

*Step 4: Request to order the transaction.* The client submits the *updatePatientVitalSigns* transaction, the R/W sets, and the signatures received from the endorser fog nodes to the ordering service, which may receive transactions from other clients.

*Step 5: Delivering the transaction.* The ordering service adds the *updatePatientVitalSigns* transaction to a block with other transactions from other clients in the same channel. It then distributes the block to all fog nodes of the concerned channel. The endorser fog nodes receive the block, as do other nodes on the channel.

*Step 6: Validating the transaction.* The endorser fog nodes and other fog nodes on the channel add the block to the Blockchain. They also check that the *updatePatientVitalSigns* transaction in the block has the right R/W sets and signatures according to the endorsement policy. Since the transaction has been endorsed, they mark the transaction as valid, update their world state based on the write set, and update the patient data.

*Step 7: Notification about the transaction.* The different fog nodes on the channel issue notifications about the new block or transaction. If the client has registered to be notified when transactions succeed or fail, it is notified of the event.

## 5.5 Conclusion

In this chapter, we described how fog-based solutions would enable the deployment of latency-sensitive smart city applications. Processing and possibly storing IoT data streams in fog nodes closer to data sources can dramatically reduce network traffic and reduce latency. The deployment of fog nodes by different organizations and businesses in the smart city and the possibility of sharing data between them would allow them to meet the computing and network requirements of smart applications, which can help achieve the objectives of the smart city stakeholders and improve the quality of services offered to citizens. We described how setting up a Blockchain network including fog nodes of the city stakeholders would secure data sharing. We considered a smart healthcare scenario as a use case and described the process of executing a transaction that aims to update a patient's vital signs information collected by a healthcare provider. This process relies on validating the transaction against a vital signs' smart contract.

**Acknowledgments** This work is supported by the UAEU Program for Advanced Research Grant N. G00003443.

## References

1. Antonopoulos, A. M. (2014). *Mastering Bitcoin: unlocking digital cryptocurrencies* (1st ed.). Sebastopol: O'Reilly Media.
2. Armknecht, F., Karame, G. O., Mandal, A., Youssef, F., & Zenner, E. (2015). *Ripple – overview and outlook* (vol. 9229(3), pp. 163–180). Lecture Notes in Computer Science. Cham: TRUST.

3. Cachin, C. (2016). Architecture of the Hyperledger blockchain fabric. In: *Workshop on distributed cryptocurrencies and consensus ledgers*
4. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of Things. *IEEE Access*, 4, 2292–2303
5. ERIS-Industries. (2021). Eris: The smart contract application platform. Available online <https://erisindustries.com/index.html>.
6. European Parliament and Council of European Union. (2016). Regulation (EU) 2016/ 679 of the European Parliament and of the Council – of 27 April 2016 – on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/ 46/ EC (General Data Protection Regulation). Official Journal of the European Union (pp. 1–88).
7. Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., & Janicke, H. (2019). Blockchain technologies for the Internet of Things: Research issues and challenges. *IEEE Internet of Things Journal*, 6(2), 2188–2204.
8. Guan, Y., Shao, J., Wei, G., & Xie, M. (2018). Data security and privacy in fog computing. *IEEE Network*, 32(5), 106–111.
9. Hardth, D. (2012). RFC 6749 The OAuth 2.0 Authorization Framework . Internet Engineering Task Force (IETF). Available online <https://tools.ietf.org/html/rfc6749>.
10. Hyperledger.org. *Hyperledger architecture* (Vol. 1). Introduction to Hyperledger business blockchain design philosophy and consensus. Available online [https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger\\_Arch\\_WG\\_Paper\\_1\\_Consensus.pdf](https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf).
11. Kashani, M. H., Ahmadzadeh, A., & Mahdipour, E. (2020). Load balancing mechanisms in fog computing: A systematic review. Available online <https://arxiv.org/pdf/2011.14706.pdf>.
12. Khan, S., Parkinson, S., & Qin, Y. (2017). Fog computing security – a review of current applications and security solutions. *Journal of Cloud Computing: Advances, Systems and Applications*, 6(19), 1–22.
13. Khattak, H.A., Arshad, H., Ahmed, G., Jabbar, S., Sharif, A.M., Khalid, S., et al.: Utilization and load balancing in fog servers for health applications. *EURASIP Journal on Wireless Communications and Networking* 2019(1), 1–12 (2019).
14. Kwon, J. (2021). Tendermint: consensus without mining (2014). Available online <https://github.com/tendermint/awesome>.
15. Linux-Foundation. (2021). Hyperledger: Advancing business blockchain adoption through global open source collaboration. Available online <https://www.hyperledger.org/>.
16. Lohkava, M., Losa, G., Mazières, D., Hoare, G., Barry, N., Gafni, E., Jove, J., Malinowsky, R., & McCaleb, J. (2019). Fast and secure global payments with Stellar. *Symposium on Operating Systems Principles, 2015*, 80–96.
17. Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: Challenges. *IEEE Access*, 5, 19293–19304.
18. Nguyen, G. T., & Kim, K. (2018). A survey about consensus algorithms used in blockchain. *Journal of Information Processing Systems*, 14, 101–128.
19. Open Mobile Alliance. (2019). Lightweight machine to machine technical specification: Core. Available online [http://www.openmobilealliance.org/release/LightweightM2M/V1\\_1\\_1-20190617-A/OMA-TS-LightweightM2M\\_Core-V1\\_1\\_1-20190617-A.pdf](http://www.openmobilealliance.org/release/LightweightM2M/V1_1_1-20190617-A/OMA-TS-LightweightM2M_Core-V1_1_1-20190617-A.pdf).
20. Qiu, G., Gui, X., & Access, Y. Z. I. (2020). Privacy-preserving linear regression on distributed data by homomorphic encryption and data masking. *IEEE Access*, 8, 107601–107613.
21. Rahman, M. A., Rashid, M. M., Hossain, M. S., Hassanain, E., Alhamid, M. F., & Guizani, M. (2019). Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city. *IEEE Access*, 7, 18611–18621.
22. Recordon, D., & Fitzpatrick, B. (2006). OpenID Authentication 1.1. Available online [http://www.openid.net/specs/openid-authentication-1\\_1.txt](http://www.openid.net/specs/openid-authentication-1_1.txt).
23. Reyna, A., Martin, C., Chen, J., Soler, E., & Diaz, M. (2018). On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems*, 88, 173–190.
24. Ripple. (2021). Easily, efficiently, cost-effectively, instantly move money to all corners of the world. Available online <https://ripple.com/>.

25. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *Computer*, 29(2), 38–47.
26. Tendermint. (2021). Building the most powerful tools for distributed networks. Available online <https://tendermint.com/>.
27. Zhang, P., Zhou, M., & Fortino, G. (2018). Security and trust issues in Fog computing: A survey. *Future Generation Computer Systems*, 88, 16–27.

# Chapter 6

## Integrating Blockchain with Fog and Edge Computing for Micropayment Systems



Jamal Al-Karaki, Deepa Pavithran, and Amjad Gawanmeh

### 6.1 Introduction

In this section, the basic concepts behind the use of modern technologies like blockchain and fog computing for solving some real-life problems (e.g., micropayment) are described. Blockchain has developed as a powerful technology enabling unlimited application and opportunities during the last decade. Among these, it provided the first practical decentralized money exchange system. While there are many successful models for electronic coins based on blockchain technology, there is still a need for efficient and fast system that can support and process micropayments and the very low scale in a convenient manner [30]. Real-time analysis of data and authentication with better connectivity and faster speed of 5G will enhance the micropayment system when integrated with blockchain.

Fog computing is a new paradigm that is considered as an extension to cloud computing. As shown in Fig. 6.1, the computation is moved from the core of the Internet architecture layer to the edge of the network where processes are closer to end users. Figure 6.1 demonstrates the concept of how fog computing allow cloud services provisioning at the edge of the network. The architecture includes end-

---

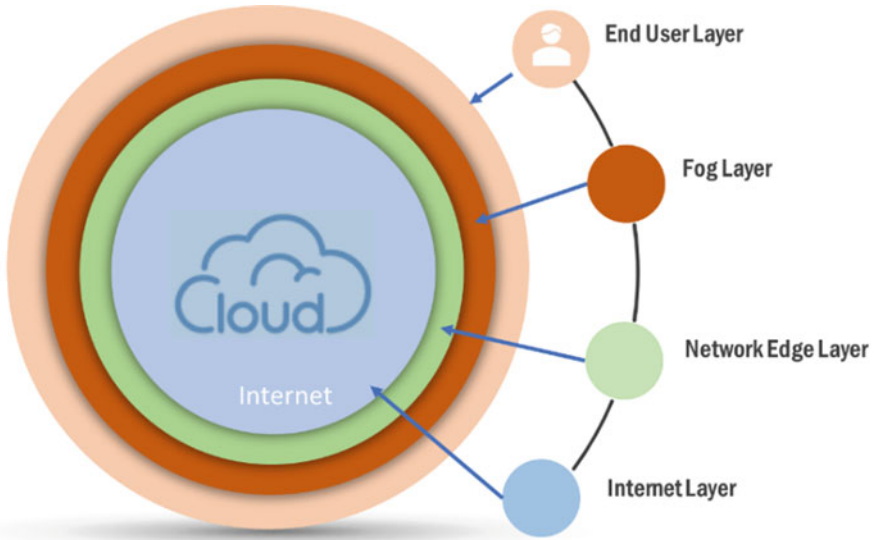
J. Al-Karaki (✉)  
Zayed University, AbuDhabi, UAE

The Hashemite University, Zarka, Jordan  
e-mail: [jamal.al-karaki@zu.ac.ae](mailto:jamal.al-karaki@zu.ac.ae); [jkarak@hu.edu.jo](mailto:jkarak@hu.edu.jo)

D. Pavithran  
Abu Dhabi Polytechnic, Abu Dhabi, UAE  
e-mail: [Deepa.Pavithran@adpoly.ac.ae](mailto:Deepa.Pavithran@adpoly.ac.ae)

A. Gawanmeh  
College of Engineering and IT, University of Dubai, Dubai, UAE  
e-mail: [amjad.gawanmeh@ieee.org](mailto:amjad.gawanmeh@ieee.org)



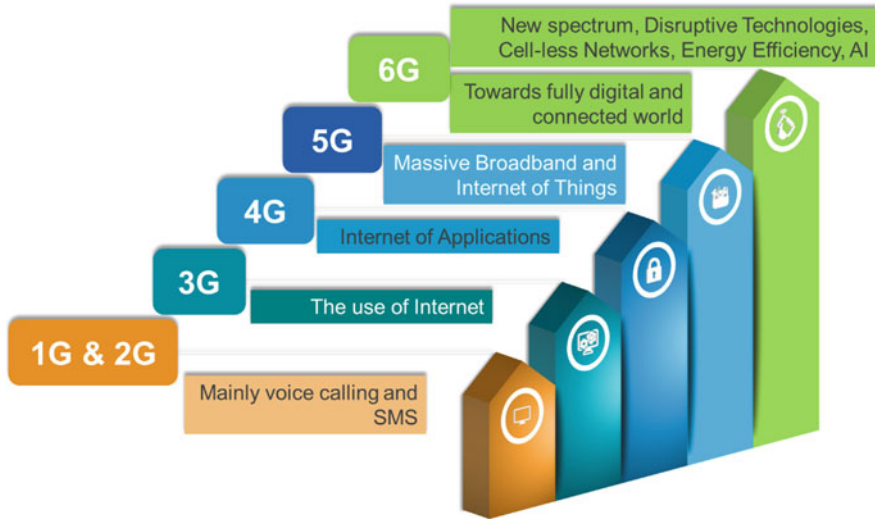


**Fig. 6.1** Extension of cloud services to be performed at the network edge using fog computing

user layer, fog layer, network edge layer, and Internet layer. Fog computing layer is perceived as a middle layer between the cloud and the Internet of Things (IoT). This multilayer extension has the benefits of enhancing several features that include security, reduced latency, and energy consumption. On the other hand, blockchain is becoming a core technology that is implemented in extensive range of applications. Many distinguished features of blockchain including security, reliability, and distributed trust management motivated extensive research on integration with fog computing. Such integration could potentially allow building a distributed, trusted, management system for processing heterogeneous data, payment and micropayments, trust and reputation, and validating user identity [31].

The Internet of Things (IoT) technology provides means for connecting anything such as devices, humans, cars, planets, road signs, etc. With massive increase of IoT devices in the coming future, current solutions consider cloud computing as a solution to resolve management issues. It is evident now that cloud computing may not be suitable for a massive IoT system. On the other hand, fog computing has the potential to manage the distribution problem on such massive scale, hence allowing for better controllability and manageability. In addition, a decentralized architecture may not be sufficient to handle sensitive transactions such as the ones needed for micropayment making blockchain-based solutions much plausible in less trusted environments [36].

Different generations of wireless and mobile technology were planned to meet the needs of both users and telecommunication network companies (see Fig. 6.2). As we move from 1G toward 5G, new applications appear with higher requirements for data rate and low latency. With more data-centric applications (e.g., smart cities,



**Fig. 6.2** Evolution of wireless communication networks and representative applications including 6G

digital cryptocurrencies, etc.), new requirements appear all the time. Furthermore, the new artificial intelligence-based smart systems residing in local cloud and fog environments will also enable a multitude of new applications. Modern communication networks will need to support this new smart system by allowing higher data transfer speeds with adaptation to heterogeneous set of networks/devices. Although 5G made a substantial step toward developing low latency, new frequency bands, advanced spectrum usage, and a complete redesign of the core network, the data-centric and automation still requires a data rate in the order of terabits per second with very low latency and many concurrent connections, which may exceed the capabilities of the emerging 5G systems. As such, a significant research was triggered to investigate the use of a new generation of wireless networks, i.e., 6G systems. The evolution of 5G to 6G will enable significant benefits where 6G could very much benefit from even higher spectrum technologies than 5G, e.g., through terahertz and optical communications. In addition, the heterogeneity of future network applications warrants new cell-less architectural paradigms like 6G. The 6G will also bring intelligence from centralized computing facilities to edge/fog devices allowing for more envisioned applications that were theoretically discussed under 5G networks. Overall, 6G will help to fill the gap between future business and public demands and what 5G can provide.

As both blockchain and fog/edge computing are based on decentralized devices rather than centralized servers, as in most other paradigms, their integration can help in driving many technologies forward. Future versions of blockchain will fix the defects in the older blockchain framework. Moreover, the use of fog computing in addition to some other advanced technologies such as machine learning will present



**Fig. 6.3** Sample applications for integration of fog with blockchain

many new features. Examples of these new features include adding new blocks to the blockchain publicly, allowing wide adoption by both individuals and businesses. In fact, when integrating blockchain with these new technologies of AI and fog computing, the operation of blockchain will completely change.

To elaborate on the use of fog computing for blockchain applications, we will now explain several capitalization factors on such integration. First, using fog computing will allow any type of devices to contribute to the blockchain contents, either directly or indirectly, and hence significantly enhancing the businesses in terms of production and running costs. Figure 6.3 shows three typical applications that can result from this integration and are explained further as follows.

1. **Assorted devices assembly:** In fog computing, smaller devices such as smartphones, tablets, and other smart devices will be closer to the edge and will act as nodes on a fog computing network. Shifting computation to the edge will make the fog faster, energy-efficient, and agiler when compared to cloud computing. As such, the devices will be added to the blockchain framework. As such, the restrictions of traditional cloud computing are removed with blockchain and fog computing integration. For example, fog computing will now remove restriction cryptocurrency mining to high-end machines. Even machines working on different operating systems can now operate in the blockchain framework allowing for cross-platform manageability. In fact, blockchain smartphones are already being used in many businesses today [32].
2. **Client token system:** Using fog computing for blockchain, the business value for idle/unused digital resources can be maximized. Since many computing resources are connected to the fog-based blockchain and these might not be completely utilized, businesses might offer a lease token for users who rent their idle equipment resources such as CPU/GPU, storage space, and bandwidth for handling business processes of the organization. For example, a reward system can be built to offer different tokens for the type of resource being shared by users. This is very promising application given the large number of devices

that can be integrated in the fog computing for the blockchain framework. As such, idle times of various digital gadgets can be utilized to access blockchain applications. The benefit will be huge for various businesses from the availability of a large number of blockchain-enabled IoT devices. Example of usage of businesses for these devices is to have a platform for trading cryptocurrencies without being reliant on large data service providers. The reward system can also allow for earned and unutilized tokens to be traded for cryptocurrencies with other users. As such, fog computing-enabled blockchain networks allow for new utilization frontiers of heterogeneous devices for both consumers and businesses. In addition, utilization of disruptive technologies like AI or big data can help blockchain change the data that is processed and exchanged throughout the Internet from all types of devices, which will allow new methods for handling cryptocurrency transactions [33].

3. **Blockchain and fog synergy:** A major challenge in a distributed environment like fog computing is to have how to employ distributed security structure in order to protect network resources and businesses. As fog-enabled blockchain performs as a mesh system with equal roles of network nodes of equal computational loads, a distributed security solution is also needed especially when various layers of the fog node heap are managed by many different units. To this end, blockchain technology is the answer for managing trust in a decentralized and distributed manner where users don't trust each other [34] [35].

This chapter explains the benefits of integrating modern technologies (fog computing, blockchain, and IoT) to solve the problem of micropayment systems. Toward the end of this chapter, we briefly present a generic solution proposal to the problem of micropayments by integrating fog computing capabilities, blockchain, and edge computing to provide a practical payment setup that allows customers to issue micropayments in a convenient manner and at a fast rate. This is achieved by utilizing the capabilities of edge computing to provide ad hoc but high computational power, as well as the fog computing technology in providing reduced latency in processing. The objective is to provide a practical and novel payment setup that allows customers to issue micropayments in a convenient manner and at a fast rate. In particular, we are interested in understanding how new technologies can make it possible for users to make micropayments in a distributed environment. We leverage distinguished features of these new technologies (e.g., blockchain and fog computing with IoT) to explain how better solutions for this problem can be obtained.

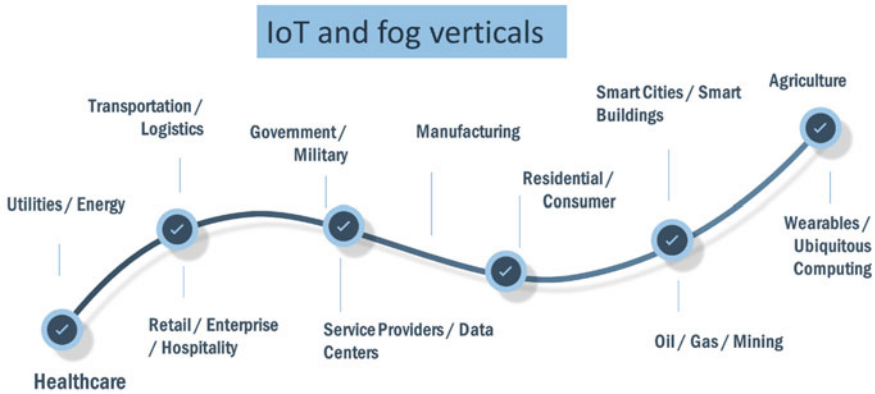
The rest of this chapter is organized as follows. In Sect. 6.2, the taxonomy of blockchain-based system architecture for fog computing using IoT is described in details. Section 6.3 describes the blockchain-based system architecture for fog computing with some related work about the use of blockchain and fog computing applications. In Sect. 6.4, we present the system model to illustrate how a system should be constructed to address the fog-blockchain-enabled micropayment system. In Sect. 6.5, we briefly present a generic model for micropayment system. In Sect. 6.6, we conclude our work and highlight some future directions.

## 6.2 The Taxonomy of Fog Computing, Internet of Things (IoT), and Blockchain

The trust problem in technology applications is enormously complex when sensitive information (e.g., cryptocurrencies) is encountered and no verification or audit mechanisms are provided. Nakamoto, in 2008 [1], presented two new concepts which heavily impacted trust system in various businesses, namely, the Bitcoin virtual currency and the blockchain, a technology that allows for making a public and open distributed ledger [30]. Recently, blockchain has been identified as a disruptive technology which has adverse impact on many industries and businesses. Blockchain has various types, e.g., public and permission based. Most of today's platforms are permission based. The smart contract is used to verify the user identity. The consensus algorithm is used to reach a decision for adding a new block to the distributed ledger. There are many variations of consensus algorithms, e.g., proof-of-work (PoW) and proof-of-stake (PoS). Many common platforms exist today for the creation of blockchain-based applications from many diverse domains, due to the tremendous advantages of blockchain technology. The number of platforms is so high and in constant change. These platforms vary in terms of the type of blockchain used, the consensus algorithm, and if smart contracts are used. An example is the Ethereum blockchain, which uses a time-stamp system and cryptographic hashes to prevent alteration retroactively and records all transactions that occur within the network. It also uses smart contracts to make sure that all of the specifications of any particular user, such as limited access time to a network, can be defined, ensured, and recorded. Other common platforms include Hyperledger Fabric, Multichain, Lisk, and Quorum [15].

The synergy between fog computing and blockchain is true for connected systems with lack of trust as well as in a disconnected or autonomous systems. The reason behind this synergy is that fog computing provides atomic features that requires trust while operating independent of central servers. Among various consensus mechanisms, some cannot work when it requires massive computing capacity that are not supported by a fog device. On the other hand, protocols such as "proof-of-stake" (PoS) are suitable of running on nodes with limited capabilities of fog nodes. The massive use of fog computing motivated the need to have an interoperable architecture for blockchain in fog environments. Several blockchain-oriented startups join the OpenFog Consortium, including iExec, Hyperchain, KeyChain, Xage, SONM, and Leatherworks [31].

The incorporation of fog computing into the Internet of Things (IoT) using blockchain poses several challenges as well as opportunities. To understand this interesting mix, potential use cases need to be studied in order to understand requirements and architecture and demonstrate the value presented by fog computing to end users. To aid our understanding, a proposed taxonomy is explained below to undersigned how fog can be employed to harness the IoT capability coupled with blockchain [36]. The proposed taxonomy has three levels, namely, verticals/vertical markets, use cases, and applications. Figure 6.4 shows sample verticals that can



**Fig. 6.4** Sample verticals for integration of fog with IoT and blockchain

result from the assimilation of fog, IoT, and blockchain. This is further explained as follows:

1. **Vertical markets:** Verticals or vertical markets are business functions where companies serve a specific audience (e.g., industry segments, or network domains, or specific classes of users) and their set of needs. Vertical markets are more dependent electronic commerce due to the need for instant transactions processing. In IoT fog environments, this is related to specific businesses with dedicated and highly specialized team for an individual vertical. For each vertical, a set of standards, policies, procedures, and protocols are developed to help control its widespread use. Here are some of the verticals that are especially important to IoT and fog:
2. **Use cases:** The second layer in this taxonomy splits each vertical into sections that are served by a single IoT platform. For example, use cases might include smart highways, autonomous vehicles, and drones, among others in the transportation vertical. When designing fog-based solution for a particular use case in a vertical, a solution can be effectively leveraged for another use case in the same vertical.
3. **Applications:** In this layer, a specific hardware/software solution can be built to provide certain IoT capabilities to satisfy customers' needs. For example, in the transportation vertical, some applications can be scheduling, fuel optimization, passenger entertainment, shipment tracking, and staff communications. These applications can be installed on the fog network by the transportation supplier and make it available for different clients. Similar set of applications can also be built for other verticals making this area an emerging marketplace for fog computing and IoT applications. When the marketplace becomes prominent, third-party developers can be leveraged to grow the IoT and fog software or applications.

The above-suggested taxonomy clearly shows how IoT networks will use fog network and fog nodes to serve most of the verticals in Fig. 6.3, their associated use cases, and the wide range of applications serving each use case.

### **6.3 Blockchain-Based System Architectures for Fog Computing**

In this section we will discuss various paradigms where blockchain-based systems were used within centralized data centers as well as distributed ones. For this reason we will first present cloud-based and then fog- and edge-based blockchain architectures.

#### ***6.3.1 Cloud- and Big Data-Based Blockchain***

Cloud architecture is a centralized one where the system consists of a huge number of data centers as well as computational resources. On the other hand, blockchain network is a distributed public ledger where data and operations are incorporated in a sequence of transactions [2]. These transactions are recorded and verified throughout a distributed and decentralized network of nodes. This decentralized architecture has been shown to have unprecedented capabilities to develop several types of applications. In addition, several research works have demonstrated how blockchain can be used for assured data provenance capability for cloud computing systems [3]. Such model can make use of the capabilities of both paradigms. For instance, the decentralized architecture can utilize every node participating in the system to enhance the efficiency of cloud services.

On the other hand, integrating blockchain with cloud services results in having all data operations conducted transparently and permanently recorded. One benefit of this model is having a new level of trust between service users and cloud providers. In addition, maintaining provenance can help in increasing the trust toward cyber threats and provide strong background to enable proactive cyber defenses [3, 4]. Figure 6.5 shows the micropayment dataflow within the user, edge devices, and blockchain. Users make the payment request with microprocessors. Edge/miner nodes verify the balances and acknowledge. If adequate balance is not available, the request will be rejected. Balances are updated on all nodes within the blockchain.

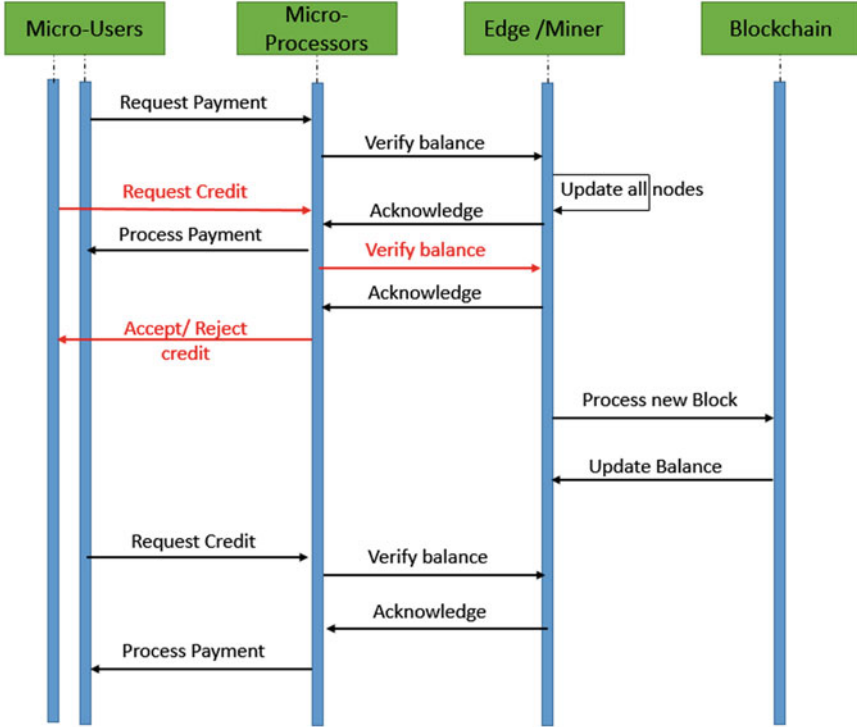


Fig. 6.5 The micropayment dataflow within the three layers (user, edge devices, and blockchain)

The authors of [5] presented a security framework for data storage in the cloud using blockchain. The method is based on dividing data encrypted blocks that are processed randomly within blockchain networks. Juneja et al., the authors in [6], proposed another blockchain-based framework for access control system that enables data processing during retraining in real time. In another approach Shafagh in et al. [7] presented a blockchain-based system with access control and management features for IoT Data. Several other approaches were proposed to securely and efficiently collect, organize, and audit big data for model building and accurate prediction using machine learning methods [5–11].

The work in [12] presented a blockchain-based user authentication algorithm that is intended to address cloud insider attacks. The blockchain is used for storing and processing credentials within secure transactions. Then, access is provided only after proof of authentication blockchain-based transaction (PoAh) [13].



### **6.3.2 Fog- and Edge-Based Blockchain**

Fog computing emerged as a new paradigm that enables making use of end users to perform computations and other network operations that are related to the core network. The work in [14] presented a lightweight security method that integrates fog computing with the blockchain. The framework makes use of edge and fog computing capabilities in order to enhance the speed of signing and validating signed data.

Authors in [15] presented a cooperative framework that integrates blockchain and fog architectures for food supply chain. Authors used capabilities of blockchain technologies to enhance the transparency of data sharing as well as information flow and management capacity between all end users of the supply chain system. The work in [37] proposed a method that integrates blockchain with fog computing networks in order to enhance computational power consumption and storage spaces. This is achieved through a heuristic that was designed to enhance hashing time for blockchain throughout device collaboration. Table 6.1 shows the overview of state-of-the-art work on blockchain/edge/fog technologies with potential applications in micropayment systems, or to support micropayments in IoT systems. It is obvious that existing methods do not make use of the current capabilities of the 5G networks. In fact, some of the aforementioned proposals were there before 5G came to the picture. In addition, most of existing solutions are trailered toward particular applications, mainly related to IoT.

Practically, there are many other contemporary applications that may make use of Blockchain/5G enabled instant micropayment biotins, especially, the gaming area.

## **6.4 Fog-Blockchain-Enabled Micropayments**

In the Internet of Things (IoT), the existing payment method includes using a prepaid card or mobile devices to make the payment. In this system, the users have to provide cards, enter the password, or operate devices. In addition, this system relies on a trusted third party for processing financial transactions. This process can be automated using blockchain by employing a hardware cryptochip, where the device can make the payment automatically without the need for a trusted third party. This will help to ease the complicated transaction procedures.

Two reasons why a traditional blockchain system is inappropriate for micropayment system are that cryptocurrency exchanges impose transaction fee that may become higher than the data value. Secondly the traditional systems are less scalable and have low transaction speed. In addition, the reliability of blockchain mainly depends on how its consensus is designed. A poorly designed consensus can entirely disrupt the blockchain process and can lead to business loss. Blockchain uses a distributed ledger technology where all parties share the ledger making it

**Table 6.1** Overview of state-of-the-art blockchain/fog/edge methods with potential IoT or micropayment support

References	Integrated technologies			Application	Micropayment	
	Fog/Edge	Cloud	5G		Support	Blockchain
[16]	✓	✓		Healthcare		
[17]	✓	✓		IoT devices		
[18]	✓			Smart city		
[37]	✓			Industrial 4.0 applications		[37]
[38]	✓			AI data processing		[38]
[19]				Smart grid	✓	
[20]				Micropayment	✓	Sidechain and Lightning network
[17]	✓			IoT security		
[21]				Gas meter	✓	Blockchain
[22]				Organization	✓	Ethereum
[23]				IoT payments	✓	Ethereum
[24]				Intermediate payments	✓	Blockchain/TumbleBit
[25]				Off-chain payments	✓	Lightning network
[26]				Anonymous payments	✓	Blockchain

a transparent system. This ensures the reliability of the system and avoids illegal modification of the system.

The system proposed by Lundqvist [19] utilizes a smart socket and smart cable that communicates with the socket. The objective is to enable blockchain-based micropayment system to automatically allow a “Thing” to pay for its electricity. This can be utilized by electric cars to automatically refuel. The cable pays for the amount of electric energy using Bitcoin. Hence, the cable needs to be set up with Bitcoin accounts and users need not be aware of the payment.

### 6.4.1 Benefits of Integrating Blockchain with Fog Computing

IoT infrastructure is mainly adopted for monitoring and controlling applications in critical infrastructure. Integrating this infrastructure with cloud computing provides on-demand storage and processing services. Since sensors are deployed in huge numbers in IoT era, this results in big data that requires a large bandwidth for data collection and acquisition. In addition, moving data into data centers where costs are at lowest is considered one of the challenges in this area.

A model of micropayment system enabled by the Bitcoin system to make payment for smart devices is given in Fig. 6.6 [21]. It includes the application, users, the network communication module, Wi-Fi module, processor, cryptochip,

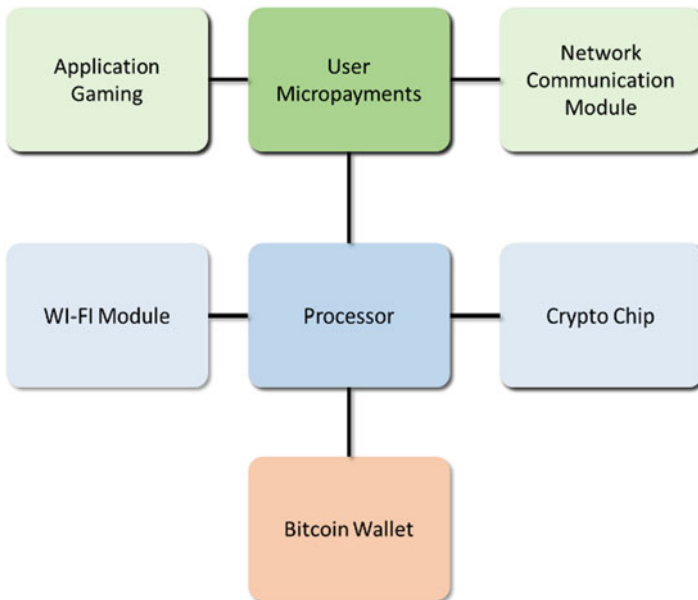
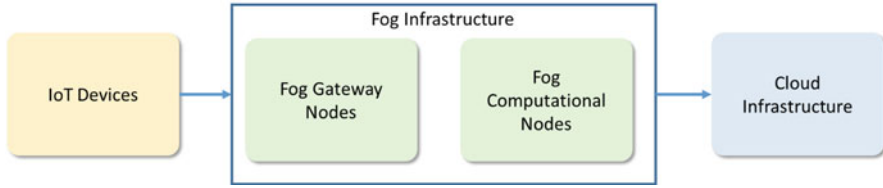


Fig. 6.6 A Model of micropayment system using Bitcoin



**Fig. 6.7** General fog-enabled IoT system architecture

and Bitcoin wallet. A Bitcoin secure payment module makes the transactions. The payment module contains a processor with a Bitcoin wallet stored on it and an elliptic curves cryptography (ECC) cryptochip.

Processing the data near to the edge of the network can improve the quality of service provided for the end user. Integrating blockchain with fog computing can provide data security, transparency, and trust without the need for a central party. This in turn provides better control over data and communication. Fog computing can also provide data storage on-site and can process and classify the data based on the sensitivity. Analysis can be done at the fog layer to identify what data should be stored in the blockchain and what data should be moved to the centralized cloud.

A general fog-enabled IoT system is based on three layer architectures: device layer, fog layer, and cloud layer [17] as shown in Fig. 6.7. The fog layer serves as a layer between IoT and cloud. The fog infrastructure includes the fog gateway nodes and fog computational nodes. Before sending the data to the cloud, the data is processed from the fog computational nodes. This in turn reduces the latency in sending the data to the fog. Fog gateway node acts as an interface between the IoT layer and cloud layer.

#### ***6.4.2 Integrating Blockchain with Fog/Edge Computing for Micropayments***

Blockchain has a robust and trusted solution for financial transactions; it can serve as a billing layer in between a distributed network of heterogeneous devices. Memon et al. [27] proposed three different configurations for hybrid IoT based on the applications. It is a three-tier architecture with “Things,” edge/fog layer and cloud layer. “Things” are objects or devices deployed into a smart environment which are connected into a blockchain-based peer-to-peer communication network. The middle layer is the fog/edge layer where fog is computing resources that have processing, storage, and controlling capabilities available locally to IoT devices. The third layer is cloud, where only certain applications will be communicating to the cloud layer. These are mainly industrial applications that require high processing and storage capabilities. For applications that require micropayment, the cloud layer is inactive, and edge/fog layer will be based on need only. Hence for

objects in smart home, vehicle-to-vehicle communication, and traffic management, “Things” can make payment without interacting with the cloud layer. Applications such as smart meter, utility billing that requires micropayment edge/fog layer, will be active, whereas cloud layer is based on need only. This indicates that micropayment applications can be processed within in the “Things” and edge/fog layer. Introducing blockchain within the “Thing” or the edge/fog layer can create a robust and transparent payment system. However, due to the large transaction fees and less scalability, blockchain-based payment system needs extensive changes in processing micropayments.

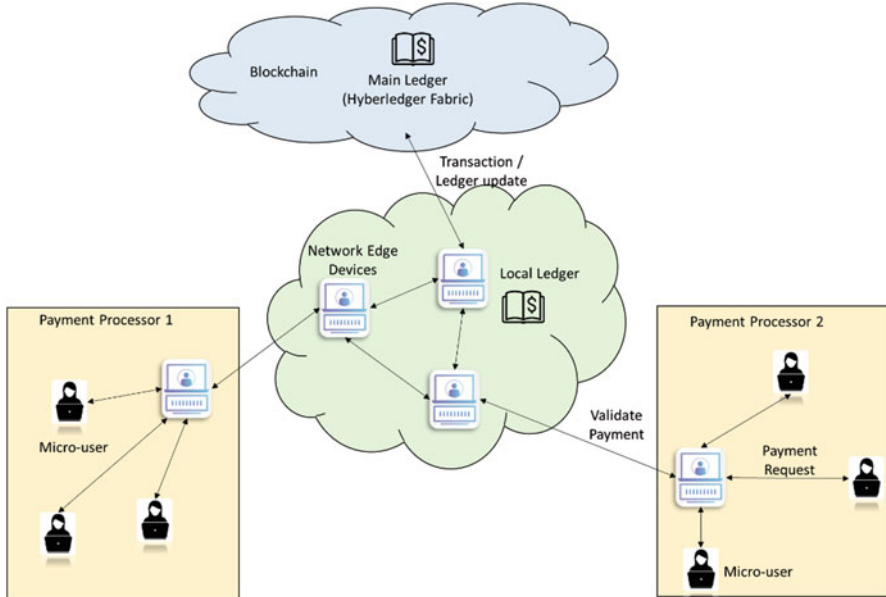
### ***6.4.3 Lightning Network: Example of Micropayment Solution***

Lightning network (LN) is an off-chain protocol that can process micropayment. The transaction fee is negligible and can be processed in wide ranges in the micropayment domain. It can be used as payment solutions for IoT applications and digital goods and services. The advantage of using LN network is cheaper and faster payment processing with enhanced scalability, latency, and throughput. Unlike other blockchain networks, the LN does not broadcast the transactions to the entire network. Two parties in the network can perform off-chain transactions through local database. Hence, the on-chain transactions, which are computationally expensive and slow, can be avoided, and a secure communication channel is created between two parties. LN is created on top of Bitcoin network [28, 29].

LN runs by first opening a channel by creating a transaction request to the edge node. When this transaction is validated, payment is executed and the balances are updated. When any of the party is willing to terminate the transaction, the updated ledger with balance must be synchronized. Its operation also includes punishment when any of the party misbehaves. In such case all the channel-related funds can be kept by the other party. Transactions cannot be finalized in cases when payments are made in single direction or if the balance of a party reaches zero [20].

## **6.5 A Hierarchical Fog-Blockchain Micropayment System**

In this section, we show an architecture of a hierarchical fog-blockchain micropayment system. The main practical problem in most cryptocurrency is the lack of scaling and the ability to process instant small amount of payments. These are very often needed in different applications, in particular in the gaming area. Among the solutions that were proposed in the literature is to use an off-chain network, called lightening network (LN), in order to enable transactions between without actually performing them in the main blockchain. This provides users with the ability to do as many transfers as they want internally, without having to update the main ledger in the blockchain [27]. This obviously has several advantages in terms of



**Fig. 6.8** Blockchain-based edge/fog computing generic model for micropayment systems

efficiency and scalability as well as reducing transaction costs. Figure 6.8 shows the fog/edge computing model for micropayment system using blockchain. The edge device keeps a local ledger, whereas the main ledger is maintained within the blockchain in cloud. Micropayments are accepted and processed by the edge devices through the local ledger. It is noted that this model can provide fast and reliable exchange of data with low latency. An edge device can register with the fog node/server using a private key. The authority of the transaction of the edge device is verified using the public key. When processed, the peers in the same payment processor group can request transactions such as resource exchange of data of the registered edge device.

In the LN-based approaches, the role of the end user is limited to performing transactions; in addition, there are no criteria which defined on who to aggregate the transactions among users, and then perform the actual blockchain update. In this work, we intended to present a layer's micropayment that makes use of the LN architecture as well as the capabilities of edge/fog computing features. The objective is to enable accepting and processing micropayments instantly by a layer of users. This will be conducted in an LN fashion off-chain. However, users will be acting as fog/edge nodes that can contribute to the blockchain as PoS or PoW or using any other concept. In addition, other layers for processors that act as intermediate one between users and the blockchain, which can be called processors layer, will have to aggregate these and process them through the normal blockchain. Figure

6.7 illustrates this architecture where edge nodes can act as users or processor. In addition, they can contribute to the system by acting as minors in order to conduct proof-of-stake or proof-of-work.

1. **Micro layer:** here micropayments are processed fast; this includes transactions with very small amounts that are processed by edges. Each edge will gain positive reputation for processing micropayments successfully. These are instantly processed, and aggregated together to form transactions that will be processed by the macro layer. Transactions can be confirmed by the edge. Every participating node can act as an edge. These transactions are processed instantly by each processor. Processors will then aggregate these payments into transactions that can be processed through the normal blockchain, be it Bitcoin, ether, or any other infrastructure. The main objective of this layer is to enable instant payment. Processor can coordinate their limits with end users based on trust gained by these users.
2. **Processors layer:** in this layer, transactions are aggregated and processed between edges as regular blockchain transactions; these are slow, and take time to be confirmed and processed. Transactions can be confirmed only by blockchain. The aggregation process is performed at this level by the edge processor node based on the current state of the ledger and the balance of every user. These will be eventually updated and synchronized among all edge nodes. In addition, processors will have to validate micropayments across other processors. This should not be time-consuming and will require a lightweight protocol step that ensures consistency between processors. Such protocol can be implemented based on 5G technology.

As demonstrated in Fig. 6.9, the micropayment processor layer performs transactions with the blockchain. These processors are typically edge/fog nodes. They hold a consistent and up-to-date ledger that can be synchronized at this layer without interfering with the blockchain. Any messages exchanged between the processors layer, or between this layer and users, or between this layer and the blockchain will be synchronized in this layer. Hence, this layer is responsible for facilitating transactions between users, or users and processors by manipulating transactions along with associated data and then synchronizing this with remaining processors. This synchronization mechanism can guarantee that micropayments are processed and locked until eventually aggregated into the blockchain.

A generic architecture of micropayment system with its integration with other system is provided in Fig. 6.10. The role of 5G technology is fundamental in the process of synchronization among processors in their layer. For a small-scale network, this can be done on the spot with such high-performance capabilities, while large-scale networks will require more involved solution. Another open area is establishing a trust-based system that regrades edge/fog nodes at both layers. This can significantly enhance the performance of the network by reducing the complexity of the synchronization problem. Finally, employing smart contract in this context is another open area that requires further investigation.

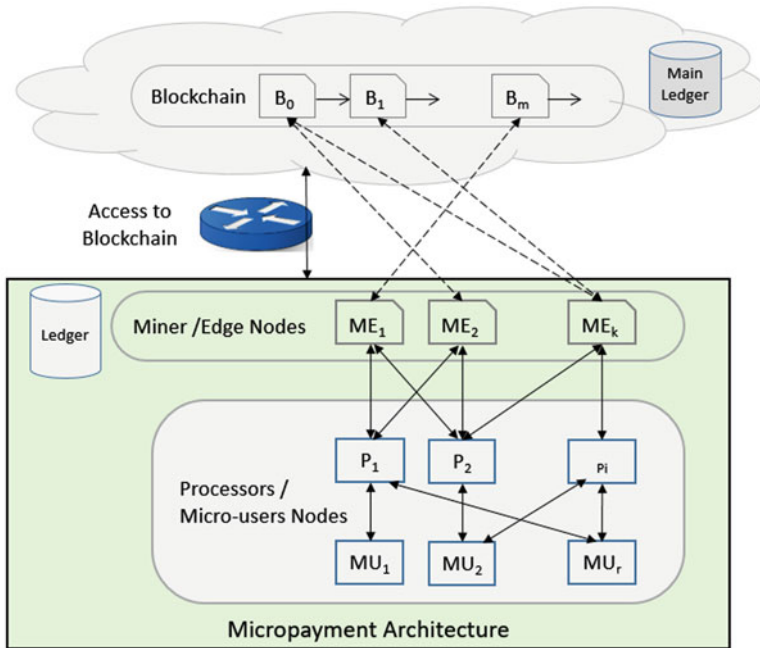


Fig. 6.9 Edge-/fog-enabled blockchain architecture

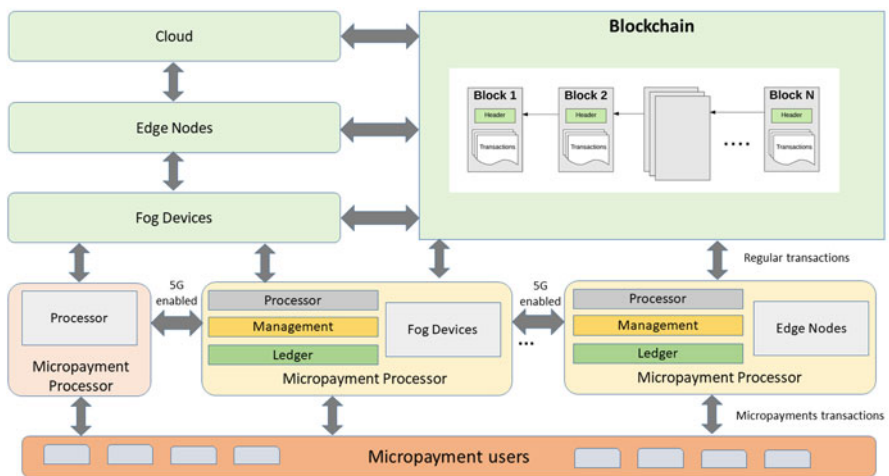


Fig. 6.10 A generic architecture of micropayment system throughout the integration of fog/edge with blockchain



## 6.6 Conclusion and Future Work

This chapter presented challenges as well as opportunities that might arise from the integration of disruptive technologies such as fog computing, blockchain, and IoT as well as 5G and 6G for micropayments solutions. Fog computing improves scalability, latency, and throughput compared to cloud environment. By integrating fog computing with blockchain, many advantages in terms of security and cost can be obtained. Further integration of advanced technologies like artificial intelligence and big data to blockchain and fog computing can support many business verticals. Micropayments are adopted into a large number of applications. However, individually processing micropayments will result in higher transaction fees. In some cases, transaction fee can exceed the payment value. Due to this reason, traditional cryptocurrency blockchain like Bitcoins is inappropriate for micropayment transactions. This chapter also explained the benefits of integrating modern technologies (fog computing, blockchain, 6G, and IoT) to solve the problem of micropayment systems. This is achieved by utilizing the capabilities of each technology (e.g., edge computing) to provide ad hoc but high computational power as well as reduced latency in transaction processing. The chapter also highlighted the various relationships among these technologies and surveyed the most relevant work in order to analyze how the use of these disruptive technologies could potentially improve the micropayment system functionality. The chapter concluded by presenting a generic solution proposal to the problem of micropayments by integrating fog computing capabilities, blockchain, and edge computing to provide a practical payment setup. Such model will be technically pursued in another work as it needs more involved elaboration, implementation, and performance evaluation.

**Acknowledgments** We would like to thank the book editors for giving us the opportunity to contribute to this timely and useful book with an interesting topic.

## References

1. Reyna, A., Martín, C., Chen, J., Soler, E., & Díaz, M. (2018). On blockchain and its integration with IoT. Challenges and opportunities. *Future Generation Computer Systems*, 88(2018), 173–190. <https://doi.org/10.1016/j.future.2018.05.046>
2. Bouachir, O., Aloqaily, M., Tesng, L., & Boukerche, A. (2020). Blockchain and fog computing for cyber-physical systems: Case of smart industry. *arXiv*, 53, 36.
3. Singh, P., Nayyar, A., Kaur, A., & Ghosh, U. (2020). Blockchain and fog based architecture for internet of everything in smart cities. *Futur*, 12(4), 1–12. <https://doi.org/10.3390/FI12040061>
4. Jang, B., Guejong, S. H., Jeong, J., & Sangmin, J. Fog computing architecture based blockchain for industrial IoT. *International Conference on Computational Science Cham*, 11538, 593–606. <https://doi.org/10.1007/978-3-030-22744-9>
5. Baniata, H., & Kertesz, A. (2020). A survey on blockchain-fog integration approaches. *IEEE Access*, 8, 102657–102668. <https://doi.org/10.1109/ACCESS.2020.2999213>

6. Memon, R. A., Li, J. P., Ahmed, J., Nazeer, M. I., Ismail, M., & Ali, K. (2020). Cloud-based vs. blockchain-based IoT: A comparative survey and way forward. *Frontiers of Information Technology and Electronic Engineering*, 21(4), 563–586. <https://doi.org/10.1631/FITTEE.1800343>
7. Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. [www.Bitcoin.Org](http://www.Bitcoin.Org). <https://doi.org/10.1007/s10838-008-9062-0>
8. Fan, Y., Wang, L., Wu, W., & Du, D. (2021). Cloud/edge computing resource allocation and pricing for mobile blockchain: An iterative greedy and search approach. *IEEE Transactions on Computational Social Systems*, 1–13. <https://doi.org/10.1109/TCSS.2021.3049152>
9. Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., & Njilla, L. (2017). ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. *Proceeding – 2017 17th IEEE/ACM International Symposium on Cluster Cloud and Grid Computing (CCGRID) 2017*, pp. 468–477. <https://doi.org/10.1109/CCGRID.2017.8>
10. Liang, X., Shetty, S. S., Tosh, D., Njilla, L., Kamhoua, C. A., & Kwiat, K. (2019). ProvChain: Blockchain-based cloud data provenance. *Blockchain for Distributed Systems Security*, 69, 67–94.
11. Nawaz, A., et al. (2020). Edge computing to secure iot data ownership and trade with the ethereum blockchain. *Sensors (Switzerland)*, 20(14), 1–17. <https://doi.org/10.3390/s20143965>
12. Li, J., Wu, J., & Chen, L. (2018). Block-secure: Blockchain based scheme for secure P2P cloud storage. *Information Sciences (NY)*, 465, 219–231. <https://doi.org/10.1016/j.ins.2018.06.071>
13. Juneja, A., & Marefat, M. (2018). Leveraging blockchain for retraining deep learning architecture in patient-specific arrhythmia classification. *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI) 2018*, 2018, 393–397. <https://doi.org/10.1109/BHI.2018.8333451>
14. Shafagh, H., Burkhalter, L., Hithnawi, A., & Duquennoy, S. (2017). Towards blockchain-based auditable storage and sharing of iot data. *CCSW 2017 – Proceedings of the 2017 Cloud Computing Security Workshop, co-located with CCS 2017, 2017*, 45–50. <https://doi.org/10.1145/3140649.3140656>
15. Yuan, Y., & Wang, F. Y. (2018). Blockchain and cryptocurrencies: Model, techniques, and applications. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 48(9), 1421–1428. <https://doi.org/10.1109/TSMC.2018.2854904>
16. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., & Wang, F. Y. (2019). Blockchain-enabled smart contracts: Architecture, applications, and future trends. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 49(11), 2266–2277. <https://doi.org/10.1109/TSMC.2019.2895123>
17. Tian, Z., Li, M., Qiu, M., Sun, Y., & Su, S. (2019). Block-DEF: A secure digital evidence framework using blockchain. *Information Sciences*, 491, 151–165. <https://doi.org/10.1016/j.ins.2019.04.011>
18. Huang, B., Liu, Z., Chen, J., Liu, A., Liu, Q., & He, Q. (2017). Behavior pattern clustering in blockchain networks. *Multimedia Tools and Applications*, 76(19), 20099–20110. <https://doi.org/10.1007/s11042-017-4396-4>
19. Deep, G., Mohana, R., Nayyar, A., Sanjeevikumar, P., & Hossain, E. (2019). Authentication protocol for cloud databases using blockchain mechanism. *Sensors (Switzerland)*, 19(20), 1–13. <https://doi.org/10.3390/s19204444>
20. Puthal, D., & Mohanty, S. P. (2019). Proof of authentication: IoT-friendly Blockchains. *IEEE Potentials*, 38(1), 26–29. <https://doi.org/10.1109/MPOT.2018.2850541>
21. George, G., & Sankaranarayanan, S. (2019). Light weight cryptographic solutions for fog based blockchain. *6th IEEE International Conference on Smart Structures and Systems (ICSSS) 2019*. <https://doi.org/10.1109/ICSSS.2019.8882870>
22. Carbone, A., Daveev, D., Mitreski, K., Kocarev, L., & Stankovski, V. (2018). Blockchain based distributed cloud fog platform for IoT Supply Chain Management, 51–58. <https://doi.org/10.15224/978-1-63248-144-3-37>

23. Wu, D., & Ansari, N. (2020). A cooperative computing strategy for blockchain-secured fog computing. *IEEE Internet of Things Journal*, 7(7), 6603–6609. <https://doi.org/10.1109/JIOT.2020.2974231>
24. Rahmani, A. M., et al. (2018). Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. *Future Generation Computer Systems*, 78, 641–658. <https://doi.org/10.1016/j.future.2017.02.014>
25. Tuli, S., Mahmud, R., Tuli, S., & Buyya, R. (2018). *FogBus: A blockchain-based lightweight framework for edge and fog computing*. [Online]. Available: <http://arxiv.org/abs/1811.11978>.
26. Lundqvist, T., De Blanche, A., & Andersson, H. R. H. Thing-to-thing electricity micro payments using blockchain technology. *GIoTS 2017 – Global. Internet Things Summit*, Proceeding, 2017. <https://doi.org/10.1109/GIOTS.2017.8016254>.
27. Robert, J., Kubler, S., & Ghatpande, S. (2020). Enhanced lightning network (off-chain)-based micropayment in IoT ecosystems. *Future Generation Computer Systems*, 112, 283–296. <https://doi.org/10.1016/j.future.2020.05.033>
28. Xu, Q., Li, M., Huang, X., Xue, N., Zhang, J., & Sheng, A. A blockchain based micro payment system for smart devices. *Signature*, 256(4936), 115.
29. Pouraghily, A., & Wolf, T. (2019). A lightweight payment verification protocol for blockchain transactions on IoT devices. *2019 International Conference on Computing, Networking and Communications (ICNC), 2019*, 617–623. <https://doi.org/10.1109/ICNC.2019.8685545>
30. Hao, Z., Ji, R., & Li, Q. (2018). FastPay: A secure fast payment method for edge-IoT platforms using blockchain. *Proceeding – 2018 The Third ACM/IEEE Symposium on Edge Computing (SEC 2018)*, pp. 410–415. <https://doi.org/10.1109/SEC.2018.00055>.
31. Heilman, E., AlShenibr, L., Baldimtsi, F., Scafuro, A., & Goldberg, S. (2017). TumbleBit: An untrusted bitcoin-compatible anonymous payment hub. <https://doi.org/10.14722/ndss.2017.23086>
32. Poon, T., & Dryja, J. (2016). The Bitcoin Lightning Network: Scalable off-chain instant payments. <https://doi.org/10.3758/BF03205969>
33. Green, M., & Miers, I. (2017). Bolt: Anonymous payment channels for decentralized currencies. *Proceedings of the ACM Conference on Computer and Communications Security, 2017*, 473–489. <https://doi.org/10.1145/3133956.3134093>
34. Lee, S., & Kim, H. (2020). On the robustness of lightning network in bitcoin. *Pervasive and Mobile Computing*, 61, 101108. <https://doi.org/10.1016/j.pmcj.2019.101108>
35. Yutao, J., Wang, P., Niyato, D., & Xiong, Z. (2018). Social welfare maximization auction in edge computing resource allocation for mobile blockchain. In *2018 IEEE international conference on communications (ICC)*, pp. 1–6. IEEE.
36. Wu, Y., Wang, Z., Ma, Y., & Leung, V. C. M. (2021). Deep reinforcement learning for blockchain in industrial IoT: A survey. *Computer Networks*, 108004.
37. Wu, Y., Dai, H.-N., & Wang, H. (2020). Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. *IEEE Internet of Things Journal*.
38. Wu, Y. (2020). Cloud-edge orchestration for the internet-of-things: Architecture and AI-powered data processing. *IEEE Internet of Things Journal*.

# Chapter 7

## Medical Prescription Traceability Using Blockchain-Based Decentralized Application



V. Kakulapati and Parimi Shiva Kalyan

### 7.1 Introduction

Medicare is changing its perspective on life, is becoming increasingly intelligent, and reliant on 6G communication systems. “5G facilitates rapid and convenient coordination among devices, systems, and physicians.” It delivers a better clinical environment; however, interoperability can also write a better Medicare medicine prescription [1]. Fast altitude and latency wireless communication allows 5G to enhance access to medical care. The patient’s background of prescriptions, such as essential medication details, is accessed automatically with such a distributed ledger ePrescription approach. All this reduces the chance of the inappropriate method, prescription, and sometimes medication being prescribed to a patient. The readily accessible electronic record can prevent prescription inconsistencies, eliminate inaccuracies to humans, or misconceptions. Even though the patient’s previous prescription is available to physicians and medical practitioners, it reduces the risk of mismedication being dispensed and provided to the patient [2]. Because all this is implementing blockchain, it can be assured that data are not altered or distorted and that everyone can be confident.

Medical prescription traceability has played a significant role since the origins of the pandemic when the required medication used to change hands a lot to reach the person in need of the treatment, and that is when the drugs were abused and sold for much higher rates and their concentrations were altered. Prescription as a term refers to “an authorized drug/medicine usage given/advised by a medical practitioner to the patient.” Prescription traceability will provide a footprint in the ecosystem of

---

V. Kakulapati (✉)  
Sreenidhi Institute of Science and Technology, Hyderabad, Telangana, India

P. S. Kalyan  
Accenture, Hyderabad, Telangana, India

pharmaceutical drugs and make sure that the passage or the supply chain through which the handover of the medicines/drugs happens is safe and verified by the central body in authority for a given geographical location.

Transactions that occur in products or consignments in real-time monitoring from the basis and during the supply chain are known as traceability.

### ***7.1.1 The Advantages of Traceability***

The capability of traceability is to trace the supply chain status and trace back the history of drugs in the location under consideration. The health care supply chain is monitored by a universal standardized identification organization from producer to patient. The pharmaceutical firm has a way of ensuring operative and quick assessment of items from particular batches or quantities of prescriptions from the marketplace by using the traceability procedure. This procedure can be accomplished in a minimum amount of time to prevent treatment errors and provide the importance of a speedy inventory, which is a significant challenge as all the supply chain inadequacies have to be analyzed.

Using blockchain technology, there are two main principles, confidentiality and traceability, which take care of the conservative trust issues on all open, merged, and societal levels. These features are not appropriate to give a complete result that leads to blockchain associated with robust encryption algorithms [3]. Blockchain's association with cryptography algorithms makes it confidential that traceability, security, and administrative work in various businesses and the medical and supply chain are central to analytical solutions. Once blockchain has recorded data, it cannot be transformed or cancelled entirely. This perpetuity and recognizability of data are a basic necessity for any health care organization. Therefore, when blockchain is used, it creates the sense of imminent events.

The following list of issues is solving by blockchain technology in the health care domain.

- Reliable storage and truthful fortification
- Protection of privacy-preserving data
- Distribution of data
- Detection and liability of data

These issues can be tended to separately with the correct utilization of cryptography and privacy-preserving methods and can lead blockchain technology to being a confidential distributed ledger. The distributed nature of blockchain technology [4] generates one ecosystem of patient information that can be rapidly and effectively referenced by specialists, clinics, drug specialists, and anybody involved in diagnosis. In this way, blockchain technology can lead to early treatment and personalized protection.

Data can be protected in insightful medical systems with 5G technology blockchain to prevent misinterpretation. Blockchain compatibility with 5G

technologies will improve existing health care to provide greater reliability. Blockchain was known as one of the most significant communication devices for 6G networks [4–6], and the leveraging of 6G mobiles is essential. Intelligent 6G health care had to provide a different approach to addressing issues in the 5G technology. The more comprehensive and pervasive incorporation of blockchains in wireless communication will boost primary health care performance and increase efficiency with greater decentralization, safety, and privacy. The problem of privacy is one of these technological problems. Also, the integrity of health care data is feasible because of the infallibility of blockchains. Patient confidentiality and reliable data processing without centralized controlled third-party companies can be achieved with blockchain technology.

By implementing traceability, we can cover a broad range of domains in the pharmaceutical market using various methods, including the two different ends of the governing body, i.e., the top-most authorities and the bottom-level organizations/dealers. This blockchain-based framework would be responsible for maintaining a ledger type of structure that would store each aspect of transactions made to/for the drugs in the chain and record the process path and workflow of the drugs present in the supply chain.

## 7.2 Related Work

Faster than 4G, 5G was anticipated to be a facilitator for all other technologies, e.g., the internet of everything (IoE), convergence in industry, intelligent transport, and remote health care, by offering ultra-high reliability, latency as low as 1 ms, improved data and network capacity [7], and many more.

Because of the significant utilization of cryptocurrency in any process, a protected Medicare framework guarantees sensitive data in the cloud to the patient [8]. Numerous scholars discussed the different types of models for incorporating cloud-based protection mechanisms [9, 10] and then integrated cryptocurrency into medicine to migrate cloud data along with key management issues. Decentralized blockchain technology has replaced the centralized control structure problems in health care for safety. The utilization and performance of blockchain in decentralized health care have managed every track or file system properly [11]. A medical history dissemination network with a blockchain-integrated system for transitioning costs and trade.

Any set of nodes (blocks) has been adding malicious nodes and malicious miners to execute malicious operations in the network owing to a grey attack. E-health care providers offer this information to each miner on the internet if a pharmacy or physician gives a patient a list of medications.

This decentralized activity can be amplifying as actions may be verified utilizing blockchain-based digital currencies. Such concepts can be effectively incorporated, leading to several benefits. During the last decade, advanced analytics emerged and have provided people with tremendous business insights by analyzing massive

datasets. The data created and processed in the cloud can be examined side-by-side to give the user valuable information to improve productivity and benefit.

### ***7.2.1 Management of Prescriptions***

Appropriate administration of prescriptions is significant to provide the best medical provisions. Nowadays, substantial scope issues such as the opioid crisis have developed because of mismanagement of the prescription [12]. Many investigations developed to eliminate the barriers to appropriate prescription administration. All operations are all securely accumulated by blockchain technology known as BlockMedx [13], which utilizes an Ethereum-based method to administer the prescription processes securely. Once the physician issues a prescription to a patient, then the specified pharmacist can check the prescription by blockchain technology before supplying the medicines. For tracking prescriptions, intelligent contract applications [14] are used in addition to Ethereum, and facilitate separate gateways for physicians and pharmacists interested in the prescription procedure. The delivery of drugs to patients is restructured by ScriptDrop [15] and releasing patients from having to show their prescriptions at the pharmacy. ScriptDrop also traces the usage of drugs using computer-generated associates, and they utilize blockchain technology to trace the delivery information. ScalaMed [16] operates a patient-centric model of blockchain-based resolution for prescription tracing and stalking of all prescriptions, which designates the digitization of prescription for solving prescription mismanagement issues. Many blockchain technology-based solutions have been developed for prescription administration; however, a centralized conservative organization can facilitate a solution for firms where few parties have to participate.

### ***7.2.2 Traceability Using Ledger Systems***

Ledger systems have been the primary tracing tools since the increase in alterations and manipulations in the supply chain for products, especially medicines [17]. These give rise to various ways of counteracting the fake drugs introduced into the supply chain by dealers who gain enormous profits at the cost of human lives. Using the ledger system, one would maintain a log record of the timestamps when the drug moved and from where it moved. From the source to the target destination, everything is recorded in the ledgers present at distinct locations, which would be later compared to check for any manipulations.

### **7.2.3 Prescription Traceability**

Prescription traceability plays a significant role in tracking the status of a patient's record once he/she visits the hospital until the end of the chain when the patient receives the prescribed drugs. This form of traceability helps to find the loopholes, if any, existing within the system. This offers a whole new dimension to the pharmaceutical drugs in the supply system. As the traceability increases, it proposes a new unexplored solution to the complex problem of black-marketing and illegal use of drugs and stops drug abuse.

The permission of data on blockchain [18] defines a permission-controlled smart contract that defines the permission of content management, the consent of intelligent contract management, and permission to join the blockchain. To join the traceability platform, a company needs to send a request off-line to get approval to join the blockchain network. The traceability company updates the permission-controlled smart contracts before the company can join the blockchain network and synchronizes historical transactions.

## **7.3 Proposed Method**

The proposed system uses a blockchain-based framework to trace the movement of the medicines prescribed in the supply chain ranging from various levels, including different parties such as pharmacy, hospital, patient, and the governing health body.

This application uses a decentralized blockchain-based ledger system, where each transaction has a signature of one of the parties mentioned earlier. These parties verify the medical prescription produced to them or produced by them to the following parties by their transaction signature, encrypted in a hex code, which is further confirmed by the next level of authority in charge (Fig. 7.1).

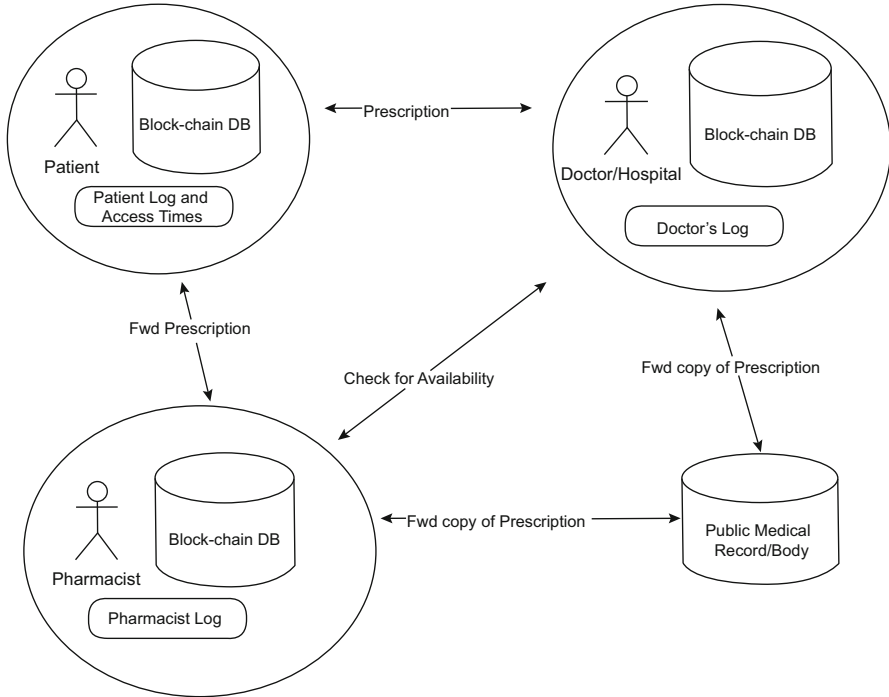
This application programs in Solidity the smart contracts in the "Contracts" folder of the blockchain files, written in Solidity pragma header files. These smart contracts have the following actors at the code level: medical prescription owner, medical prescription moderator, transfer responsibility, previous counter party, counter party, initiating counter party, and state.

## **7.4 Methodology**

### **7.4.1 Blockchain Technology**

Blockchain technology [19] satisfies elementary prerequisites for universal traceability:





**Fig. 7.1** The framework of the proposed work

- Impossible to use data
- Accessible anywhere and adaptable
- Simply familiarized and implemented throughout the world

Limitations:

- All transactions should be stored in blockchain with a bit of data and not blocks with more features
  - Blockchain technology can process only a few transactions per second
  - Each transaction cost is dependent on the lumps of that type of network, and the instructions agree with them
  - Hacking is highly improbable

The proposed blockchain decentralized application works on an online compiler where the contracts are written in Solidity and executed, which gives out the transaction details; each transaction done has a “Gas Price.” This transaction via Gas Price is recorded for further details (Fig. 7.2).

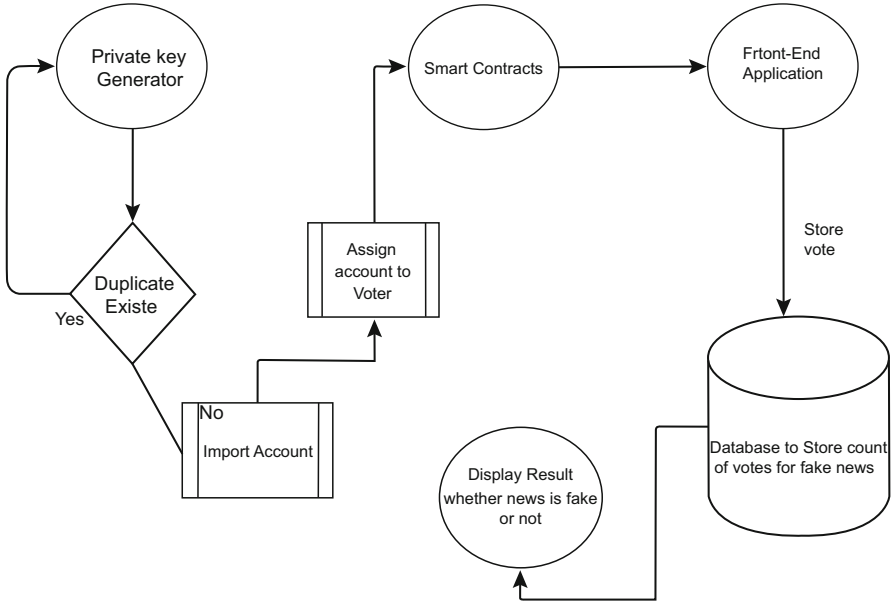


Fig. 7.2 The workflow of the proposed work

### 7.4.2 Traceability

A electronic ledger containing an uninterruptedly increasing order of blocks is known as blockchain. Each block contains a complete transaction list of records. In this process, each block has a parent block, and the first block in the chain is called the genesis block. The genesis block hash code complements the next block’s title, then its hash code is calculated by the hash of the genesis block and the transactions of the block equally. The second block hash code becomes the header of the third, and so on. In this way, the blocks are associated with each other along with a timestamp. The association can be pursued in reverse to the genesis block [20, 21]. This characteristic of blockchain facilitates data origin to retain and chronicle the traceability of events and may also support exploring backward in the chain.

### 7.4.3 Key Generation Functionality

The keys for account creation on MetaMask are generated by Ganache. Every instance of Ganache from npm has a mnemonic that can generate a set of private keys and get those private keys for an individual to vote. Functions such as mnemonicToSeed (),

`getPublicKey ()` are used to obtain the public string and private string in the key generation segment.

```
'address': node.getWallet (). getAddressString ();
'privateKey': node.getWallet (). getPrivateKeyString ();
'publicKey' : node.getWallet (). getPublicKeyString ();
To include Ethereum wallet we use
const wallet = require ('ethereumjs-wallet ');
const privateKey = addrnode.getWallet (). getPrivateKey ();
```

The `getWallet ()` function is primarily responsible for returning the wallet, which is associated with the response obtained from the request.

The `getPrivateKeyString ()` function returns a private key string. This string cannot be “Null.” The private key varies accordingly, with the wallet importing. In this case, the wallet is Ganache, which provides Ethereum transactions over a gas price.

The `getPublicKeyString ()` function returns a public key string. This combination of the public key and private key obtained from `getPrivateKeyString ()` and `getPublicKeyString ()` is used to import accounts over MetaMask.

#### ***7.4.4 Implementing Smart Contracts***

Self-executing contracts are mostly simple computer codes executed and supervised by various individuals/organizations present in the ring. They remove the necessity for a middleman in the blockchain transactions, thereby ensuring transparency and recording of each transaction happening over the network. In this work, implementing a smart contract can record the number of votes in favor or against a specific set of fake news.

The significant applications of society are disrupted by blockchain technology with use of the voting process. This process can ensure that only registered participants can vote and only reliable votes are counted. The identity of participants can be verified by the distributed ledger and smart contracts to preserve the tracking of each polled vote. For a fair election, the foremost step is to implement the public-accessible ledger [23]. Democracy Earth [24] and Follow My Vote [25] are two start-up organizations predicted to disturb the consensus by developing a voting methodology for the government based on blocking.

The voting system is implemented by defining a structure in the Solidity file. There are two basic functions present in the Elections Solidity file, responsible for the voting process for fake news: `addFakeNewsSet ()` and `vote ()`. Other functions present in Smart Contracts include `keyGeneration ()`, `keyMatch ()`, and `importAccount ()`.

### 7.4.5 *Function addFakeNewsSet ()*

The function `addFakeNewsSet ()` takes a string array-type parameter, which is a set of collections of fake news spread across social media gathered from different resources and groups and whose authenticity needs to be sent for testing.

The function `addFakeNewsSet ()` adds data, i.e., collects fake news gathered from various sources in an array into the election system structure to take place. The structure for adding fake news sets consists of an ID, count of votes, and the fake news to display for the voting system, thereby creating a back-end for the voting system to bring it to reality. Every time the fake news set is updated in the array, the counter value is incremented to keep a count and display how much news turned out to be fake from the total number of news presented.

### 7.4.6 *Function Vote ()*

The function `vote ()` takes the unique identity number given to fake news as a parameter and counts the number of votes cast for a particular identity number. The identity number is provided in a back-end process, whereas only the fake news is visible to the voter to choose and vote on at the front-end. This ID number counter increases every time a vote is casted for a specific set of fake news. The vote counter keeps track of the number of votes received for a piece of specific fake news and increases the incoming votes for it. The private keys used in this process to vote are also stored in the blockchain network to keep track that no voter can vote multiple times for the same fake news in the same session.

```
function vote (uint id)
{
//vote recording
}
```

### 7.4.7 *Importing Accounts Using Private Keys*

Use the generated private keys via `getPrivateKeyString ()` method, the account on MetaMask is imported via the private key and the public string, i.e., the seed phrase. These imported accounts are initialized by Ethereum so that the individual can vote.

The imported accounts in a session are assigned to distinct voters to continue the voting process to vote for the authenticity of the news provided in the session. The results are collected and shown to the general public with maximum votes for news results in more “fake” content, and the least votes mean it is “less fake” than the other news in the set.

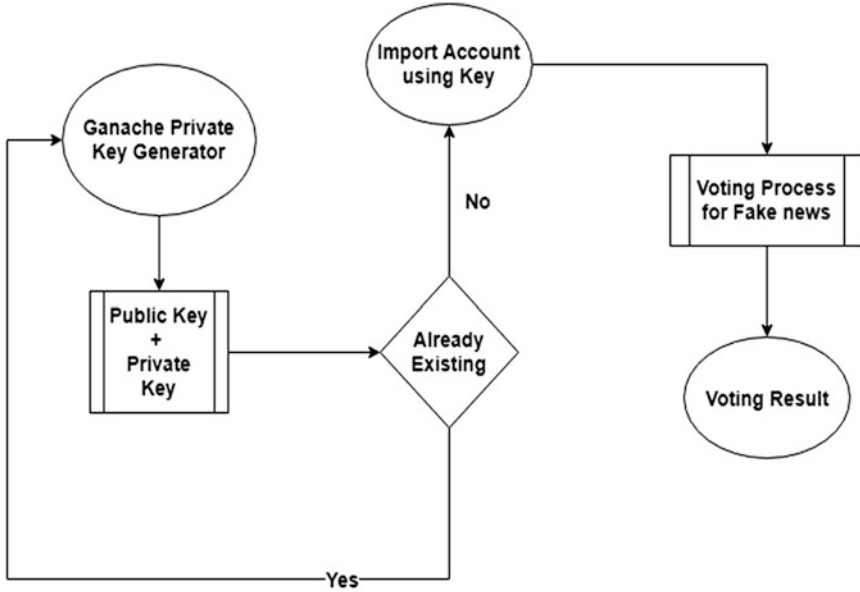


Fig. 7.3 Importing account for the voting process

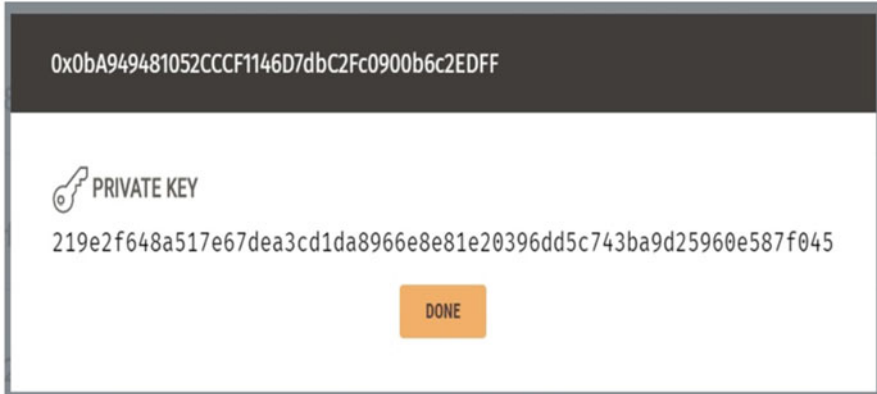
Figure 7.3 depicts the framework involved in importing an account for the user/the individual who would vote for the authenticity of fake news spread on social media.

### 7.4.8 Front-End Voting System

The front-end, or the user end, comprises an HTML with JavaScript-validated page responsible for showing the use case of the application, i.e., displaying the fake news via a drop-down menu, and the user gets to select one of them that is not valid. The front end is responsible for collecting the user input/choice and reverting the same in the database, which in-turn increases the counter. This collected information is later shown with each item of fake news, and opposite it, the number of votes it received on the “fake scale.”

## 7.5 Performance Factors and Results

Figure 7.4 depicts the private key generated from the PKG (private key generator), which generates a private key so that the accounts to vote for the authenticity of the news can be imported using the private key and the voting process can continue.



**Fig. 7.4** Generating private keys to import vote accounts

Figure 7.5 depicts the npm terminal running the web3 lite-server. The lite server is a CLI that has a static HTTP server, which acts as a backbone for single page applications. The lite-server package can be installed with the following Figs. 7.5 and 7.6.

MedicalPrescriptionOwner = Patient for whom the medical prescription is written.  
 MedicalPrescriptionModerator = Various doctors who modify or write new prescriptions to the patient in the cycle recorded.  
 PreviousCounterparty = The previous doctor who wrote the medical prescription/the doctor whom the patient visited earlier.  
 InitiatingCounterParty = Hospital which initiated the prescription for the patient.  
 State = How many hands did the prescription go through, i.e., the number of doctors who prescribed for the same patient.

When the contract is executed, it requires two addresses, Medical Prescription Owner and Medical Prescription Moderator. The address has to be written from the record and deployed to see/track the provenance. A wallet address is generated from the hash function using the scroll menu. Any hash value can be selected that would be stored in the ledger.

A cost is deducted from the account every time an individual enters the details or inputs a transaction/data into the medical prescription metadata, i.e., a change in the metadata recorded. Keys are generated by the hash function, which are then used to input details about the ledger's medical prescription provenance. These keys can be used later for security purposes. Each key is different and unique, hence acting as an authentication factor—all functions and values after a single entry in the ledger depict a medical prescription (Fig. 7.7).

```

C:\Users\user\Desktop\election>npm run dev

> election@1.0.0 dev C:\Users\user\Desktop\election
> lite-server

** browser-sync config **
{ injectChanges: false,
  files: [ './**/*.html,htm,css,js' ],
  watchOptions: { ignored: 'node_modules' },
  server:
    { baseDir: [ './src', './build/contracts' ],
      middleware: [ [Function], [Function] ] } }
[Browsersync] Access URLs:
-----
    Local: http://localhost:3000
    External: http://192.168.1.6:3000
-----
    UI: http://localhost:3001
    UI External: http://localhost:3001
-----
[Browsersync] Serving files from: ./src
[Browsersync] Serving files from: ./build/contracts
[Browsersync] Watching files...

```

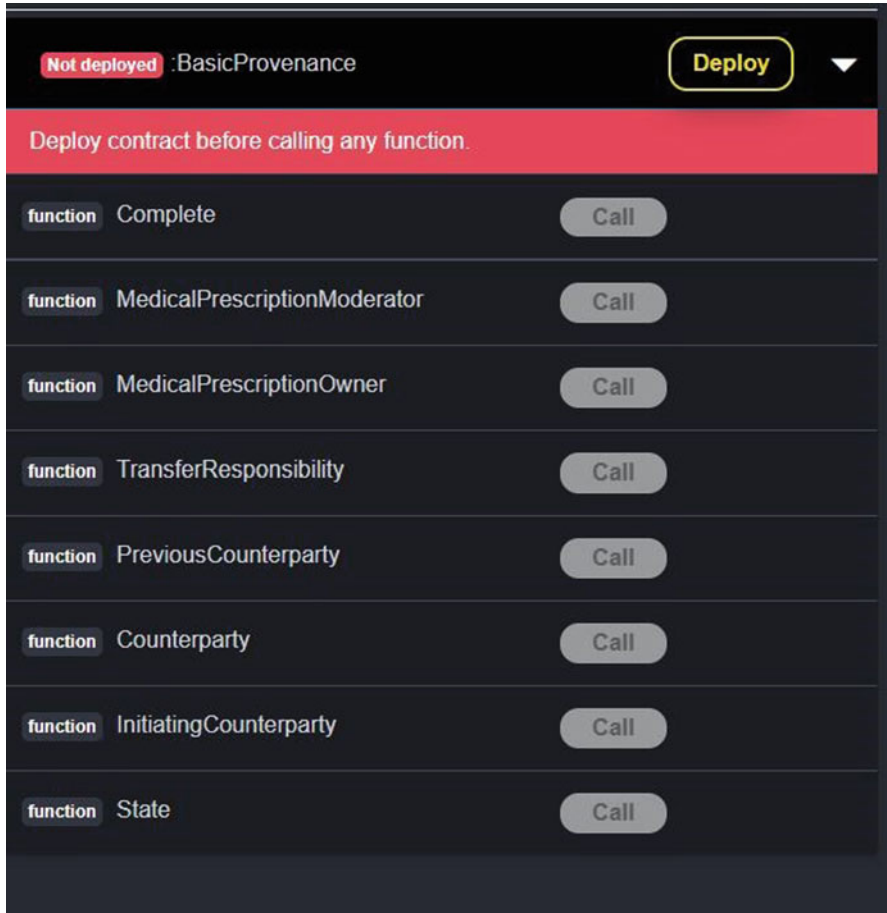
Fig. 7.5 Running lite-server for front-end application

Here, the state represents the number of times the patient's ledger for provenance is modified, i.e., the number of times the medical prescription is generated. In this case, it is one (Figs. 7.8 and 7.9).

```
npm install -save-dev lite-server
```

Figure 7.10 depicts the application where the voter votes for the fake news out of the given options, as seen in the image above. The user gets a chance to use his one vote to pick out the fake news from the given set of fake news. Here, it is “Rs.10 Rupee coin banned by Indian Government”.

Figure 7.11 maps the result gathered after the voter/individual votes for the fake news. It shows the number of votes gained by particular news by different voters. The more the votes, the higher score it has on the “Fake Scale,” i.e., the more votes, the further away it is from the truth. The account number of the voter is also shown below for reference.



**Fig. 7.6** The functions used in medical prescription

Figure 7.12 depicts the drastic increase that can be observed in the general public using the internet over time. As the smartphone industry occupies the top place in the electronics section, the world wide web has also hit the top rank. From the figure it is concluded that fake news has also increased to the top levels after the general public was introduced to different sources on the internet, and hence the massive variety of fake news being spread among the network with the least technology present to avoid the circulation of such fake news.

It can be observed that not only the performance but other constraints such as security and ledger are added features in the blockchain-based voting system to curb fake news compared with the overall voting system. It can also be observed from the above figure that the interest of the general public in the mainstream voting system has decreased because of tampering and for many other reasons, whereas



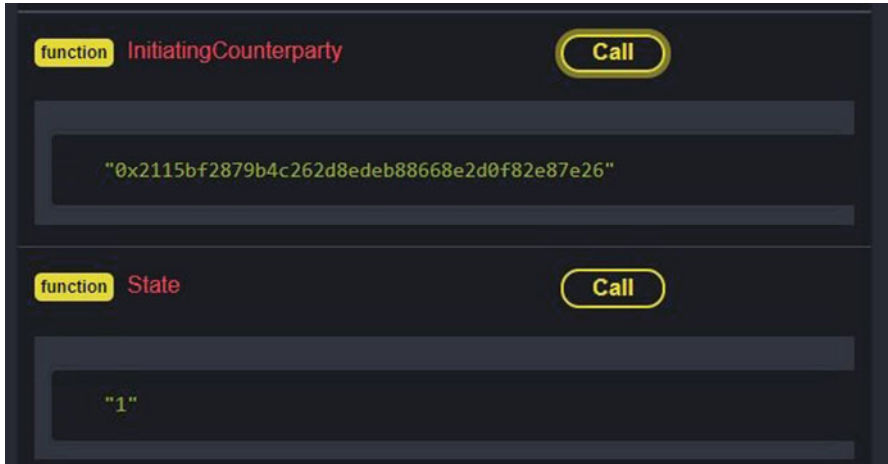


Fig. 7.7 Number of times the medical prescription is generated

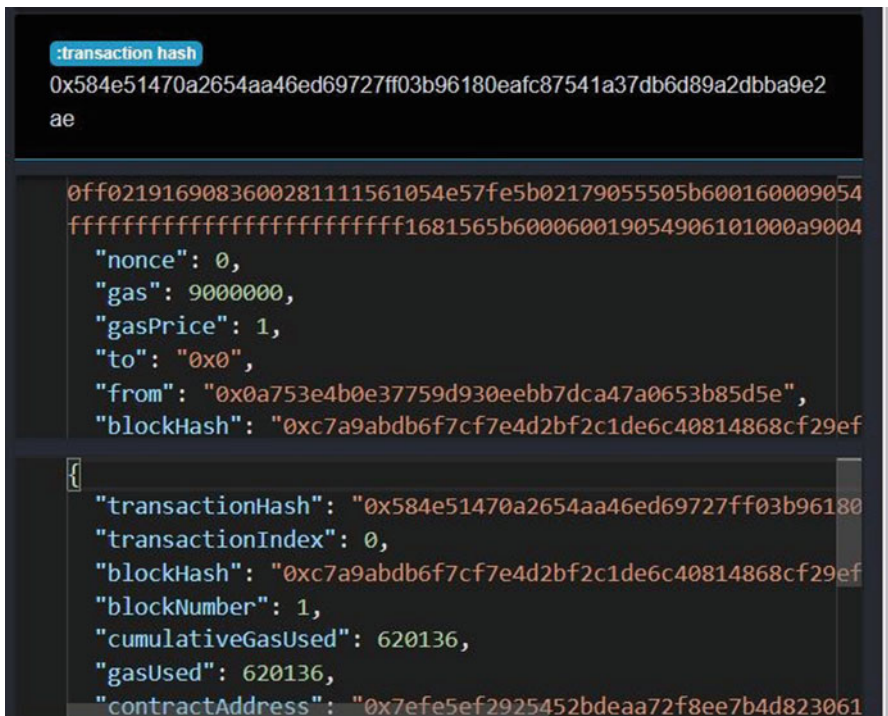


Fig. 7.8 Cost of the transactions with hash keys used to enter details

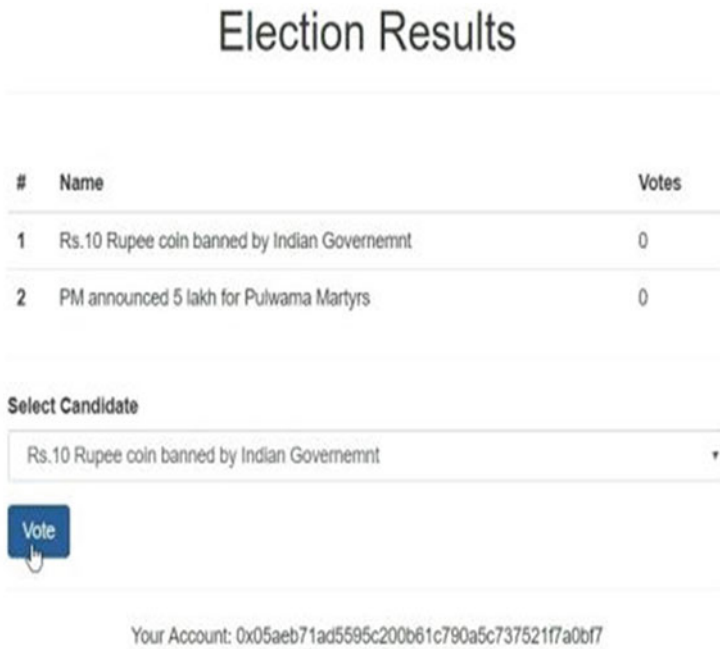


Fig. 7.9 Voting user interface/blockchain implementation



Fig. 7.10 Post-voting scenario

the emerging technology “blockchain” can be seen by many eyes. In this system, we can say that the outcome/result can be seen as the general interest to curb fake news increases as blockchain-based voting system was introduced.

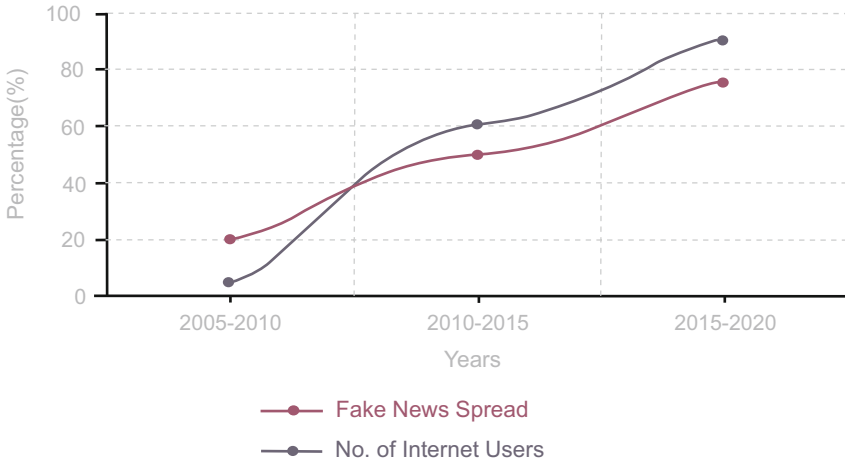


Fig. 7.11 Fake news vs number of internet users

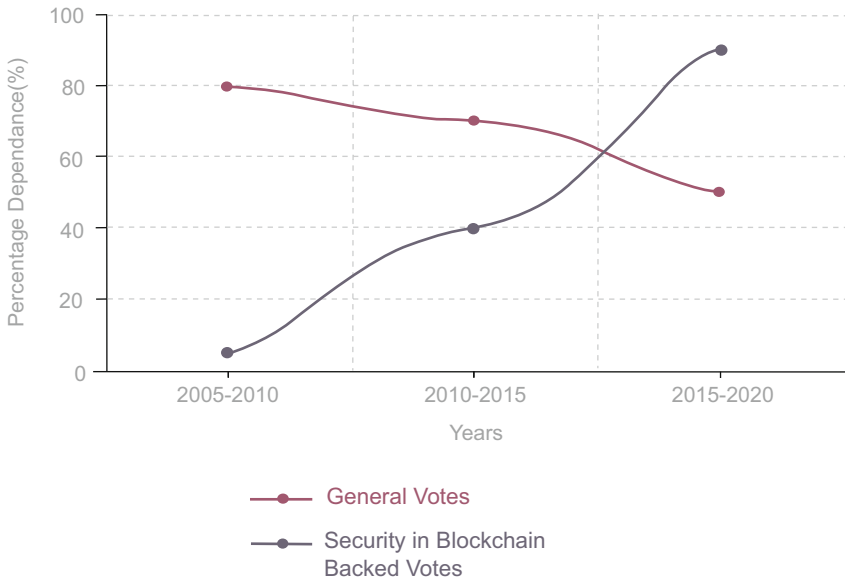


Fig. 7.12 Dependency on the quality factor for blockchain votes

## 7.6 Conclusion

As the quantity of fake news spread on various social platforms such as WhatsApp and Facebook has increased, there has been an immense need to stop the news from getting into the beliefs of the general public's belief. The blockchain-based voting

system acts as a perfect solution to these problems by creating a secure yet efficient and transparent voting system where there is a distribution of power, decreasing the chances of anarchy, and ensuring that fake news is identified and voted on by voters to curb it. A secure blockchain-based wallet can give credits to the user according to the previous votes that an individual made against fake news. The individual awarded with credits can link his account with a blockchain-based voting system and vote for the authenticity of the news with the credits gifted from the previous transaction, hence generating enthusiasm among the public regarding the curb on fake news.

## 7.7 Future Scope

Vendor lock-in problems in health care can be minimized by utilizing decentralized blockchain services. The evaluation of the system by health professionals involved in their use in clinical practice proposed improvements in the potential weak points detected to be enabled. In the future, the network size can be increased and the performance and feasibility of blockchain-based decentralized prescription traceability can be checked by applying it in real time.

## References

1. [https://www.lightreading.com/partner-perspectives-\(sponsored-content\)/standards-are-right-prescription-for-5g-healthcare-applications-/a/d-id/768310](https://www.lightreading.com/partner-perspectives-(sponsored-content)/standards-are-right-prescription-for-5g-healthcare-applications-/a/d-id/768310).
2. <https://www.kaleido.io/blockchain-blog/just-what-the-doctor-ordered-a-blockchain-based-prescription-platform#:~:text=With%20a%20blockchain%2Dbased%20ePrescription,including%20critical%20drug%20interaction%20information.&text=This%20further%20reduces%20the%20chances,form%2C%20dosage%20or%20even%20medication>.
3. Goldreich, O., et~al. (1994). Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1), 1–32.
4. <https://builtin.com/blockchain/blockchain-healthcare-applications-companies>.
5. Aazhang, B., et~al. (2019). *Key drivers and research challenges for 6G ubiquitous wireless intelligence (white paper)*, 09 2019. [Online]. Available: <http://jultika.oulu.fi/files/isbn9789526223544.pdf>
6. Chowdhury, M. Z., et.al. (2019). 6G wireless communication systems: applications, requirements, technologies, challenges, and research directions. *arXiv preprint arXiv:1909.11315*.
7. Zhang, Z., et~al. (2019). 6G wireless networks: Vision, requirements, architecture, and key technologies. *IEEE Vehicular Technology Magazine*, 14(3), 28–41.
8. Saad, W., et~al. (2019). *A vision of 6G wireless systems: Applications, trends, technologies, and open research problems*, 10265. <https://doi.org/10.1109/MNET.001.1900287>
9. Chang, C., et~al. (2015). Design and implementation of an IoT access point for smart home. *Applied Science*, 5, 1882–1903.
10. Singhal, A., et~al. (2016). Intelligent accident management system using IoT and cloud computing. In *Proceedings of the 2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, India, 14–16 October (pp. 89–92).
11. Kshetri, N. (2017). Can blockchain strengthen the internet of things? *IT Prof.*, 19, 68–72.

12. Dwivedi, A. D., et-al. (2019). A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors*, 19, 326.
13. Skolnick, P. (2018). The opioid epidemic: Crisis and solutions. *Annual Review of Pharmacology and Toxicology*.
14. <https://blockmedx.com/en/>. Retrieved from BlockMedx – Secure e-prescribing platform.
15. <https://github.com/tylerdiaz/Heisenberg>. Retrieved from solving prescription/pharmaceutical logistics using smart contracts:
16. <https://www.scriptdrop.co/>. Retrieved from prescription delivery in workflow:
17. URL <https://www.scalamed.com/>. Retrieved from ScalaMed:
18. Hanifatunnisa, R., et.al. Blockchain based e-voting recording system design, Published in 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA).
19. Xu, X., et-al. (2018). *Designing blockchain-based applications a case study for imported product traceability*, 399–406. <https://doi.org/10.1016/j.future.2018.10.010>Future Generation Computer Systems 0167-739X/©2018 Elsevier.
20. <https://www.tenthpin.com/insights/blockchain-supporting-traceability-in-life-sciences-myth-or-reality/>
21. Zheng, Z., et-al. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE International Congress on Big Data (BigData Congress)* (pp. 557–564). IEEE.
22. Tama, B. A., et-al. (2017). A critical review of blockchain and its current applications. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)* (pp. 109–113). IEEE.
23. Pilkington, M. (2016). 11 Blockchain technology: Principles and applications. In *Research handbook on digital transformations* (Vol. 225). Edward Elgar Publishing.
24. Jacomet, N. (2017). *Democracy earth, the promise of a safe and independent online voting system*. Available online: <https://medium.com/open-source-politics/democracy-earth-the-promise-of-a-safe-independent-online-voting-system-37366935db5e>. Accessed on 20 Mar 2019.
25. Osgood, R. *The future of democracy: blockchain voting*. Available online: <http://www.cs.tufts.edu/comp/116/archive/fall2016/rosgood.pdf>. Accessed on 23 Mar 2019.

# Chapter 8

## Optical and Wireless Convergence Network Based on Blockchain



Hui Yang

### 8.1 Digital Identity and Anonymous Access Authentication Mechanism

With the rapid development of the Internet of Things and mobile Internet, 5G fronthaul networks need to access more Internet of Things terminal devices to adapt to the continuous growth of network energy efficiency and network capacity [1–4]. The cloud radio access network (C-RAN) can aggregate the computing resources of all base stations and transmit the collected wireless signals through an optical system [5], which is a typical implementation case to achieve the above requirements. However, due to the various types of device access in the Internet of Things [6, 7], trusted device access and the security of the access network have become important issues in the 5G fronthaul network.

In order to cope with the above problems, we innovatively proposed the blockchain-enabled trusted anonymous access (BlockTrust) structure in the 5G fronthaul scenario and introduced an anonymous access recognition strategy in the optical carrier wireless network scenario [8]. BlockTrust can solve the problem of trusted access authentication of devices in the Internet of Things through a tripartite agreement between manufacturers, devices, and operators, effectively reducing network operating costs, and enhancing wireless, optical, and BBU resource optimization.

Three service modes are involved in BlockTrust architecture among manufacturers, blockchain, and controllers. They implement the full process of trusted access identity and access accommodation in networks from three phases, including digital identity service, anonymous access identification, and trusted service provisioning.

---

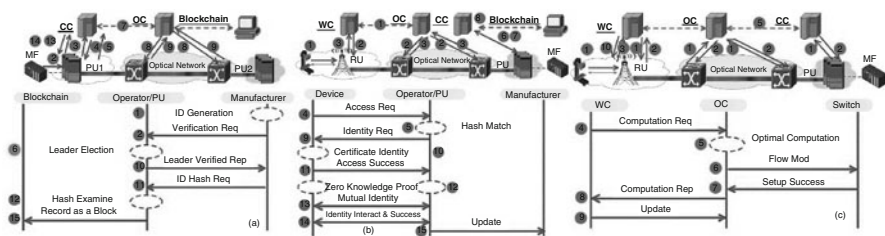
H. Yang (✉)  
Beijing University of Posts and Telecommunications, Beijing, China  
e-mail: [yanghui@bupt.edu.cn](mailto:yanghui@bupt.edu.cn)

### 8.1.1 Digital Identity Service

To guarantee the access credibility in C-RAN, digital identity is a key enabler of the transition from physical world to digital world. A device can access the network credibly by using the digital identity, which is provided to the application system. Figure 8.1a shows interworking procedure of digital identity service.

According to the rules of unified name, manufacturer generates the unique digital identity for a device with a pair of secret keys including company information, abstracted device type, production time, blockchain information, etc. [9]. For anonymity of device, the manufacturer produces a digital signature cryptographically using the private key in secret key pair, and then the public key will be sent to blockchain. The participants in platform should check the identity through voting. Firstly, manufacturer broadcasts the public key and digital certificate in blockchain. CC receives the broadcast information and forwards the message to other participants for voting competition. The participants receive the message and verify the certificate with public key to judge whether it is legal. Then, they turn to active state and return the active signal. The fastest one reaching active state is selected as leader node that sends a ready message and prepares to verify with the digital certificate.

When CC receives the message, the controller sends the message to OC and performs path computation among the participant PUs, and then set up the spectrum path to accommodate the access channel. The manufacturer sends the identity hash of new device to the leader node after verification. If hash already exists, the leader will broadcast the hash to other participants and also send a message to the manufacturer. After receiving the message, the manufacturer regenerates the digital identity for the device and sends the identity hash of new device to the leader again. The participants verify the identity hash of the device. Since the digital identity of the device is generated by manufacturer, the participants will return a message with a successful verification. If the number of verified participants exceeds  $N/2 + 1$ , the leader indicates such identity application successfully and returns the result of voting to manufacturer. A new transaction should be established with identity hash



**Fig. 8.1** Procedure in (a) digital identity service, (b) anonymous access identification, and (c) trusted service provisioning modes

of digital signature and public key. Then, the transaction should be recorded into blockchain with a block, which can be inquired by other legal devices.

### ***8.1.2 Anonymous Access Identification***

Anonymous access identification, as shown in Fig. 8.1b, ensures the trusted device accesses into C-RAN after verification without revealing its privacy information, which is a good trade-off between the privacy and credibility of equipment. Such scenario involves bidirectional authentication between device and network operator where both of them have digital identity and identification.

First, the device can access network with the network operator identification. When a device service arrives at network, the device blinds own secret keying material for privacy protection and sends a request to network operator with the hash of digital identity for verification. After receiving the request, SDN controllers perform path calculation between the device and PU nodes considering optical, wireless, and PU resources utilization. Then, spectrum and wireless integrated path is set up based on the selected path with the optical fiber and wireless links. Device sends a verified request to the PU with hash, and the network operator will check it in blockchain whether the record is matched. If device is certified legally, the network operator responds the device with its own digital certificate and zero-knowledge proof (ZKP) code [10]. Then, the device attempts to decrypt digital certificate by using the published key. The device can make sure the network operator is legal if the decrypting is successful, which can avoid to access the pseudo-base station.

After bidirectional authentication, the ZKP with anonymous identification should be performed to enable the network operator for verifying whether the device is legal without getting its digital identity. This decreases the identification cost in terminal access. A congruential method is utilized to implement the ZKP where device sends a constant to network operator to calculate the remainder of the constant and ZKP code. Next, network operator sends a message with the calculation result and new constant to the device, waiting for device verification. If wrong occurs in the calculation result, the device indicates the network operator is out of consideration; otherwise, the device calculates the remainder of the new constant and ZKP code and sends to network operator. Then, the device is allowed to access into the C-RAN if the calculation result is verified as correct, and vice versa. Finally, network operator registers the identity information of commissioning device into blockchain, and manufacturer returns the reply to the device for updating.

### ***8.1.3 Trusted Service Provisioning***

Trusted service provisioning can accommodate the trusted access service after tripartite authentication and address the service provisioning with multiple layer



optimization among the optical spectrum, wireless, and processing resources. Figure 8.1c illustrates its interworking procedure.

WC and OC termly detect the traffic status of each RU and optical switch to obtain the wireless frequency and spectrum utilization. When a new device service identified by network operator and manufacturer arrives at RU, RU transforms a setup demand to WC for trusted service provisioning. Then, WC conducts resource analysis and forwards the results to OC. Furthermore, multiple resource optimization can be addressed to decide which PU can be the destination for trusted service accommodation. An optimal path from RU to PU with trusted requirement can be computed to ensure the optimization among wireless frequency, spectrum, and processing resources. Based on the above-calculated PU candidates, OC demands the optical switches to establish the path via credible nodes and allocates optical spectrum on path. When OC receives the response of successful setup from the credible switches and nodes, it should reply to WC with the updated information of path and spectrum. Next, WC sends the setup request to RU in order to modulate the wireless frequency to optical spectrum, sewing up the end-to-end path from authenticated device to PU. Moreover, CC should modify the utilization of PU to achieve the synchronization according to a message from WC.

## 8.2 Trusted Multi-Domain Cooperation Mechanism

With the commercialization of 5G technology [11], it is foreseeable that the number of connected devices for the Internet of Things will greatly increase, which will generate a large amount of data that needs to be processed in real time [12]. However, due to the limitations of the energy and computing power of the Internet of Things devices, these data rely on the cloud. Service places extremely high requirements on the processing capacity and security of the existing single service provider. United cloud adopts a similar concept of a joint company [13]. By allowing the cloud platforms of different service providers to collaborate and integrate network resources between different service providers, it can efficiently provide customers with cloud services and respond to more customer access.

In response to the needs of the Internet of Things, a multilevel trust structure based on the advantages of blockchain is proposed [14–16]. As shown in Fig. 8.2, network devices have three trust levels. Trust I network devices manage network devices and regulate network operation. They have the right to add and delete network devices based on device credit value. Trust I devices will improve their trust value and collect network tokens when serving the network. The network equipment with intermediate trust (Trust II) is the equipment with low credit, that is, the quality of service it provides is relatively poor. However, Trust II devices can upgrade themselves to Trust I by continuously providing high-quality services. Trust III devices are network access devices, which provide computing, storage, and other services to other devices. When Trust II or Trust III devices become untrusted due to poor service quality, their accounts will be frozen, which means that they

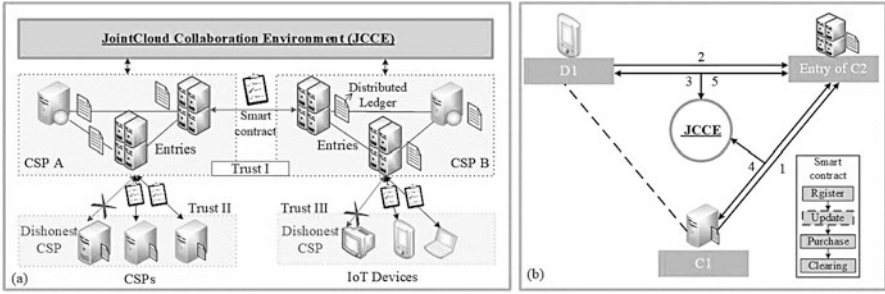


Fig. 8.2 (a) Hierarchical trust network architecture, (b) transaction processing process

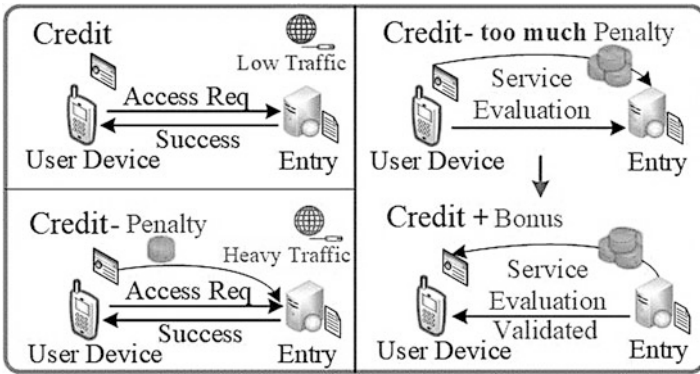


Fig. 8.3 Cooperation mechanism with token

cannot continue to engage in network services and profit from them. In this case, the blockchain consensus device acts as a joint cloud collaboration environment to provide blockchain network services for the device, as shown in Fig. 8.2b. The blockchain system acts as the middleware of network services. Some nodes of the blockchain system are selected to implement the subscription service of blockchain transactions. First, the transaction is subscribed by the node, and then sent by the node to other network devices that subscribe to the transaction.

In the multilevel trust structure, a credit token-based access control and multiparty cooperation mechanism is designed. As shown in Fig. 8.3, the different credit ratings in the above structure are divided according to credit tokens and service complaint rates. The service provider with more credit tokens and the lowest service complaint rate in the entire network will be given the authority to maintain the network. According to the traffic pressure it receives, it is allowed to collect credit tokens that are positively related to the traffic pressure. Attempting to access a network node under a higher load needs to pay a certain amount of credit coins to its owner. Due to frequent requests that need to pay a high amount of credit tokens, this makes it impossible for malicious users to continuously initiate access

to network nodes under high load. At the same time, credit tokens are a pass to the network, and malicious users who have lost their credit tokens cannot operate on the network, thereby preventing continuous attacks by malicious users and ensuring network security.

### 8.3 Distributed Trusted Routing Calculation Method

With the rapid development of Internet of Things applications and wireless network technology, a large number of network terminal connection needs have been generated [17]. At the same time, various new network applications have also made business composition more and more complicated, which puts forward high requirements for the flexibility of network control. The software-defined data center network (SDCN) can provide large-capacity content storage functions and large-bandwidth data transmission functions, and can customize the allocation of spectrum resources according to network functions and business needs, meeting various new applications on the network control the requirements for flexibility [18, 19].

In response to the above requirements, we proposed a blockchain-based SDCN distributed trusted control (BlockTC) structure [20], as shown in Fig. 8.4, which solves the trust and privacy issues among multiple controllers in the distributed SDCN to adapt to the 5G era. In order to realize the leakage of private data in multi-domain MEC, we propose network-driven collaborative routing verification (ND-CRV) for the scenario of missing routing data in the cloud, as shown in Fig. 8.5; for the scenario of complete routing data in the cloud, we propose cloud-driven collaborative routing verification (CD-CRV), as shown in Fig. 8.6. It should be noted that in the blockchain, the controller constitutes the consensus group of the blockchain network through strict member access and management mechanisms.

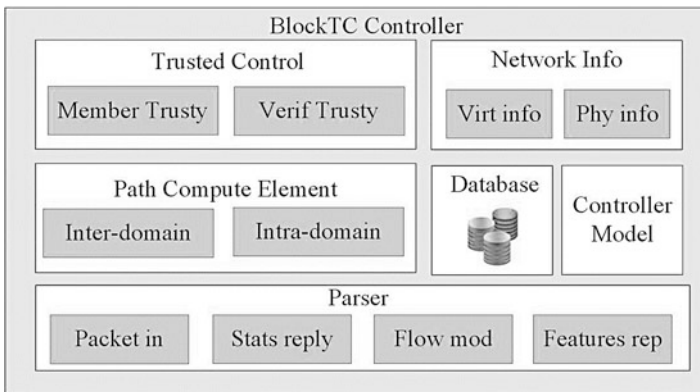


Fig. 8.4 The functional model of BlockTC

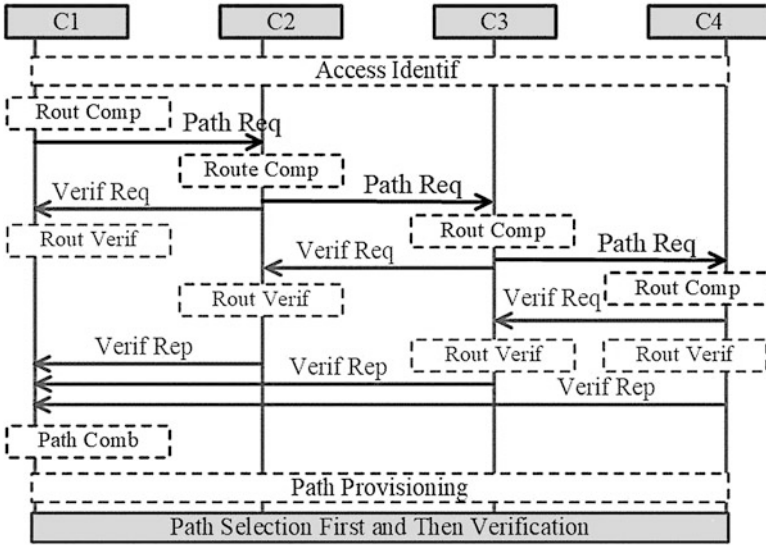


Fig. 8.5 Network-driven collaborative routing verification (ND-CRV)

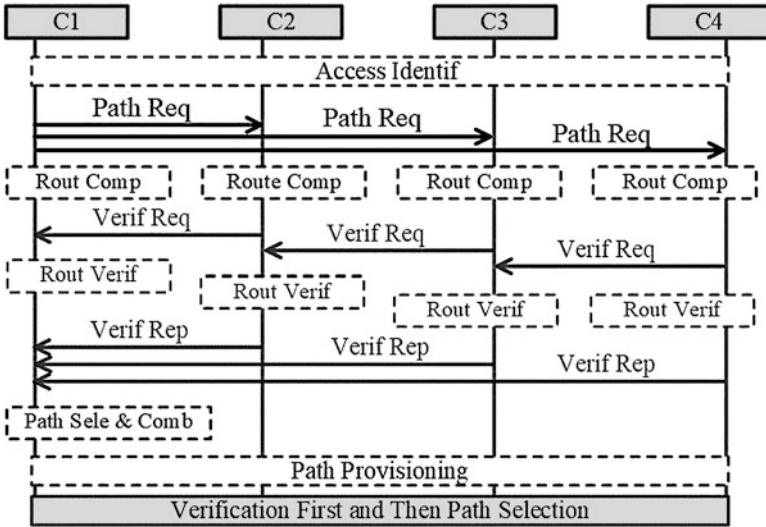


Fig. 8.6 Cloud-driven collaborative routing verification (CD-CRV)

Figure 8.5 describes the interaction process of ND-CRV in BlockTC. When the controller assumes all the tasks of the MEC server, the ND-CRV is responsible for the calculation of the PCE-based intra-domain path. Through blockchain-based authentication services, each controller will obtain a blockchain-trusted identifier and perform distributed consistency in the chain network. At the beginning of the interaction, we set the cross-domain interaction request to pass through controller 1 (C1) in domain 1. C1 calculates the requested intra-domain path according to the intra-domain topology, selects the path with the shortest distance as the best path, and sends a new request with virtual path topology to the subsequent controller. After calculating and selecting the path in the domain, the subsequent controller C2 sends the verified request and its virtual information to the next controller to ensure that the route is credible. Virtual information can be used to identify paths within the domain. The iterative process ends when the sink node is reached, and the path calculation result is returned to C1. C1 receives verified paths from other domain controllers, and arranges the trusted paths into trusted routes in order. Multi-server edge computing tasks are suitable for routing configuration between controllers. In ND-CRV, the controller implements trusted routing in the way of first calculating and then verifying.

The CD-CRV scheme is applied to the situation where the controller is not sensitive to the characteristics of the MEC server, as shown in Fig. 8.6. In this case, the administrator can predetermine the sparse routing between domains and send corresponding messages, and the MEC task simultaneously sends requests to the selected domains through the cloud. Hence, the intra-domain controller only needs to calculate intra-domain routes and verify neighboring routes, thereby speeding up the routing process. After the controller receives the request, it will calculate the selected path from its domain to the subsequent domain and calculate the total path length of the candidate route. Then, they send a route verification request with multiple candidate paths to subsequent controllers to complete route verification. CD-CRV uses the same routing verification algorithm as the ND-CRV solution, but the difference is that CD-CRV verifies multiple paths in the domain at the same time, and finally calculates multiple trusted routes. By comparing the path length combination of each route, the controller can select a route with the least transmission cost from the trusted routes as the final route. In CD-MCR, the cloud determines the inter-domain path, and the controller determines the trusted route by first verifying and then calculating.

## 8.4 Fast Fault Recovery Mechanism

With the development of big data and cloud computing, traditional network architectures are becoming increasingly rigid, and the tightly coupled design of the data plane and control plane has led to a long period of time for adding a new function to network equipment [21]. In order to improve the current TCP/IP network architecture, software-defined networking (SDN) applications were born [22]. Thanks to

the advantages of large capacity, long distance, low cost, and large access volume, optical and wireless networks have replaced traditional transmission networks and have been widely used [23]. Applied in the live network, software-defined optical and wireless networks integrate optical transmission networks and wireless access networks into the SDN architecture. Operators can dynamically orchestrate network architecture and functions according to their own business requirements, and the control layer provides unified scheduling and control capabilities. The control plane is very important to the entire optical and wireless network, so the security of the control plane has also received extensive attention. Hackers can use vulnerabilities such as easily modified flow tables and firewall policy conflicts to pass through existing firewalls. Once a control plane attack causes a single point of failure, malicious traffic can enter the network for data theft and malicious modification, posing a great threat to network security.

Based on this, combined with the advantages of blockchain, optical network, and wireless network technology [24, 25], we propose a software-defined network controller failure recovery strategy based on blockchain in the context of optical and wireless networks. The first is blockchain-based high-efficiency security strategy (BHSS) [26]. As shown in Fig. 8.7, while ensuring the fault-tolerant operation of distributed SDON control [27, 28], it can also provide data/state consistency about the controller and the underlying switching equipment.

In BHSS, the whole network structure is divided into four layers: intelligent contract layer, consensus layer, control layer, and data layer. The smart contract layer is used to set up specific computer protocols to achieve information dissemination, verification, or execution of contracts. The data layer is an edge processing node deployed at the edge of the network to realize edge computing tasks and data

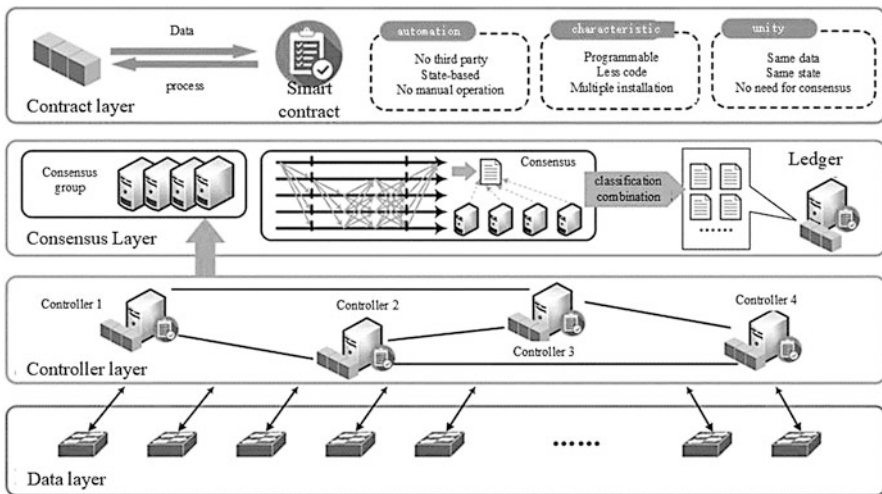


Fig. 8.7 Blockchain-based high-efficiency security strategy

storage. The control layer is the optical node deployed in the backbone network, which uses the traditional SDON method to achieve routing calculation and other functions. The consensus group is composed of distributed multicontrollers in the control layer, which verifies the blockchain network transactions, and the smart contract can be executed after the verification is qualified. Smart contracts allow trusted transactions without a third party, which are traceable and irreversible [29]. The routing request will be sent to all the controllers in the verification group, and each controller performs routing calculations independently and then performs routing verification through multiparty consensus. The routing result of successful consensus will be sent to the underlying network by each controller, and the flow table will be updated when the optical switching node receives the routing result of the threshold set by the adopted consensus algorithm. Through the multiparty consensus of the verification group, BHSS can realize trusted routing calculations, and a small number of faults or malicious controllers will not be able to affect the normal operation of BHSS. In addition, all controllers in the verification group store the consistent routing information, and optical switching node status of the entire network after the consensus is successful, so as to achieve a high degree of consistency in the data/status of the entire network.

Based on BHSS, we further propose a blockchain ledger-based recovery algorithm (BLRA), which realizes effective recovery of controller failures through a preset smart contract. In the algorithm design, the blockchain stipulates that every  $n$  traffic processing results of the controller form a blockchain data block, and an evaluation factor  $\alpha$  is established to calculate the optimal network controller of the switch. Standardization factors include calculations, network resources, and other parameters. In the resource parameters, this article defines  $U_c$  to describe the degree of use of the controller's computing unit, and uses  $\sum_{l=1}^{H_c} W_{c,l}$  to represent the state of network bandwidth resources, where  $H_c$  and  $W_{c,l}$  indicate the number of hops and traffic occupancy ratio of the path from the service, respectively. For service-related indicators, this article uses  $\sum_{t=1}^p e_{c,t}$ , where  $t$  describes the latest correlation of controller  $c$  in  $p$  services related to the switch. In summary, the evaluation factor  $\alpha$  with  $k$  candidate-slave controllers is described as formula (8.1), where  $\beta$  and  $\gamma$  can be adjusted according to specific conditions:

$$\alpha = \frac{U_c}{\{U_i\}}\beta + \frac{\sum_{l=1}^{H_c} W_{c,l}}{\left\{\sum_{l=1}^{H_c} W_{i,l}\right\}}\gamma + \frac{\left\{\sum_{t=1}^p e_{i,t}\right\}}{\sum_{t=1}^p e_{c,t}}(1 - \beta - \gamma) \quad (8.1)$$

Due to the variability of services, when the controller fails and needs to be restored, the traditional fixed master-slave relationship distributed control structure cannot provide flexible remapping functions for the optical switching node, so that the controller-optical switching node after failure recovery. The mapping relationship cannot provide optimal network services. Utilizing the strong consistent data provided by BHSS, BLRA can seamlessly remap the optical switching node to



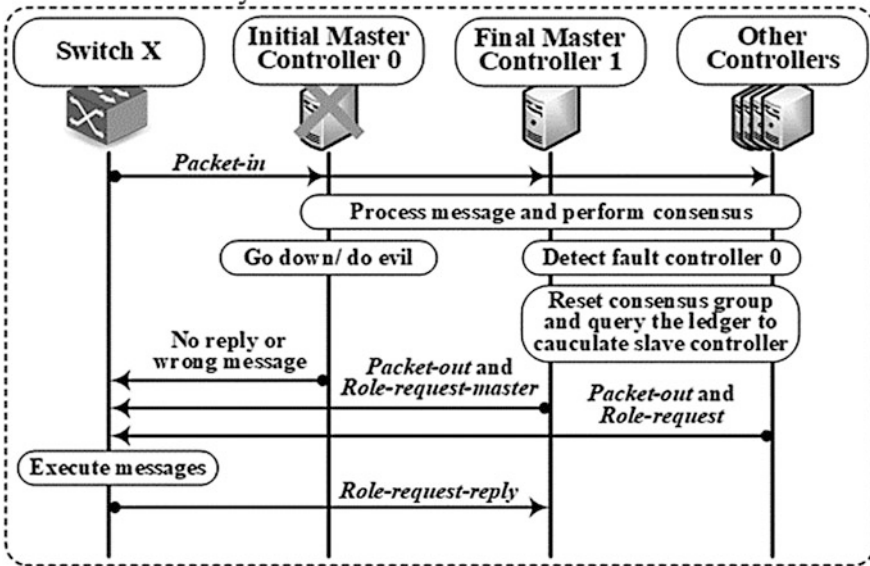


Fig. 8.8 Fast failure recovery algorithm based on blockchain ledger

the best slave controller when the master controller fails, as shown in Fig. 8.8. The error detection function in the consensus mechanism can detect the faulty controller immediately, and other normal controllers immediately execute the preinstalled BLRA smart contract to recover from the fault and send the remapping result to the optical switching node along with the flow table without consuming additional resources. When dynamically selecting the slave controller for failure recovery, BLRA applies a global load balancing mechanism to comprehensively consider the application resources, bandwidth resources, and business relevance in the current network to select the optimal slave controller to make the remapped network structure best network services.

### 8.5 Conclusion

In this paper, from the perspective of security, the possible problems of 6G network are studied. Aiming at the privacy exposure of network devices, this paper proposes a digital identity and anonymous access authentication mechanism based on blockchain, which effectively improves the security of network devices; aiming at the data isolation problem of distributed network, this paper proposes a trusted multi-domain cooperation mechanism based on blockchain, which successfully solves the data island problem of Internet of Things; aiming at the intrusion of network controller scenario, this paper proposes a trusted multi-domain cooperation



mechanism based on blockchain, a distributed trusted routing calculation method based on cross-domain control cooperation of blockchain, which ensures the stability of the Internet of Things system and can provide trusted services when attacked; for network controller failure, this paper proposes a fast fault recovery mechanism based on blockchain, which further ensures the stability of the Internet of Things system.

## References

1. Yao, Q., Yang, H., Yu, A., & Zhang, J. (2019). Transductive transfer learning-based spectrum optimization for resource reservation in seven-core elastic optical networks. *IEEE/OSA Journal of Lightwave Technology*, 37(16), 4164–4172.
2. Yang, H., Wu, Y., Zhang, J., Zheng, H., Ji, Y., & Lee, Y. (2018). BlockONet: blockchain-based trusted cloud radio over optical fiber network for 5G Fronthaul. OFC2018, W2A.25, San Diego.
3. Yuan, J., Yang, H., Liang, Y., Yao, Q., Jiao, L., & Zhang, J. (2020). *Blockchain-based Bonus-penalty Access Control Strategy for IoT Service in Cloud Radio Over Fiber Network*. International Wireless Communications and Mobile Computing (IWCMC).
4. Yu, A., Yang, H., Nguyen, K. K., Zhang, J., & Cheriet, M. (Mar. 2021). Burst traffic scheduling for hybrid E/O switching DCN: An error feedback spiking neural network approach. *IEEE Transactions on Network and Service Management*, 18(1), 882–893.
5. Yoon, C., & Cho, D. (Oct. 2015). Energy efficient beamforming and power allocation in dynamic TDD based C-RAN system. *IEEE Communications Letters*, 19(10), 1806–1809.
6. Yang, H., Wang, B., Yao, Q., Yu, A., & Zhang, J. (Dec. 2019). Efficient hybrid multi-faults location based on Hopfield neural network in 5G coexisting radio and optical wireless networks. *IEEE Transactions on Cognitive Communications and Networking*, 5(4), 1218–1228.
7. Yang, H., Zhang, J., Ji, Y., & Lee, Y. (Aug. 2016). C-RoFN: Multi-stratum resources optimization for cloud-based radio over optical fiber networks. *IEEE Communications Magazine*, 54(8), 118–125.
8. Dong, S., Yang, H., Yuan, J., Jiao, L., Yu, A., & Zhang, J. (2020). *Blockchain-based cross-domain authentication strategy for trusted access to Mobile devices in the IoT* (pp. 1610–1612). 2020 International Wireless Communications and Mobile Computing (IWCMC).
9. Yuan, J., Yang, H., Dong, S., Yao, Q., Jiao, L., & Zhang, J. (2020). *Demonstration of Blockchain-based IoT devices anonymous access network using zero-knowledge proof*. International Wireless Communications and Mobile Computing (IWCMC).
10. Li, W., Guo, H., Nejad, M., & Shen, C.-C. (2020). Privacy-preserving traffic management: A Blockchain and zero-knowledge proof inspired approach. *IEEE Access*, 8, 181733–181743.
11. Yang, H., Yao, Q., Yu, A., Lee, Y., & Zhang, J. (May 2019). Resource assignment based on dynamic fuzzy clustering in elastic optical networks with multi-core fibers. *IEEE Transactions on Communications*, 67(5), 3457–3469.
12. Yang, H., Zhao, X., Yao, Q., Yu, A., Zhang, J., & Ji, Y. (2020). Accurate fault location using deep neural evolution network in cloud data center interconnection. *IEEE Transactions on Cloud Computing*, 99, 1–11. <https://doi.org/10.1109/TCC.2020.2974466>
13. Yang, H., Guan, L., Nan, J., Zhao, X., Liang, Y., Yao, Q., Yu, A., & Zhang, J. (2019). *Intelligent optical network with AI and Blockchain*. ICOCN2019, Invited.
14. Yang, H., Yuan, J., Yao, H., Yao, Q., Yu, A., & Zhang, J. (March 2020). Blockchain-based hierarchical trust networking for JointCloud. in *IEEE Internet of Things Journal*, 7(3), 1667–1677.

15. Yang, H., Li, Y., Guo, S., Ding, J., Lee, Y., & Zhang, J. (2019). Distributed Blockchain-based trusted control with multi-controller collaboration for software defined data center optical networks in 5G and beyond." OFC2019, Th1G.2, Mar. 2019, San Diego.
16. Yang, H., Zheng, H., Zhang, J., Wu, Y., Lee, Y., & Ji, Y. Blockchain-based Trusted Authentication in Cloud Radio over Fiber Network for 5G. IEEE ICOCN2017, Young Scientist Award, YS-11, Aug. 2017, Wuzhen, China.
17. Yang, H., Yu, A., Zhang, J., Nan, J., Bao, B., Yao, Q., & Cheriet, M. (Mar. 2021). Data-driven network Slicing from Core to RAN for 5G broadcasting services. *IEEE Transactions on Broadcasting*, 67(1), 23–32.
18. Yang, H., Zhan, K., Kadoch, M., Liang, Y., & Cheriet, M. (2020). BLCS: Brain-like distributed control security in cyber physical systems. *IEEE Network*, 34(3), 8–15.
19. Yang, H., Liang, Y., Yao, Q., Guo, S., Yu, A., & Zhang, J. (Jun. 2019). Blockchain-based secure distributed control for software defined optical networking. *China Communications*, 16(6), 42–54.
20. Yang, H., Liang, Y., Yuan, J., Yao, Q., Yu, A., & Zhang, J. (2020). Distributed Blockchain-based trusted multidomain collaboration for Mobile edge computing in 5G and beyond. *IEEE Transactions on Industrial Informatics*, 16(11), 7094–7104.
21. Wu, Y. (2020). Cloud-edge orchestration for the internet-of-things: Architecture and AI-powered data processing. In *IEEE Internet of Things Journal*.
22. Yang, H., Zhan, K., Yao, Q., Zhao, X., Zhang, J., & Lee, Y. (2020). Intent Defined Optical Network with Artificial Intelligence-based Automated Operation and Maintenance. *Science China Information Sciences*, 63(6), 160304.
23. Wu, Y., Wang, Z., Ma, Y., & Leung, V. C. M. (2021). Deep reinforcement learning for blockchain in industrial IoT: A survey. *Computer Networks*.
24. Wu, Y., Dai, H.-N., & Wang, H. (2021). Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. *IEEE Internet of Things Journal*, 8(4), 2300–2317.
25. Yang, H., Zhang, J., Zhao, Y., Ji, Y., Han, J., Lin, Y., & Lee, Y. (Aug. 2015). CSO: Cross stratum optimization for optical as a service. *IEEE Communications Magazine*, 53(8), 130–139.
26. Liang, Y., Yang, H., Yao, Q., Guo, S., Yu, A., & Zhang, J. (2019). *Blockchain-based efficient recovery for secure distributed control in software defined optical networks* (Vol. 2019, pp. 1–3). Optical fiber communications conference and exhibition (OFC).
27. Yang, H., Zhang, J., Zhao, Y., Han, J., Lin, Y., & Lee, Y. (2016). SUDO: Software defined networking for ubiquitous data center optical interconnection. *IEEE Communications Magazine*, 54(2), 86–95.
28. Yang, H., Zhang, J., Ji, Y., He, Y., & Lee, Y. (2016). Experimental demonstration of multi-dimensional resources integration for service provisioning in cloud radio over fiber network. *Scientific Reports*, 6, 30678.
29. Kou, S., Yang, H., Zheng, H., Bai, W., Zhang, J., & Wu, Y. (2017). *Blockchain mechanism based on enhancing consensus for trusted optical networks*. ACP.

# Chapter 9

## Fog Computing Security and Privacy for Internet of Things (IoT) and Industrial Internet of Things (IIoT) Applications: State of the Art



Yasmine Harbi, Zibouda Aliouat, and Saad Harous

### 9.1 Introduction

Cloud computing, which was introduced in 2008, permits the use of centralized and scalable computing and storage resources to enhance the quality of services and decrease the costs of data management. There are different types of cloud providers, namely, public, private, and hybrid. Public cloud providers supply, on-demand, computing and storage services over the Internet. Private cloud providers produce computing services to internal users and offer flexibility and convenience. Hybrid cloud providers consist of a combination of public and private clouds; the public cloud allows the creation of a scalable solution, while the private cloud preserves critical data access control [18, 20].

The Internet of Things (IoT) enables physical objects to collect data and communicate with each other through the Internet. These objects generate an enormous amount of data which has to be stored, processed, and presented in a seamless and efficient form [14]. The interconnection of IoT objects provides several sophisticated IoT applications such as smart healthcare, smart transportation, and smart manufacturing, known as Industrial IoT (IIoT). IIoT uses machine-to-machine technology to automate the process of manufacturing with limited human intervention. It aims to better control the production process, data, and issues to provide efficient and reliable final products [28].

The IoT devices are generally resource-constrained; they have limited storage and processing capacity. The combination of cloud computing and IoT provides

---

Y. Harbi · Z. Aliouat  
LRSD Laboratory, Ferhat Abbas University of Setif-1, Setif, Algeria  
e-mail: [yasmine.harbi@univ-setif.dz](mailto:yasmine.harbi@univ-setif.dz); [zaliouat@univ-setif.dz](mailto:zaliouat@univ-setif.dz)

S. Harous (✉)  
College of Computing and Informatics, University of Sharjah, Sharjah, UAE  
e-mail: [harous@sharjah.ac.ae](mailto:harous@sharjah.ac.ae)

centralized data storage, powerful data processing, and fast applications spreading with few costs [23]. Unfortunately, the centralization of cloud resources increases network latency. In addition, cloud computing cannot support the requirements of mobility and location awareness of IoT applications.

To deal with these concerns, a new paradigm called fog computing was introduced by Cisco in 2012 [6]. It serves as a middle layer between cloud and IoT devices that offers computation, storage, and networking services. The fog nodes are deployed in a distributed architecture at the edge of the network and have the following features:

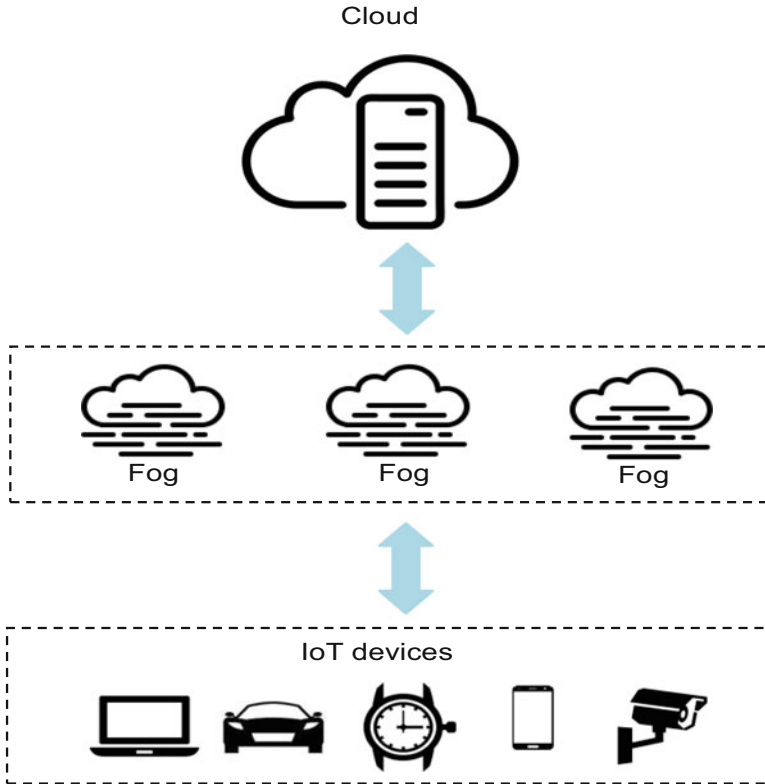
- **Low latency:** the fog nodes are close to IoT devices which reduces the latency of communications.
- **Geographic distribution:** the fog nodes are deployed in different locations which allows receiving a high-quality data stream from IoT devices.
- **Location awareness:** the fog nodes can manage and control location information to support IoT devices with location-based services at the network edge.
- **Mobility:** a fog node can be any static device such as a traffic camera, or any mobile device such as a smart vehicle or smartphone.
- **Large scale:** the fog nodes provide distributed services to support scalability when the number of IoT devices increases.
- **Heterogeneity:** the fog nodes can be used for different IoT applications.

To address the growing number of connected objects and emerging applications in IoT, fog nodes collaboratively and intelligently manage computing, storage, and networking/communication resources at the network edge near to the IoT devices as depicted in Fig. 9.1. Therefore, the data transfer time and the amount of network transmission are greatly reduced [9].

Fog computing can effectively meet the requirements of real-time/latency-sensitive, mobility-based location-aware applications. The fog nodes connect with IoT devices and users through wireless connection modes such as Bluetooth, Wi-Fi, and 5G, and with the cloud by the Internet in order to efficiently utilize computing and storage services of the cloud [1].

In spite of the advantages of fog computing adoption in IoT networks, fog nodes meet different types of security and privacy threats [13]. The existing security solutions in cloud computing could be adopted to overcome some security vulnerabilities in fog computing, but it still faces several challenges due to its typical features, such as decentralization, mobility, location awareness, and heterogeneity.

In this chapter, we shed light on the concepts of IoT, IIoT, and Industry 4.0 and highlight the difference between these terms. We present the role of fog computing in various IoT applications. Then, we introduce the security issues of fog-enabled IoT systems. Lastly, we review promising techniques that solve the security threats of fog and analyze open challenges to improve the security of fog-assisted IoT applications. The remainder of this chapter is organized as follows. Section 9.2 introduces the concepts of IoT, IIoT, and Industry 4.0. A taxonomy of common fog-enabled IoT applications is provided in Sect. 9.3. Section 9.4 presents the security



**Fig. 9.1** Fog-enabled IoT architecture

issues of fog computing in IoT. Security solutions and challenges are discussed in Sect. 9.5. We conclude the chapter in Sect. 9.6.

## 9.2 Internet of Things (IoT), Industrial Internet of Things (IIoT), and Industry 4.0

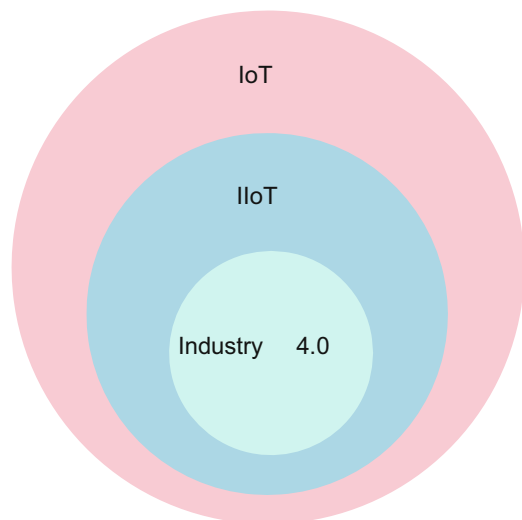
The term “Internet of Things” was first coined by Kevin Ashton in 1999 as a concept for connecting things or objects to the Internet. Over the past few years, the IoT has gained significant mindshare in both academia and industry fields due to the potential capabilities that it promises to offer. The ultimate goal is to enable objects around us to efficiently sense our surroundings, seamlessly communicate with each other, and act intelligently to provide a smart environment. The characteristics of IoT bring several interesting benefits for different domains, specifically industrial applications.

The IoT is expected to bring significant impact to the industry leading to the concept of Industrial IoT (IIoT). IIoT comprises machine-to-machine (M2M) and communication technologies with data automation and exchange. It paves the way to ameliorate production efficiency and quality and decrease production costs by connecting machines with information systems. As a consequence, a large amount of data is collected and smartly processed to provide optimal industrial operations. Communication technologies in IIoT are classified into wired and wireless. Two types of wireless communication technologies are being employed in IIoT; small-area wireless communication technologies such as Zigbee and Bluetooth, and large-area wireless communication technologies including 4G and 5G [8].

The current industrial manufacturing is undergoing new technology called Industry 4.0 by integrating the IoT paradigm with the cyber-physical systems (CPSs) [33]. The Industry 4.0 concept, known as the fourth industrial revolution, has drawn a lot of attention in recent years. It is widely adopted due to the use of Internet technologies and smart computing in industrial production and manufacturing to intend data automation, reliability, and control. It aims to make complete autonomous systems by reducing human intervention.

The core concept behind IIoT and Industry 4.0 is the use of advanced technologies such as IoT, 5G, cloud computing, fog computing, etc., to offer smart and cost-effective industrial processes and operations. As a subset of IoT, the focus of IIoT is ensuring efficient management, optimized monitoring, and controlling with cost reduction of industrial applications. Industry 4.0 is a subset of IIoT which focuses on safety and efficiency in manufacturing. Figure 9.2 shows the relation between IoT, IIoT, and Industry 4.0 that are closely related concepts but cannot be interchanged.

**Fig. 9.2** IoT, IIoT, and Industry 4.0



IIoT devices are typically distributed in noisy and harsh environments and have real-time and reliability requirements to collect environmental data and deliver control decisions. The quality of service (QoS) offered by IIoT is often measured by satisfying the end-to-end deadlines of the real-time sensing and control tasks executed in the system. Since the workflow of the smart manufacturing system will be processed automatically without human intervention, real-time response, low network latency, and reliability are critically needed. Thus, the data transmission/response and decision-making should be optimized to be immediate and accurate in the context of Industry 4.0.

To realize the opportunities offered by IoT and IIoT, there are many challenges such as resource-constrained devices, low-latency requirements, real-time performance, data management, and security and privacy that cannot be adequately addressed by cloud infrastructure. This calls for an efficient network architecture that can enable computing, control, storage, and networking functions close to IoT devices.

### **9.3 Fog Computing Applications in IoT and IIoT**

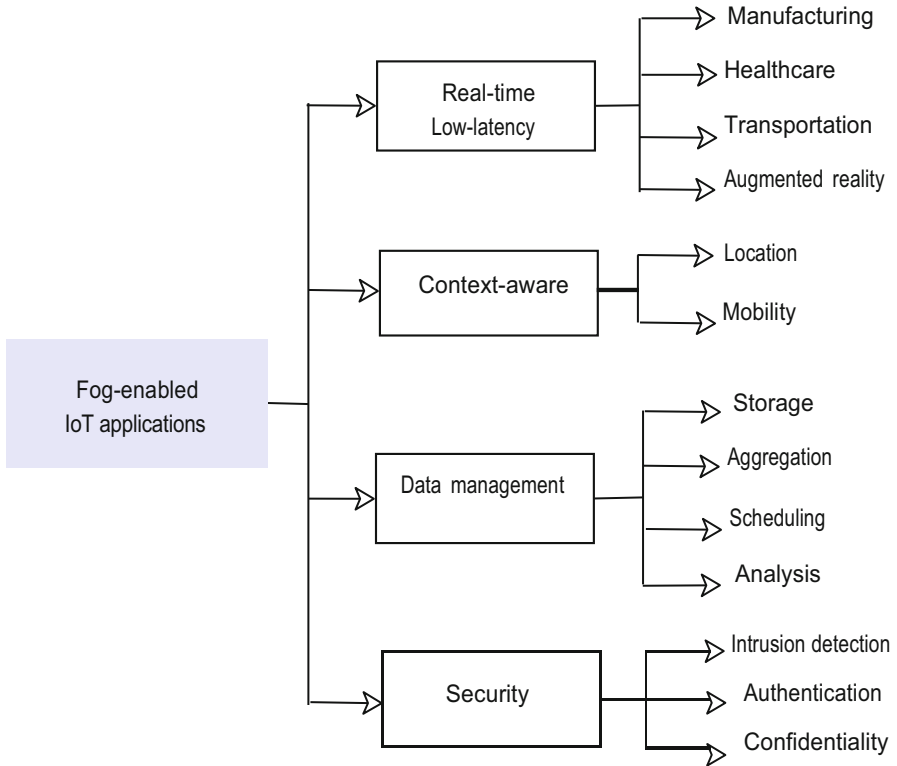
This section describes existing works in the literature that discuss the integration of fog computing with the IoT in various applications. A classification of major fog-enabled applications is presented in Fig. 9.3.

#### **9.3.1 Real Time and Low Latency**

The fog nodes distributed at the network edge obtain the collected data from the IoT devices and provide local computation and storage services. Because the fog nodes are close to IoT devices, they can handle real-time services and offer low latency for different IoT environments including manufacturing, healthcare, transportation, and virtual reality.

The industrial process requires most of the tasks to be performed locally where a middleware is required between the industrial environment and the cloud. The full potential of manufacturing systems can be achieved with the presence of fog nodes to perform local data processing and management tasks. In a manufacturing system, the use of sensors, actuators, and robots will help to improve the productivity and avoid unnecessary costs. Fog nodes can provide local data processing with low latency to actuators and robots [2].

In the healthcare domain, data are generated by a large number of sensors that require low-latency and real-time processing. The fog nodes receive the collected data from sensors implanted on the patient body to detect urgent health conditions and take actions instantaneously. Therefore, fog computing has a crucial function in smart e-healthcare systems. Vilela et al. [32] highlighted the benefits of fog



**Fig. 9.3** Classification of fog-enabled IoT applications

computing in healthcare IoT systems by providing a comparison to a cloud-based architecture. The proposed architecture allows patients' data to be locally analyzed by fog-based gateways. This enables real-time processing and enhances the latency compared to the cloud-based architecture.

An intelligent transportation system (ITS) is a subset of IoT that deals with transportation which constitutes the backbone of any country. A roadside unit (RSU) can be equipped with a fog node to provide smart parking, smart traffic load, and smart traffic lights. For instance, fog nodes use the collected data from video cameras and sensors distributed on roads to identify the pedestrian appearance and change the traffic lights [22]. In addition, fog nodes can be used to monitor congestion and navigation for drivers. Specifically, the fog nodes can receive a navigation request from a vehicle and cooperate with each other to rapidly generate the requested driving path [25].

Augmented reality applications are highly latency-sensitive; the latency of the response must be very small to offer a good experience for users. Therefore, fog computing has the potential to become a major player in the virtual reality domain. Zao et al. [35] designed an augmented brain-computer interaction game based on



fog nodes and cloud servers. Their system performs continuous real-time brain state classification where fog nodes analyze the data collected by sensors to detect the brain state of the player.

### 9.3.2 *Context-Aware*

The combination of cloud computing and IoT allows the centralization of data processing and storage and provides various applications and services to users. However, the cloud cannot directly access local contextual data such as users' mobility and location information. Fog nodes can manage the data and communication of IoT objects and locally provide the required services. Furthermore, they support location-aware and mobility-aware applications.

In [11], the authors focused on the mobility of sensor nodes in the context of IoT. The proposed system is based on fog computing and uses a handover mechanism for de-registering a sensor node from a source access point and registering it to a new access point.

In order to support location-aware applications, Yang et al. [34] presented fog-assisted IoT architecture where a mobile device can query and search points of interest by providing its location to a local fog node. The authors demonstrated the advantage of fog computing in terms of latency of location-based service searching compared to cloud-based schemes.

### 9.3.3 *Data Management*

Fog nodes have considerable computing and storage resources to locally manage the data collected by IoT devices. This can largely decrease the communication overhead between the fog nodes and the cloud server and efficiently achieve rapid data access and update. In addition, the fog nodes can perform simple processing (e.g., data aggregation, data scheduling, and data analysis) on the received data collected by numerous IoT devices.

Fu et al. [10] investigated the data storage in IIoT where a large amount of data is continuously generated by different sensors. Specifically, the time-sensitive data is stored locally by fog nodes and non-time-sensitive data is transmitted to the cloud server. As a result, the proposed scheme can greatly enhance the effectiveness of data storage and retrieval in IIoT systems.

Since fog nodes act as intermediate nodes in the network, they transmit the data gathered by IoT nodes to the cloud and share the data received from the cloud with the IoT nodes. Therefore, conventional communication functions such as data aggregation are quite important to improve the communication overhead. Lu et al. [21] addressed the data aggregation for heterogeneous IoT devices using

fog computing. The aggregated data is filtered by fog nodes to avoid inaccurate decisions made at the control center.

Chekired et al. [7] designed a multitier fog architecture for the IIoT system, where the collected data need to be scheduled in real-time constraints. They employed two priority queuing models to rapidly schedule emergency data and requests of different things installed inside the industrial factory.

With the huge volumes of data generated from various kinds of IoT devices, data analysis becomes a major challenge for cloud-based architecture. Tang et al. [29] introduced hierarchical fog-based architecture to support big data analysis in smart cities. The proposed hierarchical fog computing architecture offers high-performance computing, provides rapid response, and enhances the communication bandwidth.

### 9.3.4 Security

IoT devices are prone to various attacks that can affect the availability, confidentiality, and integrity of transmitted or stored data. In IIoT, the data can be misused by adversaries resulting in manufacturing malfunctions. Security protection mechanisms can be implemented on fog nodes to exclude the need for software and hardware installation, management, and updating on IoT objects. Zhou et al. [37] applied fog computing to mitigate distributed denial-of-service (DDoS) attack in IIoT. Their system is based on three-level mitigation architecture (i.e., field firewall devices, local fog nodes, cloud server) in order to provide rapid and accurate attack detection.

Since IoT devices have limited resources, fog nodes can be adopted to satisfy major security requirements such as authentication, confidentiality, and key management to secure IoT environments. Alharbi et al. [4] proposed a fog computing-based security system to secure IoT communications. They used a challenge-response authentication mechanism to verify the sources of suspicious traffic. Hu et al. [19] presented a face identification and resolution framework based on fog computing for IoT. The framework is mainly comprised of user devices, fog nodes, and cloud servers. The authors adopted several cryptographic techniques to preserve the personal information of users. Zhang et al. [36] proposed a key management scheme based on contributory broadcast encryption where fog nodes negotiate a public key with an end-user device. This latter sends an encrypted session key to the fog nodes to achieve confidentiality of further communications.

## 9.4 Security Issues of Fog Computing in IoT and IIoT

The integration of fog computing with the IoT can bring potentially tremendous benefits to human beings such as smart healthcare, smart manufacturing, smart

cities, etc. The fog supports real-time interactions between IoT devices and provides different services with low latency requirements. In addition, fog computing allows local data processing and storage which reduces the amount of data that needs to be transmitted to the cloud and enhances the network bandwidth. Fog computing has the ability to support context-aware networks and perform security operations that need huge resources on IoT devices.

The fog-assisted IoT applications face several security issues and concerns that will have an important impact on the potential benefits of the fog. Because fog computing is an extension of the cloud, it inherits various security risks. The fog computing architecture is exposed to several kinds of cyberattacks [27]:

- *Denial-of-service attack*: an attacker can disrupt the services provided by fog nodes by flooding the network with superfluous requests to make them unavailable to IoT devices/users.
- *Man-in-the-middle attack*: the IoT devices exchange messages with fog nodes in real time, which allows malicious attackers to secretly eavesdrop or modify the transmitted data between these parties.
- *Impersonation attack*: a malicious attacker can impersonate a legitimate fog node to offer misleading or phishing services to users.
- *Sybil attack*: a malicious attacker can pretend to be legitimate users by manipulating false identities and pseudonyms to exploit real-time services offered by fog nodes.

The collected data of IoT devices are analyzed and stored on local fog nodes, which makes it easy for attackers to gain access to users' data. Moreover, fog computing provides many IoT location-based services and functions due to its feature of localization. However, the users' location information is obviously exposed in fog computing. Therefore, privacy is a serious problem in fog computing-enabled IoT applications as sensitive data are included in the transmission, processing, storage, and sharing by fog nodes [26]. Aleisa et al. [3] discussed several access control models that can be applied in fog computing to preserve the privacy of IoT data.

Table 9.1 shows the security issues of the fog-enabled IoT applications presented in Sect. 9.3.

Fog computing provides a collection of real-time and low-latency services to IoT users. External attackers can access the fog services if there are no authentication and access control mechanisms to prevent unauthorized access [16].

**Table 9.1** Security issues of fog-enabled IoT applications

Application	Security issues
Real time and low latency	Man-in-the-middle, Sybil attack, Unauthorized access
Context-aware	Data privacy
Data management	Data privacy
Security	Impersonation, Sybil attack

Fog computing reduces the complexity of data management and supports mobility awareness and location awareness in various IoT applications. This launches new security and privacy issues, mainly, the disclosure of data privacy. For example, a driver on-road wants to find a restaurant by exposing his/her location to a local fog node.

The IoT takes advantage of fog computing to satisfy different security requirements. Fog nodes may cooperate with each other to provide security enhancement services for IoT users. However, multiple trust levels and various trust relationships should be established between fog and other nodes to overcome compromised fog nodes.

It is very necessary to develop secure and effective intrusion detection and privacy-preserving mechanisms to mitigate the identified security issues of fog computing in different IoT applications.

## 9.5 Security Challenges of Fog Computing in IoT and IIoT

Fog computing is an emerging technology that can be used to improve several requirements of IoT applications. However, it is vulnerable to various security threats. Existing security measurements of cloud computing are not appropriate for fog computing due to its distinctive properties, notably decentralization, mobility, and heterogeneity [24].

In fog-enabled IoT applications, an intrusion detection mechanism can be implemented on fog nodes to prevent any malicious external and internal attacks. Host-based intrusion detection systems (HIDS) collect and analyze the data about the system for the purpose of intrusion detection. HIDS identifies if the cloud is attacked or not based on the analysis on modification of host file systems and program behavior [31]. Network-based intrusion detection system (NIDS) is another type of intrusion detection system that analyzes the network traffic to detect malicious activities and attacks [15]. To discover intrusions using NIDS, several machine learning algorithms and data analysis approaches can be used for network traffic scanning.

The intrusion detection systems are useful for detecting malicious intrusions and attacks. However, they are less efficient for fog computing because of the heterogeneous and decentralized architecture. The cooperation of fog nodes is crucial to prevent various attacks and improve the efficiency of the intrusion detection system. Sharing information among cooperative fog nodes needs to guarantee all the participating entities are fully trusted. To measure the trust level of fog nodes, different access control models can be applied [3]. Selecting an access control model in fog computing depends on the application requirements. Attribute-based encryption (ABE) is a well-known public encryption technique based on attributes to guarantee fine-grained data access control [12]. There are two types of ABE: ciphertext-Policy attribute-based encryption (CP-ABE) and key policy attribute-based encryption (KP-ABE). Alrawais et al. [5] proposed a

cost-effective key distribution scheme based on CP-ABE to provide authentic and secure communications among fog nodes in IoT. However, evaluation of fog nodes' trustworthiness and design of a trust model by collecting and managing the attributes and behavior information about fog nodes is a challenging task in decentralized and dynamic architecture.

As the number of IoT devices increases, the data generated will also increase. Fog nodes have significant computational capabilities to perform data processing and analysis. Specifically, the data are submitted to the local fog nodes that provide distributed computation services with low latency. Some transmitted data may be considered sensitive, for example, in e-healthcare systems, wearable devices collect and send personal health conditions of patients through fog nodes. Therefore, encrypting the collected data by IoT devices is needed before transmission to fog nodes [32]. The fog nodes momentarily store the received data or perform further analysis before delivering it to the cloud. The analysis of encrypted data is critically important to prevent privacy leakage. To address this issue, homomorphic encryption is a widely used technique that allows computations on ciphertexts [17]. In addition, it is difficult to search on encrypted data and retrieve the required one. To achieve encrypted data search, a secure index should be constructed when uploading data to fog nodes. Searchable encryption is one of the popular methods that perform encrypted data search without revealing any private information [30]. However, these traditional security mechanisms are quite inadequate in the decentralized and dynamic architecture of fog computing.

## 9.6 Conclusion

Cloud computing evolves as an essential component for the emergence of IoT technology. With the growth of IoT, the concept of fog computing has been introduced as a new paradigm to extend (not to replace) the computational and storage resources of cloud computing. This emerging paradigm raises several security issues and challenges, specifically, in the industrial domain.

In this chapter, we provided an overview of IoT, IIoT, and Industry 4.0. We presented a classification of major fog-enabled IoT applications, including real-time and low-latency, context-aware, data management, and security. Moreover, we discussed the security issues of fog computing in IoT, including different types of security attacks and data privacy threats. Finally, we analyzed the potential challenges that should be addressed to mitigate the identified security concerns in various fog-enabled IoT applications.

## References

1. Aazam, M., & Huh, E. N. (2016). Fog computing: The cloud-iot /ioe middleware paradigm. *IEEE Potentials*, 35(3), 40–44.
2. Aazam, M., Zeadally, S., & Harras, K. A. (2018). Deploying fog computing in industrial internet of things and industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(10), 4674–4682.
3. Aleisa, M. A., Abuhusseini, A., & Sheldon, F. T. (2020). Access control in fog computing: Challenges and research agenda. *IEEE Access*, 8, 83986–83999.
4. Alharbi, S., Rodriguez, P., Maharaja, R., Iyer, P., Subaschandrabose, N., & Ye, Z. (2017). Secure the internet of things with challenge response authentication in fog computing. In *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–2. IEEE.
5. Alrawais, A., Althothaily, A., Hu, C., Xing, X., & Cheng, X. (2017). An attribute-based encryption scheme to secure fog communications. *IEEE Access*, 5, 9131–9138.
6. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16.
7. Chekired, D. A., Khoukhi, L., & Mouftah, H. T. (2018). Industrial iot data scheduling based on hierarchical fog computing: A key for enabling smart factory. *IEEE Transactions on Industrial Informatics*, 14(10), 4590–4602.
8. Cheng, J., Chen, W., Tao, F., & Lin, C. L. (2018). Industrial iot in 5g environment towards smart manufacturing. *Journal of Industrial Information Integration*, 10, 10–19.
9. Datta, S. K., Bonnet, C., & Haerri, J. (2015). Fog computing architecture to enable consumer centric internet of things services. In *2015 International Symposium on Consumer Electronics (ISCE)*, pp. 1–2. IEEE.
10. Fu, J. S., Liu, Y., Chao, H. C., Bhargava, B. K., & Zhang, Z. J. (2018). Secure data storage and searching for industrial iot by integrating fog computing and cloud computing. *IEEE Transactions on Industrial Informatics*, 14(10), 4519–4528.
11. Gia, T. N., Rahmani, A. M., Westerlund, T., Liljeberg, P., & Tenhunen, H. (2018). Fog computing approach for mobility support in internet-of-things systems. *IEEE Access*, 6, 36064–36082.
12. Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98.
13. Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys & Tutorials*, 17(3), 1294–1312.
14. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660.
15. Hamad, H. M., & Al-Hoby, M. (2012). Managing intrusion detection as a service in cloud networks. *International Journal of Computer Applications*, 41(1), 35.
16. Harbi, Y., Aliouat, Z., Harous, S., Bentaleb, A., & Refoufi, A. (2019). A review of security in internet of things. *Wireless Personal Communications*, 108(1), 325–344.
17. Harbi, Y., Aliouat, Z., Refoufi, A., & Harous, S. (2019). Efficient end-to-end security scheme for privacy-preserving in iot. In *2019 International Conference on Networking and Advanced Systems (ICNAS)*, pp. 1–6. IEEE.
18. Hayes, B.: Cloud computing (2008).
19. Hu, P., Ning, H., Qiu, T., Song, H., Wang, Y., & Yao, X. (2017). Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things. *IEEE Internet of Things Journal*, 4(5), 1143–1155.

20. Kumrai, T., Ota, K., Dong, M., Kishigami, J., & Sung, D. K. (2016). Multiobjective optimization in cloud brokering systems for connected internet of things. *IEEE Internet of Things Journal*, 4(2), 404–413.
21. Lu, R., Heung, K., Lashkari, A. H., & Ghorbani, A. A. (2017). A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot. *IEEE Access*, 5, 3302–3312.
22. Luan, T.H., Gao, L., Li, Z., Xiang, Y., Wei, G., & Sun, L. (2015). Fog computing: Focusing on mobile users at the edge. *arXiv*. Preprint arXiv:1502.01815.
23. Luong, N. C., Hoang, D. T., Wang, P., Niyato, D., Kim, D. I., & Han, Z. (2016). Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models: A survey. *IEEE Communications Surveys & Tutorials*, 18(4), 2546–2590.
24. Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: Challenges. *IEEE Access*, 5, 19293–19304.
25. Ni, J., Lin, X., Zhang, K., & Shen, X. (2016). Privacy-preserving real-time navigation system using vehicular crowdsourcing. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5. IEEE.
26. Ni, J., Zhang, K., Lin, X., & Shen, X. (2017). Securing fog computing for internet of things applications: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 20(1), 601–628.
27. Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78, 680–698.
28. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11), 4724–4734.
29. Tang, B., Chen, Z., Hefferman, G., Pei, S., Wei, T., He, H., & Yang, Q. (2017). Incorporating intelligence in fog computing for big data analysis in smart cities. *IEEE Transactions on Industrial Informatics*, 13(5), 2140–2150.
30. Varri, U., Pasupuleti, S., & Kadambari, K. (2020). A scoping review of searchable encryption schemes in cloud computing: Taxonomy, methods, and recent developments. *The Journal of Supercomputing*, 76(4), 3013–3042.
31. Vieira, K., Schuler, A., Westphal, C., & Westphal, C. (2009). Intrusion detection for grid and cloud computing. *It Professional*, 12(4), 38–43.
32. Vilela, P. H., Rodrigues, J. J., Solic, P., Saleem, K., & Furtado, V. (2019). Performance evaluation of a fog-assisted iot solution for e-health applications. *Future Generation Computer Systems*, 97, 379–386.
33. Wollschlaeger, M., Sauter, T., & Jasperneite, J. (2017). The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1), 17–27.
34. Yang, X., Yin, F., & Tang, X. (2017). A fine-grained and privacy-preserving query scheme for fog computing-enhanced location-based service. *Sensors*, 17(7), 1611.
35. Zao, J. K., Gan, T. T., You, C. K., Méndez, S. J. R., Chung, C. E., Te Wang, Y., Mullen, T., & Jung, T. P. (2014). Augmented brain computer interaction based on fog computing and linked data. In *2014 International conference on intelligent environments*, pp. 374–377. IEEE.
36. Zhang, L. (2019). Key management scheme for secure channel establishment in fog computing. *IEEE Transactions on Cloud Computing*.
37. Zhou, L., Guo, H., & Deng, G. (2019). A fog computing based approach to ddos mitigation in iiot systems. *Computers & Security*, 85, 51–62.

# Chapter 10

## Concluding Remarks: Current Challenges and Future Directions



S. R. Liyanage

In this book we have seen a few powerful solutions for solving security issues in fog computing in a wide variety of architectures and applications. In examining the accomplishments of these solutions, we have also seen that many unanswered questions remain. We will summarize what has been achieved in the field of fog computing from 5G to 6G and what are the most interesting and important directions for future research.

The case studies of projects in problem-solving, scientific modeling, and theories discussed in this book shed light on the future possibilities in the area of fog computing. A case study on the zero trust architecture concept emphasizes that fog-enabled IIoT platforms will benefit from such an architecture for risk assessment, secure access control mechanisms, and reporting management. Many architectures that are applicable are still conceptual, and various security and privacy issues exist before real-world commercial implementation. Consideration of trust even at the fog level is very important, and further research on zero trust establishing schemes with blockchain technology must be done to measure the reliability of the fog-enabled cyber-physical environments. Further research on trust-based architectures and blockchain technologies are essential for enhanced security of virtually connected smart world of Industry 4.0. The application of blockchains for trust calculations is quite challenging due to the distributed nature of blockchains where trust is decentralized. However, the possible implications of a successful breakthrough in this direction would solve many security issues of fog-enabled IIoT infrastructure.

A blockchain-based architecture was proposed as a platform for secure data storage and data sharing in a smart city in one of the case studies. The selected smart city fog nodes are used to share data among stakeholders and establish a distributed shared database. Also, in the context of smart healthcare as a use case, a smart data-

---

S. R. Liyanage (✉)

Faculty of Computing and Technology, University of Kelaniya, Kelaniya, Sri Lanka

e-mail: [sidath@kln.ac.lk](mailto:sidath@kln.ac.lk)



sharing contract enables trusted decentralized data storage, authorized data sharing between healthcare entities, automates data management, and defends against unauthorized second-hand sharing. The described solution explains how fog-based methods can enable the deployment of latency-sensitive smart city applications. It is shown that processing and storing IoT data streams in fog nodes closer to data sources can dramatically reduce network traffic and reduce latency. The deployment of fog nodes can help achieve the objectives of the smart city stakeholders and improve the quality of services offered to citizens. It was shown that setting up a blockchain network including fog nodes for a smart city would also lead to secure data sharing. Smart healthcare use cases were also showcased, where the process of executing a transaction that aims to update a patient's vital signs information collected by a healthcare provider. This highlights the reliance on validating the smart contract. There are ample new research avenues for future work in related fields.

A case study on evolutionary algorithms for enhancing mobile ad hoc network security was also presented in one of the chapters. When addressing the issues related to fog computing and their security, machine learning and artificial intelligence provide a plethora of possibilities that cannot be ignored. The careful analysis and implementation of such methods has the potential to bring self-adaptive intelligent security solutions for all types of networks including fog computing. This rich area of research definitely holds promise for the future developments that should not be ignored. The main challenge in the marriage of such novel meta-heuristic algorithms is the "no free-lunch" problem. An algorithm that would yield the best results for a certain network architecture might fail miserably when a small change is made to the network. Therefore, wider studies must be carried out to identify the most suitable computational intelligence approach to improve network security.

A blockchain-based fog computing security mechanism was proposed, where the nodes in a fog computing environment work with equal capacities without any necessity of establishing trust among them. This allows the layers and infrastructure that constitute the stack of a fog node to be owned and managed by different entities. This novel concept of security necessitates the need for distributed trust in fog computing which can be realized using the concept of blockchain technology. The stored data is persistent allowing auditability, but also ensures anonymity and untraceability. The marriage of fog computing with blockchain technology can achieve increased security. The usage and application of blockchains in existing security schemes for fog computing have been presented in the case study. The comparative analysis on the communication and computational costs and also security features among the various state-of-art security protocols proposed in the line of blockchain-based fog computing environment are a rich area for future research.

Integrating blockchain with fog and edge computing for micropayment systems was presented as a case study. The integration of blockchain technology with fog computing can catalyze many technologies forward and provide tremendous advantage in terms of security and cost, as both technologies operate on decentralized frameworks. A plethora of future research possibilities can stem from this

single integrative approach. The case study also reveals how micropayments can have more latency and scalability by employing fog computing. Increased speed and connection density offered by advances in mobile networking technology will enable real-time processing of data as well as automated transaction processing between connected devices. These improvements can have far-reaching impact on financial management of businesses. Advances in 6G networks will exhibit more heterogeneity than 5G enabling different types of devices to communicate efficiently. These developments will enhance the micropayment networks where different types of IoT devices will be able to connect and hence process payments and transactions in a more secure way. One of the chapters highlighted the various relationships among block chain, fog computing, and edge computing technologies. Various forms of integration of these technologies and the associated applications were discussed, and future challenges were identified. Integrating this intelligent solution with big data in blockchain and fog computing will change the traditional business models and support the creation of efficient and fast micropayment systems. There will be a lot of opportunities for novel innovations as well as research and development with the fusion of these bleeding edge technologies.

The chapter on medical prescription traceability using blockchain-based decentralized application addressed the issue of drug trafficking through erasure of records of drugs from registered supply chains and selling them on various channels at higher prices. The case study proposes a solution where the route taken by a particular drug in the supply chain is backtracked using blockchain-based voting system. The blockchain technology has been adopted to maintain a decentralized, incorruptible, and open register of all persistent data in this application. While blockchain is transparent, it is also private and masks any user's identity with intricate and encoded codes to protect sensitive data. The decentralized aspect also facilitates easy and secure access to the information by patients, physicians, and healthcare professionals. The proposed systematic tracking system can help to enhance the protection and supply chain control of medicinal products, by electronically collecting dynamic information, for instance, sample numbers and expiry dates, and contributes to a stronger system of pharmacovigilance. This would entail contributions from all stakeholders in terms of time, energy, and regulatory efforts. The benefit, however, is high, both from a public health point of view and in terms of cost-efficiency. It has been shown that vendor lock problems in healthcare can also be minimized by utilizing decentralized blockchain services. Increasing the network size and checking the performance and feasibility of blockchain-based decentralized prescription traceability by applying it to the real-time applications have been identified as specific future research areas for this topic.

An optical and wireless convergence network based on blockchain has also been proposed to overcome problems in 6G networks. The proposed methodology entails a digital identity and anonymous access authentication mechanism based on blockchain to address the data isolation problem in the 6G network. A trusted multidomain cooperation mechanism based on blockchain is attempted for the controller intrusion scenario in 6G. A distributed trusted routing calculation method based on cross-domain control cooperation of blockchain has been proposed to

address controller failures in 6G networks. A fast fault recovery mechanism based on blockchain is also proposed in the case study. Future research to overcome these specific shortcomings of 6G networks such as the data isolation problem, controller intrusion, controller failures, and fault recovery mechanisms can definitely apply blockchain technologies.

The state-of-the-art fog computing-based solutions for physical layer security challenges in 5G and 6G networks were presented by Brighente et al. Securing the largely distributed fog computing networks, which have a larger attack surface is a significant challenge. This issue is complicated since edge nodes are usually resource-constrained and cannot resort to sophisticated cryptographic solutions. The deployment of a public key infrastructure is complicated because of the large number of connected devices in a typical fog computing network. The adoption of physical layer security (PLS) techniques to overcome these challenges and future possibilities was presented in this chapter. The advantages provided by using simple physical layer security techniques for resource-constrained fog computing nodes can guarantee network security, confidentiality, authorization, and location verification. How PLS represents a viable solution for FC security and how it can be exploited for secure-by-design network development are clearly explained. This integration of PLS in FC is a pioneering step in delivering secure distributed next-generation networks. The future potential of this approach is yet to be fully explored, and further research is needed to adopt PLS for emerging next-generation networks.

These case studies manifest the first steps in theoretically analyzing the cutting edge in fog computing systems and possible solutions. The overarching message from the above case studies is that fog computing provides a solution to potential problems in 5G and 6G networks in this era of cyber-physical systems that demand fast and secure high-bandwidth wireless connections. The development of emergent technologies that culminate blockchain, edge computing, machine learning, and AI with fog computing is only an indication of future developments. There is massive potential in these areas that warrants deeper investments in research and development. There are many open questions, and there is much important work to be done.

# Index

## A

Access control, 11, 12, 37, 42, 49, 65–66, 101, 135, 145, 154, 159  
Anonymity, 38, 49, 132, 160  
Authentication, 7, 8, 10, 20, 32, 36, 38, 41–42, 51, 60, 64–65, 69–70, 101, 141, 161  
Authorization, 33, 36, 38, 68, 83, 162

## B

Backward secrecy, 38, 53  
Blackhole attack, 8, 19, 23  
Blockchain, 11, 80, 93, 97–100, 114, 117–119, 128, 131–142  
Blockchain-based fog computing, 31–54, 79–89, 96, 100–119  
Blockchain-based voting system, 120, 127, 161

## C

Cellular free network, 160  
Channel, 48, 50, 59, 66, 67, 70, 132, 161  
Cloud computing, 1, 3, 11, 12, 18, 31, 61, 79, 83, 93, 96, 104, 145, 148, 151, 155  
Confidentiality, 9, 38, 52, 60, 64, 79, 86, 114, 152, 162  
Context verification, 74  
Context-aware, 151, 155  
Controller intrusion, 65, 162

## D

Data integrity, 38, 48, 49, 80, 82  
Data isolation, 141

Data management, 81, 149, 151–152, 155, 160  
Data security, 80, 83–84  
Data sharing, 84–89  
Decentralized application, 113–129  
Decentralized tracking system, 116  
Device privacy, 51, 80, 82  
Distributed Denial of Service (DDoS) Attack, 7, 152  
Distributed Ledger (DLT), 11, 80  
Drug traceability, 113–129

## E

Edge computing, 93–110  
Evolutionary computing (EC), 24

## F

5G technology, 3, 4, 8, 10, 15, 59–75, 93, 102, 108, 110, 115, 131, 134, 148, 161, 162  
Fog computing, 59–75, 81, 83–84, 93–110, 145–155, 159, 161, 162  
Fog nodes, 33, 45, 47, 48, 50, 51, 64, 65, 73, 75, 80, 81, 83, 85, 87, 89, 98, 105, 107, 108, 146, 149–151, 154, 155, 160  
Forward secrecy, 38, 52  
Full-duplex, 71–73

## H

Healthcare, 4, 34, 80, 84, 86, 87, 114, 115, 149, 159, 161  
Hyperledger Fabric, 46, 86

**I**

In-region location verification, 74  
 Industrial Internet of Things (IIoT), 1–13, 37, 145–155  
 Industry 4.0, 1, 4, 146–149, 155, 159  
 Internet of Everything (IoE), 2, 32, 115  
 Internet of Things (IoT), 1–13, 19–20, 32, 34, 38, 43, 45, 47, 61, 62, 64, 66, 72, 80, 82–83, 94, 98–100, 102, 105, 145–155, 161  
 Intrusion detection, 6, 10, 12, 23, 43, 63, 65–66, 154  
 IoT applications, 15, 22–26, 32, 80

**K**

Key management, 11, 42–43, 50, 152

**M**

Man-in-the-middle (MITM) attack, 9, 13, 36, 39, 153  
 Medical prescription, 113–129  
 Meta-heuristic algorithms, 160  
 Millimetre wave communications, 72  
 Multi-antenna technology, 72  
 Multiple-Input Multiple-Output (MIMO), 60, 71, 72

**N**

Non-orthogonal multiple access, 60, 71, 73–74  
 Non-repudiation, 45, 53

**O**

Optical access, 131  
 Optical and wireless convergence network, 131–142

**P**

Permissioned Blockchain, 80, 82  
 Pharmaceutical, 114, 115, 117  
 Phishing attack, 7, 153  
 Physical layer authentication, 60  
 Physical Layer Security (PLS), 8, 18, 59–75, 162

Privacy, 3, 4, 10, 11, 13, 34, 36, 38, 51, 60, 63–66, 73, 80, 81, 115, 133, 136, 141, 145–155, 159  
 Protection, 11, 32, 36, 50, 63, 66, 114, 133, 161

**R**

Real-time, 1, 4, 8, 10, 20, 32, 114, 146, 149, 150, 153, 161  
 Register, 41, 46–48, 51, 82, 89, 107, 120, 133, 161  
 Route, 20, 23, 24, 39, 138, 161

**S**

Security  
   attacks, 18–19, 23–24, 38, 46, 152  
   challenges, 154–155  
   issues, 153–154  
   solutions, 12, 147, 160  
 Security and Privacy Challenges, 60, 63–66  
 6G technology, 2, 3, 8, 12, 15, 60, 95, 110, 113, 115, 141, 159, 161, 162  
 Smart city, 78–89  
 Smart Contracts, 117  
 Social engineering, 7  
 Supply chain, 32, 102, 114, 115, 117, 161  
 Sybil attack, 8, 9, 11, 39, 40, 153  
 System, 113, 116, 127

**T**

Traceability, 53, 113–129, 161  
 Tracing, 66, 116  
 Transparent, 12, 33, 80, 100, 104, 106, 129, 161  
 Trust, 10, 19, 24  
 Trusted routing, 136–138

**W**

Wireless access, 131–134  
 Wireless networks, 62

**Z**

Zero trust architecture, 11–12