Securing Fog Computing for Internet of Things Applications: Challenges and Solutions

Jianbing Ni, *Student Member, IEEE*, Kuan Zhang, *Member, IEEE*,
Xiaodong Lin, *Fellow, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Internet of Things (IoT) allows billions of physical objects to be connected to collect and exchange data for offering various applications, such as environmental monitoring, infrastructure management, and home automation. On the other hand, IoT has unsupported features (e.g., low latency, location awareness, and geographic distribution) that are critical for some IoT applications, including smart traffic lights, home energy management and augmented reality. To support these features, fog computing is integrated into IoT to extend computing, storage and networking resources to the network edge. Unfortunately, it is confronted with various security and privacy risks, which raise serious concerns towards users. In this survey, we review the architecture and features of fog computing and study critical roles of fog nodes, including real-time services, transient storage, data dissemination and decentralized computation. We also examine fog-assisted IoT applications based on different roles of fog nodes. Then, we present security and privacy threats towards IoT applications and discuss the security and privacy requirements in fog computing. Further, we demonstrate potential challenges to secure fog computing and review the state-of-the-art solutions used to address security and privacy issues in fog computing for IoT applications. Finally, by defining several open research issues, it is expected to draw more attention and efforts into this new architecture.**

***Index Terms*—Fog computing, Internet of Things, edge computing, security and privacy.**

I. Introduction

A GROWING number of physical objects are being connected at an unprecedented rate realizing the idea of the Internet of Things (IoT) [1]. It is the internetworking of various objects and network connectivity that allows these objects to communicate and exchange data, including sensors, smart meters, smart phones, smart vehicles, radio-frequency identification (RFID) tags, personal digital assistants (PDAs) and other items (embedded with electronics, software and actuators) [2]. The interconnection of these devices enables advanced IoT applications, e.g., product tracking, environment monitoring, patients surveillance and energy management, and expands the automation to our daily life. One of the IoT applications is smart home, which enables residents to automatically open their garage when arriving home, start air condition, prepare coffee, and control lights, TV and other appliances. IoT also plays an increasingly important role in other domains, including smart city, smart grid, e-healthcare, intelligent transportation, industrial automation and disaster response. It opens the door to innovations that facilitate new interactions among "things" and human, and provides new opportunities to applications, infrastructures and services that improve the quality of our daily life.

The growth of IoT leads to the generation of large amounts of data, which possess massive computing resources, storage space and communication bandwidth. Cisco predicts that 50 billion devices would connect the Internet by 2020 [3], this number would reach 500 billion by 2025 [4]. The data produced by human, machines and "things" would reach 500 zettabytes by 2019, but the IP traffic of global data centers would only reach 10.4 zettabytes [5]. Then, 45% of IoT-created data would be stored, processed and analyzed upon close to, or at the edge of network [6]. Some IoT applications might need fast response, some might involve private data, which should be stored and processed locally, and some might produce large volumes of data, which could be a heavy burden for networks [7]. Moreover, an increasing number of devices (e.g., smart glasses, smart phones and vehicles) are involved in IoT for collecting and delivering fine-grained data, which may contain multimedia information (e.g., photos, videos and voices). The large amounts of data result in heavy network congestion and complicated processing load on devices and control systems.

With the advance of IoT, fog computing [8], [9] has been introduced to bring the provision of services closer to the end-users by pooling the available computing, storage and networking resources at the edge of the network. It is a decentralized computing infrastructure, which utilizes one or more IoT devices or near-user edge devices to collaboratively perform a substantial amount of communication, control, storage and management. Through the connections between fog nodes and devices, fog computing can reduce the processing burden on resource-constrained devices, reach the

Manuscript received March 7, 2017; revised August 6, 2017; accepted October 6, 2017. Date of publication October 12, 2017; date of current version February 26, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61728102. *(Corresponding author: Xiaodong Lin.)*

J. Ni and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: j25ni@uwaterloo.ca; sshen@uwaterloo.ca).

K. Zhang is with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Omaha, NE 68182 USA (e-mail: kuan.zhang@unl.edu).

X. Lin is with the Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo, ON N2L 3C5, Canada (e-mail: xlin@wlu.ca).

Digital Object Identifier 10.1109/COMST.2017.2762345

latency requirements of delay-sensitive applications and overcome the bandwidth constraints for centralized services [10]. Fog computing offers on-demand services and applications proximate to devices, dense geographical distributed and low-latency responses, resulting in superior user experience and redundancy in case of failure [11].

As a nontrivial extension of cloud computing, it is inevitable that some issues will continue to persist, especially security and privacy issues [12], [13]. Fog computing is deployed by different fog service providers that may not be fully trusted and devices are vulnerable to be compromised. Fog nodes are confronted with various security and privacy threats [14]. The IoT devices have constrained computing, storage and battery resources and are easy to be hacked, broken or stolen. Although the existing solutions in cloud computing could be migrated to address some security and privacy issues in fog computing, it still has its specific security and privacy challenges due to its distinctive features, such as decentralized infrastructure, mobility support, location awareness and low latency. On the other hand, fog computing offers a more secure infrastructure than cloud computing because of the local data storage and the non-real time data exchange with cloud centers. Fog nodes could be represented as proxies for end-devices to perform secure operations, if the devices lack of the sufficient resources to do so [15]. Unfortunately, the security and privacy issues and security resources in fog computing have not been systematically identified. Therefore, to study security and privacy goals of fog computing is quite critical prior to the design and implement of fog-assisted IoT applications.

The research on the security and privacy issues of fog computing for IoT is still in its early stage. In this survey, we take a closer look at the fog-assisted IoT applications, security challenges and state-of-the-art solutions. We start with the evolution from cloud to fog computing, followed by the architecture and features of fog computing. We also introduce the roles of fog nodes, including real-time services, transient storage, data dissemination and decentralized computation, which contribute to various appealing IoT applications in smart city, smart home, smart grid, e-healthcare system, intelligent transportation, etc. Then, we present the security and privacy threats and explore the security and privacy challenges in fog computing. Further, we review the promising techniques to resolve security and privacy issues, analyze how the existing approaches ensure fundamental security goals and protect users' privacy in fog-assisted IoT applications, and show our insights on the achievements and leaving problems to secure fog computing. Lastly, we present open research directions to encourage more efforts on security and privacy preservation in fog-assisted IoT applications.

The remainder of the survey is organized as follows. In Section II, we discuss the difference between this survey and the existing articles. We review the architecture of fog computing and the typical fog-assisted IoT applications in Section III, and discuss security threats in Section IV, followed by security challenges and solutions in Section V. Finally, we discuss future directions in Section VI and conclude our survey in Section VII.

II. Related Works

Recently, many articles have reviewed the state of the art of fog paradigm and discussed the promising IoT applications in a variety of domains, such as smart city, industrial automation, smart healthcare and smart gird. Yi *et al.* [16] discussed the definition of fog computing, introduced several representative application scenarios, including content delivery and caching, real-time video analytics and mobile big data analytics, and identified potential issues in the context of fog computing, such as fog networking, computation offloading and resource management. Sarkar and Misra [17] introduced the theoretical modelling of fog computing and compared its performance with cloud computing in terms of service latency and energy consumption. Varghese *et al.* [18] highlighted the feasibility of fog computing. Compared with the cloud-only model, the fog computing can reduce the average response time by 20% for a user and the data traffic by 90% between the network edge and the cloud. Sarkar *et al.* [19] assessed the suitability of fog computing by characterizing power consumption, service latency, CO_2 emission and cost, and evaluating its performance for an environment with a large number of end-devices demanding real-time services. With the increasing number of applications demanding real-time services, fog computing can decrease by 50% on response delay compared with the traditional cloud computing. Dantu *et al.* [20] discussed the reliability and adaptability of deploying fog computing on Android phones and analyzed the feasibility for the smart phones to be fog nodes. Based on the feasibility of deployment, Varshney and Simmhan [21] discussed the use cases of fog computing in urban surveillance, smart power grid and drones, and compared the distinguished features of fog, edge and cloud computing on resource characteristics, physical access and mobility support. Perera *et al.* [22] surveyed the use case scenarios in smart cities and discussed the common features of fog computing, including dynamic discovery of Internet objects, dynamic configuration, device management, network-level protocols and application-level protocols. Tao *et al.* [23] integrated fog and cloud to build 5G-enabled Vehicle-to-Grid (V2G) networks to facilitate the quick growth of enriched V2G services. Markakis *et al.* [24] exploited the fog and mist paradigms to design a beyond 5G ecosystem where heterogeneous fixed and mobile edge nodes form an archipelago of interconnected islands of resources to support smart cities and industrial automation. Some articles [25]–[27] exploited the smart gateways at the edge of the network to offer several high-level services, such as local storage, real-time local data processing and embedded data analysis, based on fog computing and presented different use cases in smart healthcare IoT systems. Although most of these articles considered the security and privacy threats as the main challenges of fog-assisted IoT applications, they just discussed the security and privacy challenges in a very high level. There is no comprehensive survey to study the security and privacy challenges in detail and review the existing techniques and solutions in fog computing, except [12]. In [12], a detailed analysis on the security threats, challenges and mechanisms was presented to explore potential synergies and venues of collaboration on the research

TABLE I
LIST OF ACRONYMS

ABE	Attribute-based Encryption
AIDC	Automated Identification Data Carriers
AMRC	Advanced Metering Regional Collector
BGN	Boneh–Goh–Nissim
BLS	Boneh–Lynn–Shacham
BYOD	Bring Your Own Device
C-RAN	Cloud-Radio Access Network
CP-ABE	Ciphertext-Policy Attribute-based Encryption
CS	Computer Science
DoS	Denial-of-Service
ECE	Electrical and Computing Engineering
F-RAN	Fog-Radio Access Network
FaaS	Fog as a Service
HIDS	Host-based Intrusion Detection Systems
KP-ABE	Key-Policy Attribute-based Encryption
IoT	Internet of Things
IP	Internet Protocol
MANET	Mobile Ad Hoc Network
NIDS	Network-based Intrusion Detection Systems
PDA	Personal Digital Assistants
RFID	Radio-Frequency Identification
SDN	Software Defined Network
TV	Television
V2G	Vehicle-to-Grid
VANET	Vehicular Ad Hoc Network

of all edge paradigms, including fog computing, mobile edge computing and mobile cloud computing. Roman *et al.* [12] presented the security threats on network infrastructure, edge data center, virtualization infrastructure and user devices, and reviewed eight security challenges and the existing mechanisms to solve these challenges. Different from the above articles, we show a comprehensive analysis on fog computing and its security and privacy issues, including (i) the evolution from cloud computing to fog computing; (ii) the typical fog-assisted IoT applications classified on different roles of fog nodes; (iii) the security and privacy challenges of fog computing and their existing solutions and mechanisms; (iv) the overall insights on the security and privacy issues that the state-of-the-art solutions cannot address due to its unique features; (v) the open research issues and future directions in fog computing, including inherent location privacy leakage, privacy exposure in data combination, detection of rogue fog nodes and IoT devices, and the decentralized and scalable secure infrastructure.

III. Overview of Fog Computing

In this section, we review the evolution from cloud to fog computing, present the architecture and features of fog computing and introduce the fog-assisted IoT applications.

A. Evolution From Cloud to Fog

IoT enables connected devices to collect data and communicate with each other. The IoT devices are generally characterized by physical objects with small size, widely distributed, with limited storage and processing capacity, such that IoT lacks of various important features, including scalability, flexibility, reliability, interoperability and efficiency. Cloud computing [28] has offered a practical solution to address these issues for IoT applications. It enables individuals and enterprises to on-demand use centralized, shared and scalable computing resources and storage services for improving quality of services and reducing management costs [29]. The integration of cloud computing and IoT enables centralized data storage and management, powerful data processing capabilities, scalable resources allocation and rapid applications deployment with minimal cost [30]. The cloud serves as the brain to effectively transform data to insight and drive productive, cost-effective action, resulting to improve the accuracy of decision-making and optimize Internet-based interactions, which dramatically changes the way we live in our daily life.

Cloud-based IoT architecture can be divided into two layers, the top layer and the bottom layer. The top layer is the data storage and control layer, in which the cloud offers an effective approach to manage and composite IoT services and implement IoT applications by exploiting the devices and data collected from these devices. Specifically, the cloud not only provides centralized storage, processing and access for large-scale data, but also offers various applications and services through the virtualization technique to users. It bridges the gap between objects and applications and hides all the complexity and functionalities in implementation. The bottom layer has billions of IoT devices connected with each other and the cloud. The pervasive presence of IoT devices around human enables to measure, infer, understand and reconstruct the environment. These devices may not only include complex devices, e.g., mobile phones, smart glasses, cameras and vehicles, but also comprise daily objects, e.g., appliances, furniture, food, clothing and work of arts [31]. The two layers are connected through communication medium and equipment, such as gateways, routers and bridges, and exchange data via standard communication protocols [32].

Despite the benefits of the integration of cloud computing and IoT are attractive, cloud computing is not a panacea that can address all the problems in IoT. This centralization of resources implies a large separation between IoT devices and the cloud, which results in the increase of the average network latency and jitter [12]. Due to this physical distance, the cloud cannot directly access local contextual information, e.g., local network condition, users' mobility pattern and precise location information. Further, the IoT devices and end-users are unable to access delay-sensitive applications because of communication delay, e.g., smart traffic lights and augmented reality. Therefore, there should be a novel technology to expand the IoT to support delay-sensitive, location-aware and mobility-supported applications.

The concept of fog computing was introduced by Cisco in 2012, which is defined as "*an extension of the cloud computing paradigm that provides computation, storage, and networking services between end devices and traditional cloud servers*" [33]. Fog computing is not a replacement of the cloud for remote data storage and processing, but complement it: fog nodes facilitate the creation of a hierarchical infrastructure, along with the cloud, in which transit data storage and local data analysis are performed at fog nodes, and permanent storage and global analysis are executed at the cloud [12]. The fog nodes are deployed heterogeneously at the edge of network proximate to the devices.

TABLE II
COMPARISON OF CLOUD AND FOG FEATURES [9]

Features	Cloud	Fog
Latency	High (Eventual Consistency)	Low (Locality)
Explicit Mobility	N/A	Yes
Architecture	Centralization	Decentralization
Service Access	Through Core	At the Edge / On Handhold Device
Local Awareness	N/A	Yes
Geographic Distribution	N/A	Yes
Scalability	Average	High
Availability	High	High
# of users/devices	Tens/Hundreds of Millions	Tens of Billions
Content Generation	Central Location	Anywhere
Content Consumption	End Devices	Anywhere
Software Virtualization	Central Corporate Facilities	User Devices/Netowrk Equipment

The initial definition has been extended to shape the features of fog computing. It is extended as "*a scenario where a huge number of heterogeneous (wireless and sometimes autonomous), ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third parties*" [9]. This definition encompasses the key ingredients of fog computing: ubiquity, decentralization, improved network capabilities and better support for cooperation among fog nodes. Under this definition, fog computing is not only an extension of cloud computing, but also a novel paradigm of its own [34]. Fog computing has basic computing, storage and networking resources to support new services and applications, and users can lease the facilities and resources provided by fog nodes to access the services and applications. "Fog as a Service (FaaS)" becomes an exciting new opportunity for the service providers, in which a fog service provider builds an array of fog nodes at geographic locations to offer certain services to many users from vertical markets. Therefore, fog computing can be deemed as a service model, in which data can be stored, analyzed and processed by fog nodes within the network, rather than in a centralized cloud [17]. Table II compares the features of the cloud and fog computing to clarify how their differences [9].

To deal with the increasing number of connected things and emerging applications in IoT, fog computing smartly orchestrates and manages computing, storage and networking resources provisioned at the network edge. By utilizing these resources close to end users, fog computing provides a range of new applications and services, such as hierarchical data analytics and smart infrastructure management (e.g., smart traffic lights, smart wind farms and smart parking reservation), to help to address challenges of cloud-based services with high delay and constrained bandwidth and IoT devices with limited resources. In fog-assisted IoT applications, the time-sensitive data is stored, processed and analyzed on fog nodes close to the devices for providing real-time control and analytics within millisecond response time [8]. If permanent storage and global analysis are required, fog nodes periodically send data summaries to the cloud. Therefore, whether an IoT device is served by the cloud, the fog or an adequate combination of both is determined by the requirements and constraints of IoT applications [35]. Hence, fog computing is not a competitor of the cloud in IoT applications; on the contrary, it is envisioned as a perfect complement for a large number of applications and services, in which the cloud computing is insufficient to satisfy their requirements.

The efforts on building a set of standardized fog computing architectures have been made in both industry and academia. Cisco proposed IOx service, which combines IoT applications within fog nodes and Cisco IOS to achieve rapid, secure and reliable services. The IOx application framework provides consistent data management and service hosting across network infrastructure products, including Cisco routers, switches and computing modules. Cisco also provides Fog Director to manage large-scale production deployments of IOx-enabled fog applications. Moreover, ISO/IEC 20248 [36] specifies a method whereby data stored within a barcode and/or RFID tag can be identified, read, interpreted and verified by fog nodes using Automated Identification Data Carriers (AIDC). The OpenFog Consortium was established to create an open architecture for fog computing, build operational models and testbeds, define and advance technology and promote business development through a thriving OpenFog ecosystem. In addition, the integrated architecture of fog computing and radio access network, called F-RAN [37], has been introduced to reduce the service latency by exploiting local signal processing and computing, cooperative resource management, and distributed storing/caching capabilities at the network edge [38]. Meanwhile, another architecture deploying fogs on top of cloud-radio access network (C-RAN) has been designed based on a software-defined network (SDN) approach to support the cloud-fog interoperation for 5G wireless network [39]. With the proposed architectures and frameworks, the applications of fog computing have been explored to support various IoT systems and services, including augmented reality [16], pervasive health monitoring [40], energy management [41] and website categories [42].

Edge computing [7], [43] is a similar concept with fog computing, which also pushes the services, storage and computing resources away from central servers to the network edge. It can reduce the communication overhead between the central servers and network edge by performing data analysis and knowledge discovery at or near the data sources [44], [45]. The edge network is defined as opposed to the core network, whereas fog computing is an

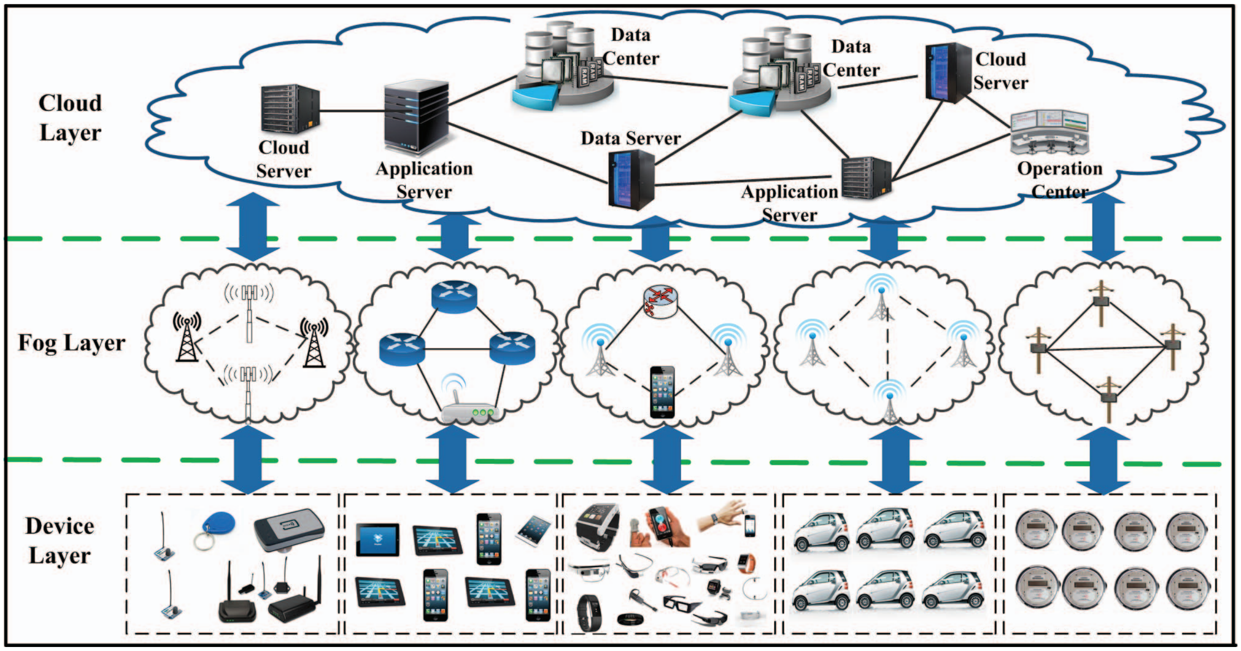


Fig. 1. Three-Layer Architecture of Fog Computing for IoT Applications: From Cloud, Fog to IoT Devices.

end-to-end horizontal architecture that distributes control, computing, networking and storage sources and services along the cloud-to-device continuum. In this sense, the differences between fog and edge are as follows [46]. First, fog computing is decentralized that enables the management and orchestration of resources distributed in the network, anywhere from the central servers to end-devices. Second, fog computing provides a seamless continuum of services from cloud to devices, rather than treating the network edges as isolated computing platforms. Third, fog computing is a horizontal platform that supports various IoT applications, even Internet-of-Everything, through the wired or wireless access networks, including but not limited to the cellular network. While the edge computing aims to offer computing capability at the edge of cellular network, e.g., the base station [47].

B. Architecture of Fog Computing

Fog computing architecture can be divided into two categories, Cloud-Fog-Device framework and Fog-Device framework. The former consists of three distinct layers, namely, the device layer, the fog layer and the cloud layer, as depicted in Fig. 1, and the latter has two layers, the device layer and the fog layer. These layers are arranged in an increasing order of computing and storage capabilities. To achieve inter layer and cross layers' communications, various communication technologies are used to connect each entity, including wired communication (e.g., Ethernet, optical fiber), wireless communication (e.g., Bluetooth, LTE, ZigBee, NFC, IEEE 802.11 a/b/c/g/n, satellite links) or a combination of both [48]. The visualization technologies, such as network functions virtualization and software-defined network [9], are used to achieve network virtualization and traffic engineering. Each layer is scalable and flexible, indicating that it can be enhanced to involve numerous entities if the demand arises. All three layers can be connected with public authorities (e.g., certificate authority, key generation center, judgers and police) through direct or indirect links. In case of any threat is detected, the public authorities immediately join to handle the accident.

The device layer has two types of devices, mobile IoT devices and fixed IoT devices. The mobile IoT devices are carried by their owners, such as wearable devices (e.g., fitness trackers, wearable cameras, smart clothes and sports bracelets) and mobile smart devices (smart phones, smart watches, smart glasses, vehicles) [48]. All devices belonging to the same owner can form a group and communicate with each other using wireless ad hoc networks. The fixed IoT devices (e.g., sensors and RFID tags) are pre-deployed in specific areas or on particular products to fulfill pre-defined tasks (e.g., products tracing, forest fire detection and air quality monitoring). These IoT devices have limited computing and storage resources, and restricted bandwidth in general [49], such that they cannot respond emerging events. Their responsibility is to collect raw data and report them to the upper layer. For example, to build smart city, there will be numerous IoT devices both fixed and mobile installed around the city, and connect with each other and collect data on all aspects of the city.

The fog layer consists of network equipment, such as routers, bridges, gateways, switches and base stations, augmented with computational capability, and local servers (e.g., industrial controllers, embedded servers, mobile phones and video surveillance cameras). These devices, called fog nodes in fog computing, can be deployed anywhere with network connections: in a smart phone, on a factory floor, on a roadside unit, in a vehicle or on top of a power pole. The fog nodes are hierarchically distributed between the IoT devices and the cloud servers in the Cloud-Fog-Device framework or above the IoT devices in Fog-Device framework. This layer tends to extend the cloud computing to the network edge. It has certain computing and storage prowess and autonomy to reduce the processing load on resource-constrained IoT devices. Apart from conventional communications (e.g., package forwarding and routing), some real-time and latency-sensitive applications can be relegated from cloud servers to fog nodes. Since the applications are located in the fog nodes only one/two-hop away from devices, they possess regional knowledge about the devices and their owners (i.e., users), e.g., local network condition, users' mobility pattern and precise location information. In Fog-Device framework, the fog nodes cooperatively offer various services without the involvement of cloud servers, e.g., decentralized vehicular navigation [50], indoor floor plan reconstruction [51], smart traffic lights [52] and local content distribution [53]. In Cloud-Fog-Device framework, the fog nodes provide transient storage and real-time analysis on the data collected by IoT devices and periodically send data summaries to the cloud through the forwarding of other fog nodes located at higher levels in the network hierarchy.

The cloud layer in Cloud-Fog-Device framework is a consolidated computing and storage platform that provides various IoT applications from a global perspective. The cloud has significant storage space and computing resources and is accessible for users at anytime and from anywhere, as long as their devices are connected to the Internet. It utilizes virtualization technology to achieve the isolation of distinct users' data and IoT applications, such that these applications can independently and concurrently provide different services to distinct users. The cloud receives data summaries from various fog nodes, and performs global analysis on the data submitted by fog nodes and the data from other sources to improve business insight in IoT applications [8], such as smart power distribution [54], health status monitoring [33] and network resource optimization [55]. In addition, the cloud also sends policies to the fog layer to improve the quality of latency-sensitive services offered by fog nodes.

C. Features of Fog Computing

Fog computing is a distributed framework that offers IoT applications at the edge of the network by leveraging edge resources. The major feature of fog computing is to tackle the IoT data locally by utilizing the fog nodes placed near users to bring about the convenience of data storage, computation, transmission, control and management. Compared with cloud computing, fog computing has five distinguished features [52] as follows:

- *Location Awareness [52]:* The location of fog nodes can be traced actively or passively to support devices with rich services at the network edge. Fog computing dedicates on local IoT applications accessible for the devices at certain areas via specific fog nodes. Therefore, it is aware of the devices' regions based on the locations of fog nodes.
- *Geographic Distribution [52]:* The fog nodes are deployed at certain positions, such as along highways and roadways, on cellular base stations, on a museum floor and at a point of interest. The reason is to guarantee that the fog nodes can receive high-quality data stream from IoT devices, even when these devices pass between two fog nodes.
- *Low Latency [52]:* Thanks to the computing and storage resources, fog nodes can provide computation services and make decisions based on local data without the cloud. Since the fog nodes are proximate to the IoT devices, the latency of the response is much lower than that made by the cloud.
- *Large-Scale IoT Applications Support [33]:* Fog computing is introduced to support large-scale IoT applications, which bring heavy management overhead to the centralized cloud. In large-scale IoT applications, such as environment monitoring, power grid management, water treatment management and climate change monitoring, fog computing has its prowess and autonomy to manage billions of IoT devices.
- *Decentralization [9]:* Fog computing is a decentralized architecture that there is no centralized server to manage resources and services. The fog nodes self-organize to cooperatively provide real-time services and IoT applications to users.

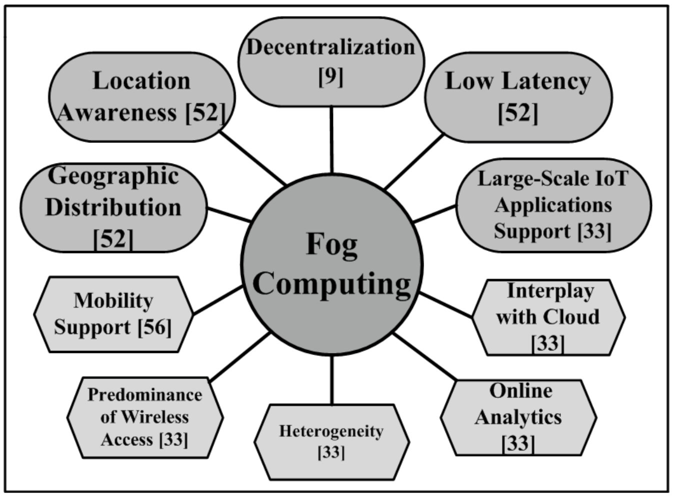


Fig. 2. Characteristics of Fog Computing.

In addition, fog computing has several general characteristics, including mobility support, predominance of wireless access, heterogeneity, online analytics and interplay with the cloud [33]. The Fig. 2 illustrates all the characteristics of fog computing, including distinguished and general features.

D. Fog-Assisted IoT Applications

In this subsection, we introduce the roles of fog nodes, including real-time services, transient storage, data dissemination and decentralized computation, and exhibit some typical applications as examples to demonstrate the prospects of fog computing in IoT, as shown in Fig. 3.

1) Real-Time Services for Fog: With computational and storage capabilities, fog nodes behave as a surrogate of cloud or a private cloud close to IoT devices, handling local real-time computation services. Specifically, the fog nodes deployed at the network edge offer IoT applications and services, and

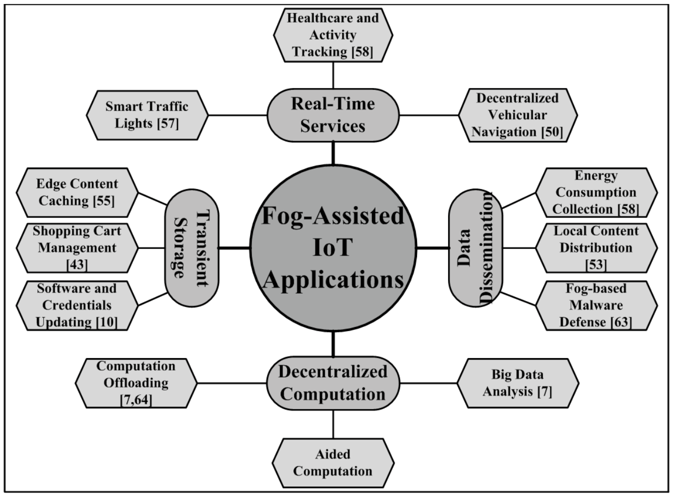

Healthcare and Activity Tracking [58]
Smart Traffic Lights [57]
Real-Time Services
Decentralized Vehicular Navigation [50]
Edge Content Caching [55]
Shopping Cart Management [43]
Software and Credentials Updating [10]
Transient Storage
Fog-Assisted IoT Applications
Data Dissemination
Energy Consumption Collection [58]
Local Content Distribution [53]
Fog-based Malware Defense [63]
Computation Offloading [7,64]
Decentralized Computation
Big Data Analysis [7]
Aided Computation

Fig. 3. Roles and Fog-Assisted IoT Applications.

receive data from the IoT devices to make decisions and control the activities of these devices within millisecond response time [8]. Therefore, many delay-sensitive IoT applications can be built to achieve fast decision-making based on collected local data. We showcase some examples of fog-assisted IoT applications, in which fog nodes offer real-time control and fast decision-making for users.

Smart Traffic Lights [57]: At a crossroads, a fog node detects the flashing lights of an ambulance or a police car using video cameras and automatically changes traffic lights to open the lines for fast driving through. The fog nodes also can use the deployed sensors on roads to detect the presence of pedestrian and bicycles and measure the driving speed and distances of approaching vehicles, and thereby change traffic lights to make convenient to pedestrian and bicycles. In addition, neighboring fog nodes coordinate to send warning signals to the approaching vehicles to avoid collision. Therefore, the smart traffic lights can do rapid response to the approaching pedestrian, bicycles and vehicles based on the collected data from video cameras and sensors. Therefore, smart traffic lights can contribute to create green traffic for vehicles and pedestrian.

Healthcare and Activity Tracking [58]: Fog computing can play an important role in smart e-healthcare systems. For example, fog computing can be used to detect, predict and prevent falls for stroke patients and sudden cardiac death for cardiovascular patients. The fog nodes (i.e., smart phones) can start alarms and send warning signals to hospitals once they detect sudden events happening, such that the chance of survival can be dramatically improved. Moreover, fog nodes can provide automatically personal care for patients in hospitals. They can collect data from sensors deployed on the body of patients to discover emergent events and thereby take actions timely to give intensive care. Therefore, smart e-healthcare systems can definitely shorten the time of first aid and improve the success rate in rescuing patients.

Decentralized Vehicular Navigation [50]: Real-time traffic information is essential to monitor congestion and navigate for drivers. How to collect real-time road conditions and respond proper paths to drivers timely is a critical problem in navigation systems. Fog computing plays an important role in local data collection and navigation result response. Specifically, fog nodes, which are upgraded roadside units that stretch to have computational capabilities and storage spaces, can maintain traffic information reported by the driving vehicles in their coverage areas. If a fog node receives a navigation request from a vehicle, it can cooperate with other fog nodes to generate a proper driving path for the querying vehicle to its destination, and rapidly return the path to the querying vehicle. Thus, the vehicle can enjoy real-time navigation services flexibly and take actions to avoid being stuck in traffic congestion promptly.

In addition, fog computing enables other low-latency and autonomous local IoT applications, such as home energy management [41], augmented reality and gaming [58], indoor location and navigation [59] and network resource management [60].

2) Transient Storage for Fog: Fog nodes are equipped with certain storage resources to temporarily maintain the data collected by IoT devices. This transient data storage not only allows users to maintain the data frequently accessed, but also achieves rapid data update in a flexible and efficient way. Generally, the fog nodes keep the data for a short time period, such as 1-2 hours, and then discard the provisional data or send the ultimate data to the cloud. Therefore, the transient storage on fog nodes can significantly reduce the communication overhead between the fog nodes and the cloud, and dramatically shorten the response time of data access and update. Several fog-assisted IoT applications can be realized, such as edge content cashing, temporary files maintenance, software and credentials updating and shopping cart management.

Edge Content Caching [55]: Fog nodes can provide content caching services for Web content access. When a user accesses content on cloud servers, the user can get served directly from the fog nodes if this content has been accessed by other users and cached on fog nodes, such that the delay of content accessing can be reduced. A platform [42] has been proposed to classify websites and define caching priority in a fog node. By doing so, Web response time can be reduced by 20%. Another application is to use on-board fog nodes over inter-state buses to provide entertainment services to passengers [57]. Specifically, the fog nodes deployed on buses can offer on-boarding video streaming and social services to passengers through WiFi and refresh the cached content to cloud servers via cellular network. Therefore, edge content caching can improve users' experience and reduce network overhead for content delivery.

Shopping Cart Management [43]: Fog computing can improve users' experience in online shopping. Traditionally, a user's shopping cart is maintained on the cloud, the update of shopping cart view may take a long time depending on network bandwidth and server loads. This delay may be longer for mobile devices due to the low bandwidth of wireless network. With fog computing, when a user accesses the shopping cart, the fog node caches it and performs the updating operations until the user loges out the cart. To keep the consistency of shopping cart, the fog node sends the final version to the cloud to permanently update the shopping cart. Therefore, fog-based shopping cart management can reduce the delay of shopping cart update and increase users' satisfaction.

Software and Credential Updating [10]: With storage space on fog nodes, there is no need for IoT devices, especially resource-constrained devices, to perform complicated protocols or procedures to acquire and update software and security credentials from remote clouds. The fog nodes can automatically acquire and cache the required software, keys and credentials on behalf of IoT devices and the IoT devices can retrieve the needed content from the fog nodes. For example, electronic control units on a vehicle no longer need to communicate with the remote off-board control systems to acquire or update software and credentials. Instead, a fog node (e.g., on-board unit, on-board controller) on the vehicle can acquire the software and credentials in advance and distribute them to the electronic control units. Thus, the communication bandwidth of IoT devices can be saved and the IoT devices do not need to perform complex operations to acquire these contents.

3) Data Dissemination for Fog:

As intermediate nodes in network, fog nodes take conventional communication functions, such as data aggregation, package forwarding and routing. They submit the data collected by IoT devices to the cloud, and distribute the data received from the cloud to the IoT devices. This two-way data delivery builds network connectivity between the cloud and IoT devices. With storage and communication resources on fog nodes, local data collection and content distribution become convenient and flexible. Moreover, the fog nodes can perform simple processing on the received data. Specifically, the fog nodes can not only deduplicate the data collected by numerous IoT devices to improve communication and storage efficiency, but also select proper audiences to increase the accuracy of content distribution. Overall, the involvement of fog nodes can significantly optimize large-scale data collection and content distribution services.

Energy Consumption Collection [58]: In smart grid, the operation center frequently collects the power consumption of every household, which is used to optimize energy generation, distribution and billing. Conventionally, smart meters equipped on houses independently report consumption measurements to the operation center, which causes unacceptable communication overhead between smart meters and the operation center. Advanced Metering Regional Collector (AMRC) can act as a fog node to aggregate the consumption measurements and report the sum of power consumption in its residential area to the operation center, such that the cost on communication bandwidth between AMRC and the operation center can be significantly reduced. In addition, AMRC can also transiently store the individual consumption and calculate the monthly cost on power for each household based on dynamic electricity price to achieve real-time load monitoring and dynamic billing, simultaneously. Therefore, the fog nodes can play an important role on energy consumption collection for smart grid.

Local Content Distribution [53]: Fog nodes can distribute local content to IoT devices on behalf of local servers. For example, fog nodes can broadcast traffic information, advertisements and other relevant information (e.g., restaurants, motels and gas stations) to the driving vehicles on roads. Another example is the parking navigation service in a large parking lot [61]. In this service, a fog node deployed at the entrance of the parking lot, can collect data from parking lot video cameras and sensors on parking spaces, analyze the data to generate the parking information, including parking lot map and vacant parking spaces, for providing parking navigation for the vehicles entering the parking lot, predicting store traffic flow and optimizing checkout staffing. Moreover, fog nodes on road can cooperatively offer parking navigation service to the vehicles for finding or reserving accessible parking spaces near their destinations [62]. Therefore, with the involvement of fog nodes, local content distribution becomes powerful and brings convenience to our daily life.

Fog-based Malware Defense [63]: Security protection mechanisms can be implemented on fog nodes to eliminate the requirements of software and hardware installment, management and update on IoT devices. For example, malware defense is a resource-consuming task for IoT devices that requires large storage space, timely processing and global intelligence. In fog computing, malware defense can be moved to a cluster of fog nodes close to the protected IoT devices and each fog node has certain threat defense capability to high-fidelity determine whether the packages contain malware or not. As a result, the fog nodes can make efficient use of shared resources to improve complementary defense capabilities, by preventing attackers from disrupting security operation. Furthermore, a cluster of fog nodes can collaboratively detect the compromised devices, assess the potential impacts of infected files and clean up these files.

4) Decentralized Computation for Fog:

With storage and computing resources, it is possible for multiple fog nodes to cooperatively perform decentralized data computation. Specifically, the fog nodes can not only collaboratively take computation tasks for the cloud, but also assist users to perform heavy computational operations on behalf of proxies. Thus, either the cloud or IoT devices can be free from heavy computational tasks.

Computation Offloading [7], [64]: In cloud computing, cloud servers have to perform heavy computational tasks for maintaining services and applications for users. Although they are assumed to have infinite computational capabilities, this centralized computation model is not energy-efficient and causes large delay on service response. Fog computing extends the computing resources to the network edge, such that the fog nodes are enabled to distributively execute the computation for cloud servers [43]. By doing so, the cloud servers can be free from heavy computational tasks, and the latency of service response can be reduced meanwhile. In addition, partial computational tasks on fog nodes can be further offloaded to smart devices to take full advantages of computing resources on these devices [65]. Therefore, computation offloading is critical to improve resource utilization, save energy consumption and reduce response delay.

Aided Computation: Because of the limited computational capability, it is hard for IoT devices to perform complicated computations, such as image processing, video clips and cryptographic operations. Therefore, how to do complex operations on resource-constrained devices is critical for users. With the involvement of fog nodes, some time-consuming computational operations can be moved to the fog nodes. For example, to verify the validity of a digital signature, a fog node can aid IoT devices to execute bilinear pairing, which is too time-consuming to be computed by the IoT devices [66]. In cloud data outsourcing [67], the fog nodes can help users to compute homomorphic authenticators to reduce the computational overhead on users' devices. In summary, fog nodes can undertake complex computing tasks for IoT devices to improve the computational efficiency.

Big Data Analytics [7]: With the increasing volumes of data generated by IoT devices, data analytics becomes a huge challenge for the cloud, specifically for latency-sensitive applications, such as finding a missing child via video analytics. Nowadays, video cameras are widely deployed at each crossroads and populated areas (e.g., airports, squares, plazas, shopping malls and railway stations). When a child is missing, it is possible to find the trajectory by using video cameras. Nevertheless, the data on video cameras is usually maintained locally due to privacy concern or traffic cost, which causes huge difficulty to leverage these data to find the missing child. Moreover, even if the data can be accessed by the cloud, the data uploading and searching have to take a long time, which is intolerable for finding the missing child. By utilizing fog computing, the request of missing child searching can be performed on the video cameras and only the result would be reported to the cloud. In this paradigm, the result can be obtained much faster than that of cloud-based searching. Therefore, local big data analytics is quite important for time-sensitive IoT applications with large-scale data.

We have mentioned various fog-assisted IoT applications according to different roles of fog nodes, which contribute to the development of smart city [68], and its critical components, including smart transportation, smart grid, smart e-healthcare and other related aspects. In Table III, we illustrate the fog-assisted IoT applications discussed in literatures.

TABLE III
FOG-ASSISTED IoT APPLICATIONS IN SMART CITY

Components	Fog-assisted IoT Applications
Smart Transportation	Traffic Management and Surveillance [24], Decentralized Vehicular Navigation [50], Smart Traffic Lights [57], Inter-state Bus Entertainment [57], Parking Sharing and Management [71], Road Surface Condition Monitoring [72].
Smart Grid	Home Energy Management [41], Microgrid Energy Management [41], Energy Consumption Collection [58], Smartphone Energy Saving [73].
Smart Healthcare	Wearable Big Data Analysis [25], Speech Treatments of Patients with Parkinson's Disease [25], Smart E-health Gateways [26], Fall Detection for Stroke Patients [40], Prediction of Sudden Cardiac Death [43], Patient Activity Tracking [58], Patient Care in Hospitals [58], Human Health Monitoring [74].
Others	Shopping Cart Management [7], Software and Credential Updating [10], Smart Industry Automation [24], Fog-radio Access Networks [38], [55], Finding A Missing Child [43], Local Content Distribution [53], Edge Content Caching [55], Indoor Location and Navigation [59], Fog-based Malware Defense [63], Fog-based Crowdsensing [75], Emergency Alert Service [76], Fog-empowered Anomaly Detection [77], Fog-based Proximity Detection [78], Fog-based Location Verification [79], Fog-based Vehicular Data Scheduling [80].

IV. SECURITY THREATS OF FOG COMPUTING

Cloud computing is vulnerable to be hacked by external attackers because of the centralized data storage and computing framework. The major cloud computing vendors, such as Google, Amazaon and Yaho, successively appeared large-scale data leakage accidents. Cloud security has become an important factor restricting the development of cloud computing. As a non-trivial extension of cloud computing, fog computing is considered to be a more secure architecture than cloud computing due to the following reasons: First of all, the collected data is transiently maintained and analyzed on local fog node closest to data sources, which decreases the dependency on the Internet connections. Local data storage, exchange and analysis make it difficult for hackers to gain access to users' data. Secondly, the information exchange between the devices and the cloud no longer happens in real-time, so that it is hard for eavesdroppers to discern the sensitive information of a specific user.

However, fog computing cannot be deemed to be secure, since it still inherits various security risks from cloud computing. In general, the fog nodes and clouds are honest-but-curious. They are deployed by fog and cloud vendors to offer specific services honestly to users for their own benefits. On one hand, for monetary reasons, they may not deviate from the protocols agreed upon among the ones involved, on the other hand, they may snoop on the content of maintained data and the personal information about data owners. Further, the employees in fog or cloud service providers might acquire personal information about users, resulting in the privacy leakage for users. In addition, the fog nodes or cloud servers may become the major targets of hackers that use any possible method to reach their own goals unscrupulously. Therefore, the fog nodes or cloud servers could be honest-but-curious, even malicious. Specifically, an attacker may launch the following attacks [12] to disrupt the fog computing.

(1) Forgery: Malicious attackers may not only forge their identities and profiles, but also generate fake information to mislead other entities. In addition, the network resources, such as bandwidth, storage and energy, would be excessively consumed by the faked data.

(2) Tampering: A tampering attacker could maliciously drop, delay or modify transmitting data to disrupt fog computing and degrade its efficiency. It is difficult to detect some

tampering behaviors, since the wireless channel condition and user mobility may result in the transmission failure and delay.

(3) Spam: Spam data refers to the unwanted content, such as redundant information, false collected data from users, which is generated and spread by attackers. The spam would result in the unnecessary network resource consumption, misleading social friends, and even privacy leakage.

(4) Sybil: Sybil attackers either manipulate fake identities or abuse pseudonyms in order to compromise or control the effectiveness of fog computing. For example, they could generate incorrect crowdsensing reports, such that the crowdsensing results may not be trustworthy. In addition, Sybil attackers could invade legitimate user's private information.

(5) Jamming: An attacker deliberately generates a huge of bogus messages to jam communication channels or computing resources, such that other users are prohibited from normal communication and computation.

(6) Eavesdropping: Malicious attackers listen on communication channels to capture transmitting packets and read the content. This type of network attack is quite effective if the data lacks of encryption.

(7) Denial-of-Service: An attacker disrupts the services provided by fog nodes to make them unavailable to its intended users, by flooding the target fog nodes with superfluous requests. This attack consumes network resources to prevent the requests from legitimate users from being fulfilled. A fog node is pretty vulnerable to denial-of-service (DoS) attacks compared with the cloud as its available resource is limited.

(8) Collusion: Two or more parties collude together to deceive, mislead, or defraud other legal entities or obtain an unfair advantage. In fog computing, any two or more parties can collude to increase their attack capability, such as several fog nodes, IoT devices, IoT devices with the cloud, or fog nodes with IoT devices.

(9) Man-in-the-Middle: A malicious attacker stands in the middle of two parties to secretly relay or modify the exchanging data between these parties, however, these two parties believe that they are directly communicating with each other.

(10) Impersonation: A malicious attacker pretends a legitimate user to enjoy the services provided by fog nodes, or impersonates a legitimate fog node to offer fake or phishing services to users.

In addition, privacy is a critical issue in fog computing as the users' sensitive data is involved in the collection, transmission, processing and sharing. Data owners are not willing to expose their privacy to others, but the leakage of privacy is oblivious. A user's privacy may include four aspects, that is, identity privacy, data privacy, usage privacy and location privacy.

(1) Identity Privacy: The identity of a user includes the name, address, telephone number, visa number, license number and public-key certificate that any information can link to a specific user. Users' identities are vulnerable to be disclosed from the information submitted to fog nodes for authentication.

(2) Data Privacy: Users' data may be exposed to an untrusted party when they are maintaining on fog nodes, and transmitting between two parties. By analyzing these data, various sensitive information can be obtained, such as a user's preference, occupation, address, health status and political inclination. For example, a medical record poses the patient's health status, and a vote exposes the voter's political intention.

(3) Usage Privacy: Usage privacy mainly refers to the usage pattern with which a user utilizes the services offered by fog nodes. For example, the readings of a smart meter may disclose the living habits of a family, such as at what time the residents go to sleep, and at what time they are not at home, which absolutely violates residents' privacy.

(4) Location Privacy: Currently, massive applications on mobile devices collect users' location information. It seems that location privacy is a kind of privacy that we have to sacrifice in order to enjoy online services, such as navigation and location-based services. However, location privacy preservation is critical indeed. From the collected location information, an attacker is able to identify a user's trajectory, identity, points of interest, etc., resulting in the exposure of users' privacy. Unfortunately, it is of difficulty to protect users' locations in fog computing. As a user can access the services provided by the nearest fog node using IoT devices, this fog node can infer that this user is nearby and far from other fog nodes. Moreover, if a user accesses multiple services offered by the fog nodes deployed at different locations, it may disclose the path trajectory to the fog nodes.

The IoT devices are the major sources of security threats of fog computing. With the increasing number of connected IoT devices, the vulnerability of IoT devices exacerbates users' concerns on security and privacy. Due to the lack of sufficient security protection, IoT devices are vulnerable to be hacked, broken or stolen. These compromised devices can become powerful and distributed sources to corrupt normal services. In October 2016, an Internet company, Dyn, was crippled by massive distributed DoS attacks from a large number of unsecured Internet-connected devices, such as home routers and surveillance cameras, which repeatedly disrupt the availability of Twitter, Netflix, Amazon and PayPal [69]. IoT botnets will remain a huge threat towards the network services. Besides, illegal network access frequently happens in a public environment. Kaspersky Lab detected almost 3.5 million pieces of malware on more than 1 million user devices in 2014 [70]. The malware steals credentials to gain access to the target-hosted networks and services. In summary, the IoT devices have been a new weapon for hackers, which brings enormous security risks towards the availability and reliability of IoT services, and thereby triggers numerous security and privacy threats towards the infrastructure of fog computing and cloud computing.

Due to the security and privacy threats in place, as shown in Fig. 4, it is crucial to build efficient and effective secure and privacy-preserving mechanisms in fog computing. Without appropriate security and privacy protection, users may be unwilling to participate in IoT applications, which impedes the success of fog computing.

V. Securing Fog: Challenges and Solutions

In this section, we further discuss security and privacy challenges of fog computing and review the existing methods that can be used to address these challenges. We also show our

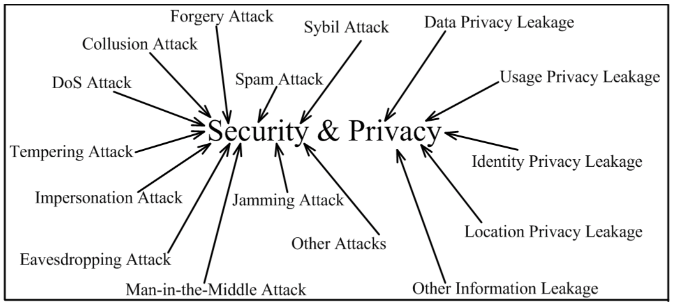


Fig. 4. Security and Privacy Threats in Fog Computing [12].

insights on these approaches and demonstrate the challenges on addressing the security and privacy issues of fog computing.

A. *Challenges and Solutions on Real-Time Services*

Fog computing provides a variety of low-latency IoT applications to users. Due to the existence of security threats in fog computing, the users cannot set their minds at ease to enjoy real-time services offered by fog nodes, if there is no effective security or privacy protection mechanism to guarantee the security or prevent users' privacy leakage in fog-assisted IoT applications. Therefore, how to build secure real-time services becomes extremely critical in fog computing. We discuss several security challenges and introduce some promising techniques to overcome these challenges.

1) Identity Authentication: Fog, cloud service providers and users in different trusted domains offer and access the real-time services, respectively. This situation brings numerous security challenges towards users' data and IoT services, as it is hard to ensure all involved entities trusted. Each user should be authenticated to guarantee the authenticity and creditability before accessing these services. Without sufficient security guarantees, it would be easy for external attackers to target the resources of services and infrastructure with impunity. For example, an external attacker may successfully pretend one of the legitimate parties to access services, and an internal attacker would not leave a trail of evidence for misbehavior detection behind their malicious activities. Therefore, it is necessary to explore identity authentication mechanisms to prevent illegitimate service access.

Currently, many identity authentication schemes have been proposed [83]–[86] to realize the identity confirmation in cloud-based or ad-hoc networks. Several efficient authentication mechanisms [87], [88] have been designed to support IoT services (e.g., face identification) and build secure framework based on fog computing. Nonetheless, these schemes do not consider the mobility of IoT devices. In fog computing, fog nodes collaboratively provide real-time services to users in large areas and the users may travel from the coverage region of a fog node to another. If each fog node independently authenticates the users when service access, the latency may be unacceptable in real-time services when the number of users increases. Cooperative authentication schemes [89], [90] are widely used to reduce authentication overhead and shorten the authentication delay for individuals. These schemes employ the users' cooperation to eliminate redundant authentication efforts on the same message by different users. Nevertheless, since users may go through several coverage areas of fog nodes in a high speed, it is also necessary to design efficient authentication schemes supporting the cooperation of fog nodes to verify the users' identities or secure sharing of authentication results among multiple fog nodes. Otherwise, the delay of real-time services would be intolerable, as each fog node has to authenticate the users' identities before offering services.

In some real-time services, users are reluctant to disclose their identities to fog nodes during authentication. For example, the drivers on road are unwilling to expose their identities; otherwise, an attacker would be easy to localize their current position and draw the daily trajectory of a specific driver. The wearable devices, such as smart glasses, smart watches and smart bracelets, also disclose the location and identity information to curious fog nodes. Anonymous authentication enables the fog nodes to confirm the authenticity of a user without exposing the user's identity. Due to this promising property, several anonymity techniques, e.g., pseudonyms [91], group signatures [92] and k-anonymity [93], are widely used to break the links between users' identities and authentication messages. With these anonymity approaches, the curious parties, including fog nodes, the cloud and other users, cannot distinguish the target users. Nevertheless, once the identities are hidden, the network connections between the fog nodes and devices have to be kept to ensure that the authentication results and service responses can be delivered to the target devices successfully. Unfortunately, it is hard to keep the connections alive because of the mobility of users in reality, especially for drivers in a high speed. To address this issue in vehicular communications, Ni *et al.* [50], [62] utilized the data retrieval approach to acquire the results of parking navigation services from roadside units for drivers. Due to the distinctive features of fog computing and diverse requirements of real-time services, whether this method can be used in fog computing is still an open problem. Furthermore, the users may be re-identified under de-anonymization attacks [94], as the anonymous data is cross-referenced with other data from different sources, e.g., social graphs, trajectory and purchase records. Therefore, the anonymity technique is not sufficient for protecting users' identities and privacy in fog computing.

Summary and Insights: The decentralization of fog computing, the low latency of real-time service and the mobility of users bring a huge obstacle on the realization of identity authentication. Cooperative authentication and other authentication approaches with collaboration of fog nodes deserve to focus on to reach the latency requirement for real-time services and support the mobility of users. In terms of anonymous authentication, the users' mobility and de-anonymization attacks are the major challenges for fog-assisted IoT applications.

2) Access Control: In real-time services, different users or IoT devices have distinctive rights to access each service. The existence of an authorization architecture is equally important for real-time services in fog computing. If there is no authorization mechanism in place, anyone can impersonate administrators, having powerful access rights to control services and infrastructure. External attackers would be able to arbitrarily access personal accounts and temper normal services. Therefore, it is crucial to deploy authorization mechanisms in every trust domain to permit administrators to define and enforce access policies in each domain. These policies enable to handle the credentials for the entities, who are deemed to be trusted based on access policies. Further, in the definition of authorization policies, various users' factors should be also taken into account, such as occupation, trustworthiness, geographical location and resource ownership.

Currently, role-based access control policy [95] is widely used in traditional Web services to allow the administrators to regulate access rights to network resource based on the roles of individuals, e.g., managers, members or assistants. Another popular access control policy is attribute-based access control [96], based on attribute-based encryption. In this policy, users with certain attributes have the access rights to resource and services if their attributes satisfy the pre-defined attribute-based policy. For example, the user with attributes ("Ph.D. student", "ECE department") can satisfy the access policy (("Ph.D. student" OR "Master student") AND "ECE department"), and a user with attributes ("Ph.D. student", "CS department") cannot access. However, fog computing is a fully distributed paradigm and fog nodes collaboratively provide services to the users located in their coverage areas, federated and distributed access control architecture is needed to define access policies for mobile users, who may travel from one fog node to another.

Even if each user can access real-time services based on the pre-defined access policy, how to manage the multiple devices owned by a user is still challenging. Nowadays, a person has several devices connecting to the Internet, e.g., laptops, tablets, smart phones, wearable devices and vehicles. All these IoT devices can be used to access real-time services. In this case, fog nodes do not care which device is used to access services, but the user who accesses it. Furthermore, the consistency of access policy should be guaranteed when a user employs different devices to access the services. However, from the perspective of users, they usually concern which device should be used to access the services. Therefore, it is crucial to deploy sufficient device management mechanisms for users and key management mechanisms for fog nodes, simultaneously. Several mobile device management protocols [97], [98] have been implemented to support access control for multiple mobile devices in bring your own device (BYOD) environment, but seldom [99] can achieve device and key management for users. Hence, to design device and key management schemes supporting sufficient access control in fog computing deserves further investigations and efforts.

Summary and Insights: Due to the decentralized feature, it is of importance to design a distributed access control mechanism with users' mobility and dynamic device management to be applicable to fog computing. Multiple devices owned by a user should access the real-time services without any configuration for new joining devices, and the revocation of old devices should be consistent for all fog nodes.

3) Lightweight Protocols Design: In fog computing, IoT devices communicate with fog nodes in one or two hops. This short-range communication makes real-time services feasible. Nevertheless, the delay of service response not only depends on the communication range and bandwidth, but also relies on the processing delay on fog nodes, which means that if the fog nodes have to perform complex computational operations to generate responses for users, the response delay is still large. In addition, due to the low computational capability of IoT devices, they have no capability to perform complicated operations or cost too much time on executing them. Therefore, the computational operations on both fog nodes and IoT devices should be lightweight.

To offer reliable services and protect data privacy, a variety of security protocols should be implemented on fog nodes, including identity authenticated key agreement, data encryption, digital signature, and spam detection [100]. If the deployed protocols are not efficient enough, they would cost a large number of computational resources and increase the response delay. The properties of lightweight cryptography have been discussed in ISO/IEC 29192, which is a new standardization project of lightweight cryptography [101]. Many lightweight cryptographic schemes [102], including blockciphers, hash functions, streamciphers and one-pass authentication ciphers, have been proposed to build efficient and secure end-to-end communications among resource-restricted devices, e.g., sensors, smart cards and RFID tags. The design of lightweight symmetric key cryptographic schemes is mature, but the feasibility of public-key solutions on RFIDs and sensors is an open problem because of the limitations on costs, areas and power, as they imply pretty low budget for the number of power, gates and bandwidth [102]. In addition, due to the unbalance of computational capabilities of fog nodes and IoT devices, the lightweight cryptographic schemes used on IoT devices may be ineffective to guarantee data security and user's privacy against curious fog nodes. Therefore, how to design lightweight cryptographic protocols with tolerable computational overhead for numerous kinds of devices to achieve secure real-time services is important in fog computing.

Summary and Insights: To provide real-time services to IoT devices with limited computational capability, designing lightweight security protocols is essential for secure fog-assisted IoT applications. Because of the inherent computational complexity of public-key cryptography, it is significantly difficult to realize the security properties that the public-key cryptography can provide based on the lightweight cryptography. Therefore, it needs big efforts to design lightweight protocols to support real-time services for fog-assisted IoT applications.

4) Intrusion Detection: In fog computing, malicious external and internal attackers can hack any entity at any time. The services may be slowly undermined by the successful attacks, if there is no proper intrusion detection mechanism implemented to discover malicious activity or policy violations on IoT devices and fog nodes. It is necessary to ensure that the whole architecture of fog computing is protected by defense mechanisms. Although the impact of these attacks is constrained in a local area in fog computing, the threats are serious as the local services may totally controlled by attackers. Fortunately, it is possible for a fog node to collaborate with the adjacent nodes and the fog nodes at higher levels in the network hierarchy to detect attacks that target large section of services [12]. This cooperation can improve the success rate of malicious attacks detection, while a reliable intrusion detection mechanism is needed for each fog node or IoT device.

Host-based Intrusion Detection Systems (HIDS) [103] are proposed to detect intrusion for the cloud by collecting and analyzing the data about system events, system calls and file systems. Through the observation on modification in host kernel, host file systems and program behavior, HIDS determines whether the cloud is hacked or not. Xu *et al.* [104] introduced an abstract model of intrusion detection and severity analysis to provide the overall security of the cloud with minimal response time and human intervention. Another kind of intrusion detection systems is Network-based Intrusion Detection System (NIDS), which detects malicious activity, such as DoS attacks, port scans and other attack behaviors, by analyzing network traffic. Using NIDS, Hamad and Hoby [105] proposed a method to offer intrusion detection as a service in cloud, and Houmansadr *et al.* [106] designed a cloud-based intrusion detection and response system for mobile phones. Some Distributed Intrusion Detection Systems (DIDS) [107], [108] were also designed to deploy multiple intrusion detection systems over a large-scale network. In DIDS, the intrusion detection systems can communicate with each other or a central server to monitor system security via system information and network flow collection and malicious activities detection. Recently, to detect the DoS attacks on intelligent traffic light control systems, Liu *et al.* [109] proposed two secure intelligent traffic light control schemes using fog computing based on the hardness of the computational Diffie–Hellman puzzle. In addition, to discover the intrusion on cyber-physical systems, such as smart grid and e-healthcare systems, many machine learning algorithms or data analysis approaches ware leveraged for network traffic analysis, for example, to detect false data injection attacks on smart meters in smart grid [110].

Summary and Insights: These intrusion detection systems may be applicable in fog computing to detect malicious activities, but they are not very efficient because of the heterogeneous, decentralized and distributed architecture of fog computing, such that the challenges on designing intrusion detection and prevention mechanisms are still numerous [12]. To prevent the various attacks, the first step is to understand the features of cyber attacks and the architecture of fog computing. In fog computing, each fog node can provide local services to the users in its area, a local defense mechanism is important for every type of services. The whole architecture cooperatively provides some real-time services to users, such that global intrusion detection mechanisms are also required in fog computing. Therefore, a balance between local and global defense mechanisms should be achieved. Furthermore, the autonomous detection for both fog nodes and IoT devices are critical in a decentralized architecture, which can reduce the overhead and improve the usability of intrusion detection mechanisms. Finally, to detect persistent threats, behavior features should be shared among cooperative fog nodes, how to share the information in decentralized architecture to achieve rapid intrusion detection and prevention is a challenging issue.

5) Resilience to Sybil Attacks: The fog computing is vulnerable to Sybil attacks, in which attackers are able to manipulate faked identities and abuse pseudonyms to compromise real-time services and IoT applications. In the presence of Sybil attackers, the normal users may be misled by the faked data and the IoT applications may generate incorrect results. A Sybil attacker may broadcast spam and advertisements, or disseminate malware and fishing websites to steal users' private information. For example, in fog-based crowdsensing, an attacker may report biased data to mislead the fog nodes to generate incorrect results or report repeated data to acquire unfair benefits. In different services, Sybil attackers have their motivations to take specific misbehavior to achieve their goals. Therefore, without an effective detection mechanism, the collective results would be easily manipulated by the attackers. Unfortunately, most Sybil attackers behave similarly to the normal users, how to detect the presence of Sybil attackers and thereby find out the Sybil attackers is extremely different, which makes Sybil defense of paramount importance.

To detect Sybil attackers, the basic information of normal users is needed for the detector to compare the difference between normal users and Sybil attackers, such as social graph, social community, behavior pattern and friend relationship. For example, Quercia and Hailes [111] proposed an mobile Sybil defense scheme to match mobile users' communities and label the users who are not in the trusted communities as Sybil attackers. In addition, some specific features, such as mobility pattern and channel characteristics, can be utilized to distinguish Sybil attackers from normal users. Park *et al.* [112] investigated the mobility of vehicles and obtained the fact that two vehicles rarely pass multiple roadside units always at the same time. Relying on this, the detector can investigate the correlation of the vehicles and roadside units in the spatial and temporal domains to identify Sybil attackers. Cryptography is a useful tool to restrict the misbehavior of Sybil attackers. Lin [113] designed a novel scheme from group signatures to resist local Sybil attackers and mitigate zero-day Sybil vulnerability in sparse VANET. In this scheme, if a user generates more than one signature on an event, these signatures would be linked to a specific user, who is labeled as a Sybil attacker. Liang *et al.* [114] investigated the trustworthy of users in service evaluation of mobile social networks and proposed a scheme to facilitate the review submission and limit the capabilities of Sybil attackers.

Summary and Insights: Although some off-the-shelf Sybil detection schemes could be adopted in fog computing, they cannot effectively defense Sybil attacks due to the decentralized architecture of fog computing. The data on single fog node may not be sufficient to find out whether a user is Sybil or not. Therefore, the cooperation of fog nodes is dramatically crucial in Sybil defense. In addition, many Sybil defense mechanisms tend to acquire users' behaviors and social relationship, the exposure of these information would directly cause users' privacy leakage. It is critical to address the privacy leakage during Sybil defense. Despite cryptographic encryption is able to hide the real information and prevent privacy leakage, it triggers a big obstacle on Sybil detection that data processing is unfeasible. Homomorphic encryption enables the processing on ciphertexts without disclosing the clear texts, but its computational and communication overhead and energy consumption are too heavy to be implemented on fog nodes. In summary, to guarantee the efficiency and effectiveness of Sybil defense while preserving privacy remains a critical and challenging issue.

6) Trust Management: Despite identity authentication and access control can get rid of the pretended or faked fog nodes and devices in IoT applications, it is still difficult to guarantee all the joining entities are fully trusted. Fog nodes may have different trust levels and establish various trust relationship with other fog nodes. As a result, a fog node may have no idea about how its partners are going to behave, when they cooperate to provide real-time services. In addition, all entities have a variety of collaborative peers at their disposal. Specifically, the users have various fog nodes available for service accessing in their vicinity, and the fog nodes also have multiple options to cooperate for providing real-time services. Nevertheless, some peers may not reach their expectations. For example, if one of the fog nodes cannot respond requests to users in time due to its low bandwidth, users' satisfaction would be decreased, although other fog nodes providing the same service with rapid response. Therefore, selecting fog nodes with high trustworthiness to cooperate with is quite important in the implementation of IoT applications and services.

A large number of trust management mechanisms [115], [116], [119], [120] have been proposed to analyze trust relationship under two trust models [115]: evidence-based trust model and monitoring-based trust model. In evidence-based trust model, any witness that proves trust relationship among users is exploited to build the trustworthiness, such as public key, address, identity, or any evidence that an user can generate for itself or other users [116]. For example, traditional cryptographic authentication is designed on the ownership of a public key or a credential. Monitoring-based trust model establishes trust among users through observing the behavior of past interactions between them. Wei *et al.* [117] proposed a trust management scheme based on the direct evidence to estimate the trustworthiness of an entity from the positive and negative feedback about this entity. When direct evidence is not available, indirect evidence such as third-party testimonies should also be complemented for estimating an entity's trustworthiness. Su *et al.* [118] provides end-to-end trustworthiness awareness based on the security attributes of all participating nodes, including security properties of systems and audit-based factors, such as past data usage actions and reputation calculations. Several works [119], [120] investigated on aggregating trust evidence from different sources and filtering out biased testimonies. Nitti *et al.* [120] defined a subjective trustworthiness model to enable each user to compute the trustworthiness of its friends on the basis of its own experience and on the opinion of friends in common with the potential servers. In addition, reputation is an important metric to evaluate the trust level of an entity and many reputation management mechanisms have been proposed to evaluate the users' trustworthiness in mobile ad hoc network (MANET) [121], vehicular ad hoc network (VANET) [122], delay tolerant networks [123] and mobile crowdsensing [124].

Summary and Insights: It is challenging to realize trust management for fog nodes due to the following reasons. First of all, because of the decentralized architecture of fog computing, it is hard to collect and manage the evidence and behavior information about fog nodes to evaluate their trustworthiness and build a trust evaluation model for all fog nodes in the network. Secondly, a trust management mechanism should be situation-specific, indicating that different trust metrics should be considered to reflect the unique properties of trust in different services and applications. Thirdly, trust management design should support scalability and consistency when changing network conditions, including dynamics of fog nodes, mobility of IoT devices, and traffic patterns. Overall, trust management for fog nodes is critical and how to achieve decentralized, situation-aware, scalable and consistent trust management mechanisms in fog computing requires extensive research efforts.

B. Challenges and Solutions on Transient Storage

Transient storage enables users to temporarily maintain their data generated by IoT devices on fog nodes. This local storage reduces the complexity of data management, but at the same time triggers new security and privacy issues, among which the leakage of data privacy is the most serious problem. Data integrity protection and secure data sharing are also important in transient storage, once the users lose physical control over their data. To achieve the data confidentiality, integrity and sharing in provisional distributed data storage, we introduce several security challenges and off-the-shelf solutions that should be integrated into fog computing.

1) Sensitive Data Identification and Protection: In IoT applications, the data collected by devices encapsulates various aspects of physical environment, including social events, pollution levels, traffic conditions and personal activities. Some data may be considered sensitive, e.g., personal activities, health status and personal information about individuals, while others are not, e.g., pollution levels and social events. Nevertheless, it is hard for the IoT devices to distinguish sensitive information from large volumes of data before uploading, since whether the data is sensitive or not is totally determined by the user, and each has his/her personal preference and choice. Furthermore, in different applications, the data from the same event has distinct security levels for different users. For example, a video camera deployed by the police captures a car accident at a crossroads and reports it to the local server. From the perspective of the police, there is no sensitive information in the video of the accident. However, this accident event is also captured by the cameras on nearby vehicles and they report this event to fog nodes, which can be used to trigger alarms for other drivers approaching this crossroads. This information may be sensitive for the reporting drivers since it leaks their current locations. Therefore, how to identify the sensitive information from large volumes of data is the first step for data protection in fog computing.

Identifying the IoT devices that produce data may not be sufficient to determine how sensitive the data is [125]. For instance, a location sensor may be deemed as generating sensitive data when it reports the movement of a particular person, but the data produced by this sensor may be less sensitive if it is deployed on a greyhound in transit. Moreover, the same sensor may produce the data with different sensitive levels, e.g., a facial recognition device in a hotel could provide the current location of a superstar, thus having more interests for the fans than the locations of common people.

Summary and Insights: One straightforward method to protect sensitive information is to encrypt all the data no matter whether it is sensitive or not. Unfortunately, this approach is quite blind and causes heavy overhead on IoT devices and communication channels. Thus, it is pretty necessary to identify the sensitive data before protecting it. Besides, sensitive data leakage happens everywhere. Almost every object, e.g., clothes, watches, shoes, jewelry and contact lenses, has sensors. Microphones and cameras embedded can record and monitor individuals' activities and report data to third parties. In summary, identifying and protecting sensitive data are equally essential to prevent information leakage for users.

2) Data Integrity Protection: When the data is maintained on fog nodes, the users lose their ultimate possession on the data generated by their IoT devices. As a result, the integrity and availability of their data are being put at risk due to the following reasons. First of all, although the infrastructure of fog computing is located at the network edge, it is still confronted with a broad range of external and internal threats towards data integrity as mentioned in Section IV. The users' data on fog nodes may be modified or deleted by attackers to destroy some evidence. For example, a fog node on roadside unit keeps the videos of a traffic accident reported by the nearby vehicles, which is an essential evidence for accident forensics. The offenders may attack the fog node to fabricate the evidence to wriggle out of punishment. Secondly, there are various motivations for fog nodes to behave unfaithfully towards users' data. For instance, the fog nodes may discard the data that is rarely accessed to reclaim the storage space for monetary reasons. Thirdly, the honest-but-curious fog nodes only do their "best" to keep the users' data and provide real-time services. Once the data is lost or corrupted, there is no mechanism to deal with this issue and the users have to accept the loss. In summary, if the problems of data integrity and correctness are not property addressed, it is hard for users to accept fog computing.

To offer the integrity guarantee for outsourced data, many provable data possession protocols [126]–[128] have been proposed to allow the data owners (i.e., users) or an trusted third party to efficiently verify the integrity of remote data in cloud storage without a local data copy. These protocols not only support various desirable properties, including public verification, data dynamic updating, privacy preservation and batch verification for multiple files, but also can be used in multiple clouds [128].

Summary and Insights: These protocols cannot be immediately utilized to check the data possession in fog computing due to the following reasons. First of all, fog nodes only provide transient storage for users' data, which may be further uploaded to the cloud data centers for permanent storage, or erased by fog nodes after it is expired. For example, a fog node only needs to store traffic collision reports for several hours and this information is of interest to the vehicles approaching the collision area. After the collision is solved and the traffic is normal, this information can be deleted on the fog node and a summary is required to report to the cloud. Therefore, the fog nodes have the rights to process, modify and delete the useless data, such that it is difficult to determine whether the fog nodes behave honestly towards the data or not. Secondly, due to users' mobility, multiple fog nodes may have the users' data. It is inefficient to check the data integrity on each fog node one by one. Thirdly, since different keys are used by IoT devices to generate the verifiable tags, the data generated by these devices cannot be aggregated during the integrity verification, even it belongs to a same user. As a result, the efficiency of data integrity verification is relatively low. Therefore, it is of significance to design secure and efficient provable data possession protocols to guarantee the data integrity and correctness in fog computing.

3) Secure Data Sharing: Fog computing provides transient storage for the data collected by IoT devices. To prevent the leakage of sensitive information, the data is encrypted before being uploaded to fog nodes, which puts huge obstacles on data sharing. Data sharing is a basic requirement that frequently happens in data storage systems. For example, the data collected by healthcare wearable devices should be shared with family doctors. Electricity consumption measurements of smart meters can be accessible for electricity system operators and utilities. Nevertheless, once the data is encrypted, it is impossible for other entities to read them, except the data owner. Therefore, how to share the data with other entities when necessary is a challenging issue for the data owner. One straightforward method is to give the decryption key to the entities with which the data is shared. This method directly exposes the decryption key. As a result, this entity can not only read the sharing data, but also access other data that the data owner does not want to share. Therefore, fine-grained data sharing becomes quite critical in encrypted data storage.

Currently, several cryptographic techniques [129], [133], [142] have been widely used to achieve fine-grained data sharing in cloud storage, e.g., proxy re-encryption, attribute-based encryption and key-aggregate encryption. To delegate decryption rights, Blaze *et al.* [129] proposed the notion of "atomic proxy cryptography", where a semi-trusted proxy can convert a ciphertext for Alice into the ciphertext that is decryptable for Bob without seeing the underlying plaintext using a proxy re-encryption key, and designed a bidirectional proxy encryption scheme based on ElGamal encryption. However, this scheme is only useful when Alice and Bob are mutually trusted, as the proxy re-encryption key can divert ciphertexts from Alice to Bob and vice versa. To address this issue, Ateniese *et al.* [130] proposed an unidirectional proxy re-encryption scheme based on bilinear maps and adopted it to achieve data sharing in encrypted file storage. Consequently, many proxy re-encryption schemes were proposed, including chosen-ciphertext secure proxy re-encryption [131] and conditional proxy re-encryption [132], to satisfy different requirements of data sharing.

Attribute-based encryption (ABE) [133] can be used to protect data security and achieve data sharing. In attribute-based encryption, a user' secret key and the ciphertext depend upon attributes (e.g., occupation, nationality, identity, status and membership). A user can decrypt the ciphertext only if the set of attributes the user has matches the attributes of the ciphertext. Currently, many attribute-based encryption schemes have been proposed [134], [135], which can be divided into two categories: ciphertext-policy attribute-based encryption (CP-ABE) [134] and key-policy attribute-based encryption (KP-ABE) [135]. The major difference between CP-ABE and KP-ABE is how to associate the access policy. In CP-ABE, a user's secret key is associated with a set of attributes and the ciphertext specifies an access policy. In CP-ABE, the access policy is encoded into the users' secret key, and the ciphertexts are generated from a set of attributes. Due to their distinctive properties, CP-ABE schemes are exploited to achieve data sharing in cloud storage [136] and KP-ABE schemes are used in e-healthcare systems to realize the sharing of personal health records [137]. To enable authentic and confidential data sharing among a group of fog nodes, Alrawais *et al.* [138] proposed an efficient key exchange protocol based on CP-ABE to secure fog communications. Alotaibi *et al.* [139] leveraged ABE and proxy re-encryption to achieve the fine-grained data access control and efficient access right revocation for data sharing in fog-assisted IoT applications. Jiang *et al.* [140] demonstrated that ABE suffers from key-delegation abuse and designed an enhanced CP-ABE to provide the protections against this key-delegation abuse issue. The proposed scheme was adopted in fog computing to achieve the fine-grained data sharing. Yu *et al.* [141] designed a leakage-resilient functional encryption scheme applicable to fog computing to achieve securely share data. Functional encryption, a cryptographic primitive developed from ABE, supports fine-grained access control policy with the match between the access control policy and attributes of users.

The concept of key-aggregate encryption was proposed by Chu *et al.* [142] to achieve scalable data sharing in cloud storage. In key-aggregate encryption, the data owner delegates the decryption capability of flexible choices of ciphertext set by releasing a constant-size key aggregated from the corresponding set of secret keys, but the other encrypted data outside the set remains confidential. Derived from this scheme, key-aggregate searchable encryption [143] and time-bound key-aggregate encryption [144] have been designed to support keyword search and bound decryption capability, respectively.

Summary and Insights: Although the aforementioned schemes can realize secure data sharing, the efficiency is still a bottleneck when they are implemented on IoT devices, as most of these schemes are constructed from the time-consuming bilinear pairing. An efficient approach is to use key management to achieve data sharing in fog computing, resulting in the challenge on the management of decryption keys. Therefore, how to achieve fine-grained data sharing with data privacy in fog computing is not an easy work. If there is no efficient encrypted data sharing scheme to realize fine-grained data access in fog computing, the utilization of data resource cannot be maximized and many real-time applications are blocked due to the lack of reliable data.

C. Challenges and Solutions on Data Dissemination

Fog nodes build two-way communications between the cloud and IoT devices on behalf of intermediates in networks. It is inevitable to disclose sensitive data during transmission due to the security threats in fog computing. End-to-end secrecy transmission can prevent data from being learnt by malicious attackers and curious entities. Nevertheless, it sacrifices many desirable properties, such as aggregation, search and sharing, which can significantly improve communication efficiency and enhance data usage. Therefore, how to achieve confidential data aggregation, search, sharing and forwarding is a promising issue in fog computing.

1) Privacy-Preserving Data Aggregation: In some IoT applications, e.g., mobile crowdsensing and smart metering, each device collects data from physical world and encrypts it to preserve data privacy before forwarding to fog nodes. The fog nodes transiently store the received data or deliver it to the cloud. During these processes, secure data aggregation is critical to prevent data leakage and reduce communication overhead. For example, in smart grid, the fog nodes can aggregate the ciphertexts of individual electricity consumption measurements reported by smart meters and submit the aggregated ciphertext to the operation center for load monitoring. In this way, the communication overhead is significantly reduced compared with the transmission of all individual measurements separately. Therefore, how to achieve secure data aggregation based on different requirements in IoT applications is pretty important.

Currently, homomorphic encryption schemes, such as Paillier encryption [145] and BGN encryption [146], have been widely used in smart grid to achieve data aggregation. These encryption schemes support additive homomorphism, indicating that the fog nodes can aggregate the ciphertexts of individual power consumptions reported by smart meters in a residential area, and the operation center recovers the sum of power consumption in that area from the aggregated ciphertext. In this way, the individual consumption of a household is protected from external attackers and the communication efficiency between fog nodes and the cloud is improved. Homomorphic encryption schemes are also used to achieve users' privacy preservation and support additive homomorphic operation in mobile social networks [147], [148] and to support secure smart metering with privacy preservation in smart grid [149]. Lu *et al.* [150] designed an efficient privacy-preserving data aggregation scheme for fog-assisted IoT applications based on homomorphic Paillier encryption, Chinese Remainder Theorem, and one-way hash chain techniques to not only aggregate hybrid IoT devices' data into one, but also early filter injected false data at the fog nodes. Wang *et al.* [151] proposed an anonymous and secure data aggregation scheme from Castagnos–Laguillaumie cryptosystem in fog-based public cloud computing. Furthermore, by integrating multiplicative homomorphism, data statistics and computation can be executed on the ciphertexts in a blind way, indicating that a third party (i.e., the cloud) can perform data statistics and computations without learning any information about the values in the ciphertexts. Due to this appealing property, full homomorphic encryption schemes [152], [153] were proposed to achieve encrypted data processing. For example, Zhuo *et al.* [154] utilized full homomorphic encryption to achieve privacy-preserving data statistics on crowdsourcing data. However, in reality, it is still impossible to adopt fully homomorphic encryption to achieve data protection and data aggregation in IoT because of its heavy computational overhead. Therefore, how to achieve privacy-preserving data aggregation without full homomorphic encryption becomes appealing. Zhou *et al.* [155] utilized one-way trapdoor permutation to realize time series data aggregation for wireless wearable communications. Besides, key distribution and key agreement mechanisms [156], [157] are employed in smart grid, wireless body area network and vehicular ad hoc network to enable multiple users to share a session key, and thereby achieving privacy-preserving data aggregation. In summary, efficient and secure data aggregation schemes should be designed to satisfy the different requirements of data aggregation in various IoT applications.

Apart from encrypted data aggregation, users' signatures can be aggregated as well to further decrease computational and communication burden. The aggregation of signatures works as follows: Given n signatures on n distinct messages from the same user, it is possible to aggregate all these signatures into a single short signature [158]. Due to this appealing property, a large number of aggregate signature schemes [158], [159] were proposed to achieve aggregation of a user's multiple signatures. Nevertheless, aggregate signatures only can aggregate the signatures generated by the same user. Consequently, multi-signature [160], sequential aggregate signature [161] and key-homomorphic signature [162] were introduced to aggregate n signatures on a same message from n distinct users. For example, Ni *et al.* [163] utilized key-homomorphic signature to aggregate the signatures generated by multiple users who report reduplicate data in spatial crowdsensing. Unfortunately, there is no efficient multikey-homomorphic signature that can aggregate n signatures on n distinct messages generated by n users [162], unless some information is pre-shared among users [164]. Therefore, how to design universal aggregate signatures and adopt them to IoT applications deserves to pay more attentions and efforts.

Summary and Insights: In fog computing, fog nodes enable to aggregate transmitting messages to reduce communication overhead on behalf of intermediates without disclosing any information in large-scale IoT applications. Nevertheless, different IoT applications have distinctive requirements on data aggregation to support a variety of properties. To define the properties of data aggregation (e.g., addition, multiplication, linear aggregation, and polynomial aggregation) and design suitable privacy-preserving data aggregation schemes are essential for particular IoT applications. In terms of signature aggregation, it is quite significant to construct the multikey-homomorphic signature to aggregate n signatures on n distinct messages generated by n users.

2) Secure Data Search: To prevent data leakage in transient storage, the IoT devices are required to encrypt the collected data before uploading. Once the data is encrypted, it is difficult for the data owner (i.e., user) or other entities to search on the ciphertexts and retrieve their interested part. However, data search and retrieving is the fundamental demand in data storage. For example, in vehicular crowdsensing, the vehicles on road report traffic information to fog nodes, and the cloud searches on the fog nodes based on the crowdsensing tasks released by customers to retrieve the required data to generate crowdsensing results for the customers. In addition, since fog nodes only provide transient storage for users, the cloud is required to search and retrieve the data summaries for persistent storage. To realize encrypted data search, secure index should be built by users when they upload data to fog nodes. Therefore, how to achieve privacy-preserving matching between searching request and secure index is the primary goal of encrypted data search in fog computing.

The notion of searchable encryption was proposed by Song *et al.* [165] to achieve encrypted data search without exposing any information. In general, searchable encryption consists of two phases: one is data and index encryption, and the other is index search and data retrieving. In the first phase, a data owner generates data index based on keywords, utilizes a symmetric encryption scheme to encrypt the data and employs searchable encryption algorithm to encrypt the data index. Both data ciphertext and index ciphertext are uploaded to a fog node. In the second phase, a user generates a trapdoor from keywords that he/she wants to search, and submits it to the fog node. Then, the fog node detects whether the trapdoor can match the index ciphertexts. If a match exists, the fog node returns the corresponding data ciphertext, and the user decrypts to obtain data. Here, the user can be the data owner or an entity that the data owner is willing to share with. If a symmetric key is used to encrypt data index and generate the trapdoor, these schemes are called symmetric searchable encryption schemes; otherwise, they are asymmetric searchable encryption schemes, in which the public key is used to encrypt data index, and the secret key is utilized to compute the trapdoor.

In symmetric searchable encryption, some works extended single keyword search and Boolean query to support various search patterns, including fuzzy keyword search [166] and ranked keyword search [167]. Others [168], [169] defined different privacy levels of searchable encryption, namely, data privacy, trapdoor privacy, index privacy and access pattern privacy, and designed schemes to achieve high privacy guarantees. For example, Rizomiliotis and Gritzalis [168] utilized oblivious RAM to design two efficient dynamic symmetric searchable encryption schemes with forward privacy. Naveed *et al.* [169] introduced a new primitive called blind storage and proposed a dynamic symmetric searchable scheme to allow the user to store a set of files on a cloud server in such a way that the cloud server cannot know the number of stored files, or the lengths of the individual files. Yang *et al.* [170]

introduced the fog-enhanced location-based services to restrict the access of users out of the coverage areas of fog nodes, and designed a fine-grained and privacy-preserving query scheme based on k-nearest neighbors algorithm to achieve the location matching without disclosing the location information. Asymmetric searchable encryption is firstly introduced by Boneh *et al.* [171] to achieve encrypted mails search on untrusted mailing servers. Consequently, many expansions, such as keyword search with a designated tester [172], public key encryption with conjunctive keywords search [173] have been proposed to satisfy different properties in real applications. In addition, predicate encryption [174] and hidden vector encryption [175] are also effective encrypted data search techniques, which can be used to realize data search based on multiple keywords. However, in asymmetric searchable encryption, the privacy of data, index and trapdoor can be preserved, but the access pattern is exposed to the cloud data center.

Summary and Insights: Although many symmetric searchable encryption schemes [176], [177] can preserve index privacy, trapdoor privacy and access pattern privacy simultaneously, privacy leakage in encrypted data search still exists in fog computing. The fact that whether a fog node returns the matched data or not will disclose users' privacy. Specifically, the users submit the collected data to the fog nodes. If the data of a specific user is obtained and the fog nodes return the matched data, it is predictable that the user ever appeared in the coverage areas of the fog nodes. For example, a vehicle drives through a sequence of fog nodes, and uploads the collected data. If an attacker searches the data collected by that vehicle, and the driving-through fog nodes would return the data collected by that vehicle, other fog nodes cannot. As a result, the vehicle's trajectory is disclosed. Therefore, preserving data privacy and users' privacy in encrypted data search is still a critical and challenging task.

3) Secure Content Distribution: Fog nodes can provide local content distribution services to the users located in their coverage areas. Some content is customized under the requests of users, e.g., navigation information and subscribed newspaper, others may be broadcasted automatically, e.g., advertisements and parking information. The distributed content may contain plenty of personal information, such as preference, incentive, current location and political inclination. The subscribed newspaper and magazines may disclose preferences and political inclination of the subscribers. To prevent information leakage, it is of importance to define who can receive the broadcasting content and what kinds of content should be obtained by a specific user. For instance, in pay-TV systems, the recommendation of TV content contains sensitive information about the receiver, including preferences and schedule, and the subscribed content should be encrypted during transmission to prevent nonsubscribers from watching them. Therefore, fine-grained secure content distribution with users' privacy preservation remains a challenging issue.

The primary task is to guarantee the accuracy of content distribution, which means that the users can discover their interested content in high probability. If the users always receive boring information from fog nodes, they would be disappointed in the content distribution services and refuse to subscribe them. To customize their services, one approach is to allow the users to select the interested content and regulate their personal lists in advance, but this approach discloses the users' preference to fog nodes. Secure service discovery mechanisms [178], [179] have been proposed to ensure that the services are only discoverable by an authorized set of users. This technique is adopted to discover content distribution services without disclosing any information about their preferences. In addition, the service providers (i.e., fog nodes) do not keep any knowledge about the users, such that the service would not be suspended, even a user enters the coverage area of a new fog node. Therefore, secure service discovery is an important technique to assist users to acquire appropriate services with privacy preservation.

Another issue deserving to pay attention is how to send the content to the users confidentially. Although a session key can be negotiated between fog nodes and users to guarantee end-to-end secure transmission, this method is inefficient if the number of receivers is large. Broadcast encryption [180] is a promising method that delivers the encrypted content over a broadcast channel, in such a way that only the qualified users can recover the content. By using broadcast encryption, multiple qualified users can receive the same content in an efficient way. Moreover, several key management mechanisms [181], [182] have been proposed to realize key update or resist collusion attacks in broadcast encryption. Nevertheless, traditional broadcast encryption schemes expose the identities of the receivers. To address this problem, anonymous broadcast encryption [183] was presented to achieve receivers' anonymity and content broadcast efficiently. In short, broadcast encryption is practical to be implemented in content distribution applications.

Summary and Insights: The environment of content distribution in fog computing is dynamic, indicating that the receivers of the broadcasting content change frequently. This dynamic makes the key management of secure service discovery and broadcast encryption more challenging than ever. When a user discovers and joins his/her appropriate service, the secret key used to distribute content before should be updated to prevent the joining user from learning the previous content. If a user leaves this group, his secret key should be revoked. Therefore, to investigate the approaches to achieve secure service discovery and anonymous broadcast encryption simultaneously in dynamic environment is significance for the development of local content distribution in fog computing.

4) Privacy-Preserving Packet Forwarding: In fog computing, the fog nodes act as intermediate nodes to forward packets received from IoT devices or other fog nodes to the upper levels. To prevent these nodes from learning the personal information about the users during packet forwarding, they can utilize encryption and anonymity techniques to encrypt the data and hide their identities, respectively. However, just keeping the data secret and anonymous is not enough in some cases, since it could be disclosed by other means. For example, in e-healthcare systems, a patient often sends his/her personal health information collected by wearable devices to a specific physician, and then the physician can diagnose diseases for the patient. If an observer or fog nodes are curious about the patient's disease, they can observe the receiver of the personal health information and guess the disease based on the medical treatment domain of the physician. Therefore, through observing and analyzing packet forwarding, the curious entities can also violate users' privacy.

To cut off the privacy exposure during packet forwarding, Lin *et al.* [184] defined two notions of content-oriented privacy and contextual privacy, and proposed a strong privacy-preserving scheme against global eavesdropping to protect both content-oriented privacy and contextual privacy against strong global attackers, who can observe the receiver of the patient's person health information in e-healthcare systems. In VANET, to improve delivery ratio, Lu *et al.* [185] proposed a novel social-based packet forwarding protocol to enable roadside units at high social intersections to assist in packet forwarding between vehicles. However, Lin *et al.* [186] demonstrated that receivers' location privacy is exposed to intermediate nodes (i.e., roadside units) in data forwarding, and presented a social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in VANET.

Summary and Insights: In fog computing, the fog nodes have the global observation on the forwarding packets and learn all the data flow about users in their coverage areas. According to the sender and receiver's locations and other available information, the fog nodes could be aware of the relationship between the sender and receiver and guess personal information about them, such as occupations, workplace, address and intimacy. Furthermore, the curious fog nodes may re-identified the senders based on their mobility patterns, even if the senders have anonymized their packets. Therefore, to prevent the privacy leakage against intermediate fog nodes during packet forwarding is crucial in fog computing.

D. Challenges and Solutions on Decentralized Computation

Fog nodes have certain computational capabilities to do data processing and analysis. However, because of the security threats in fog computing, fog nodes may be compromised by attackers, such that not only the processing data on fog nodes has risks to be exposed to attackers, but also the computation results can be controlled. Thus, how to prevent the disclosure of data privacy and ensure the correctness of computation results becomes a huge concern for the users when they outsource their computation tasks to fog nodes. Therefore, we discuss the challenges of verifiable computation, secure aided computation and secure big data analysis, and introduce several secure and verifiable computation techniques to resolve these challenges.

1) Verifiable Computation: Fog computing has computing resources to undertake computation tasks offloaded by the cloud in a distributed way. Specifically, the user submits the tasks to the local fog nodes and obtain the computation result with low latency. However, both the cloud and fog nodes cannot be fully trusted, whether the returned result is correct or not becomes a huge concern for the user, as the user cannot compute the result by himself because of the low computational capability of his devices. In addition, the fog nodes provide distributed computation services to users on behalf of the cloud at the network edge. The cloud also concerns whether the fog nodes provide correct and trusted computation results to the users. The correctness verification on computation results is significantly important for the users and the cloud. If there is no mechanism to check the correctness of the returned results, the cloud may be unwilling to offload computation tasks to fog nodes and the users are not access the services offered by fog nodes, resulting in the failure of computation offloading in fog computing.

Gennaro *et al.* [187] introduced the notion of verifiable computation and designed a non-interactive verifiable computation scheme based on garble circuit [188]. Chung *et al.* [189] utilized full homomorphic encryption schemes to construct a non-interactive verifiable computation scheme with a small size of the public key. Consequently, Parno *et al.* [190] designed a publicly verifiable computation scheme based on CP-ABE, and Papamanthou *et al.* [191] proposed a new model for verifying dynamic computations in cloud settings. To support multiple users, Choi *et al.* [192] proposed a multi-user non-interactive verifiable computation scheme using proxy oblivious transfer schemes. Gordon *et al.* [193] exploited attribute-based encryption, full homomorphic encryption and Yao's garble circuit to construct a multi-user verifiable computation scheme. Recently, Elkhiyaoui *et al.* [194] proposed an efficient publicly verifiable delegation of computation, and Zhuo *et al.* [154] adopted verifiable computation technique to design a privacy-preserving verifiable data aggregation scheme for mobile crowdsourcing.

Summary and Insights: Most of the aforementioned schemes are general constructions on verifiable computation, which introduce theoretical approaches to achieve privately or publicly verifiable computation. Nevertheless, in fog computing, the fog nodes cooperatively perform the computation tasks for users in distributed way. The errors made by one fog node can spread to other fog nodes and lead to an incorrect final result. Therefore, all the intermediate results and the final result should be verified to guarantee the correctness of results and trace the misbehaving fog nodes outputting fault results. How to design practical publicly verifiable computation schemes suitable for IoT applications in fog computing still deserves to focus on.

2) Secure Aided Computation: With the computing resources, fog nodes can assist the IoT devices to perform complex computational operations that they cannot executed by themselves. To achieve aided computation, one straightforward method for the IoT devices is to send their collected data and other needed information to the fog nodes and thereby allow fog nodes to execute the computation on behalf of devices. However, this method inevitably exposes all sensitive information to fog nodes, which may have been compromised by attackers. Moreover, the users' secret keys are required in some operations, e.g., signature generation and message decryption. If the fog nodes possess users' secret keys, they can pretend the users to do everything they want. Therefore, there should be sufficient mechanisms to permit fog nodes to execute operations for IoT devices without invading users' privacy. Otherwise, the users prefer to purchase new powerful devices to perform computations by themselves, rather than seeking assists from fog nodes.

The concept of server-aided computation is proposed by Matsumoto *et al.* [195], whose goal is to speed up secret computation using insecure auxiliary devices. Several server-aided computation protocols [196], [197] have been proposed to deal with secure exponentiation with an aided server. Specifically, Kawamura and Shimbo [196] proposed fast server-aided secret computation protocols for modular exponentiation, Cavallo *et al.* [197] securely and efficiently delegated the computation of group exponentiation to a single server. Consequently, Girault and Lefranc [198] introduced the notion of server-aided verification to speed up the verification for an authentication/signature scheme by delegating a part of computation to a powerful but untrusted server, and designed a generic method for achieving server-aided verification based on bilinear maps. Wu *et al.* [199] defined the security model of server-aided verification signature and constructed an efficient server-aided verification signature scheme based on BLS signature [200]. In addition, other server-aided schemes, including server-aided encryption [201], server-aided function evaluation [202] and server-aided key exchange [203], were proposed to speed up the computation for users. Wang *et al.* [204] employed fog nodes as trusted third parties to generate dummy positions to protect the location of users and store partial important data to reduce threats on data leakage.

Summary and Insights: In fog computing, fog nodes cooperatively assist IoT devices or users to execute computation tasks. These tasks are much more complex than exponentiation, encryption/decryption and signature verification, and multiple fog nodes are more powerful and smarter than a single server. For example, Ni *et al.* [163] utilized fog nodes to perform deduplication on the collected data from multiple IoT devices and aggregate the key-homomorphic signatures generated by these devices. Therefore, how to utilize fog nodes to aid IoT devices to perform aided computation tasks to satisfy different features and goals in IoT applications is worthy of attention and efforts in fog computing.

3) Secure Big Data Analysis: Nowadays, a large amount of users' data is recorded in various forms, e.g., videos, photos and e-documents. The analysis on these data using data mining and machine learning algorithms pose a great challenge on individual privacy in big data era. Although de-identification is widely used to prevent attackers from linking the processing data with individual's identity, the anonymous data is still easily hacked in terms of privacy. It is extraordinarily challenging on privacy preservation during big data analysis. First of all, privacy is a subjective concept, it is hard to distinguish sensitive information, which may be integrated with non-sensitive data. Secondly, the data mining and machine learning algorithms can discover potential correlation between two data sets, which brings desperate threats to privacy. Thirdly, due to the rapid development of data mining and machine learning technologies, an effective privacy preservation technique may be easily breached [205]. Therefore, it is of significant importance to understand the privacy issues in dataset and design reliable mechanisms to prevent privacy leakage in big data analysis.

To address these issues, fully homomorphic encryption [152], [153] and differential privacy [206] are widely used to realize large-scale data analysis and preserve users' privacy simultaneously. Fully homomorphic encryption allows the computation to be executed on ciphertexts and thereby generating an encrypted result, which is the ciphertext of the result of operations performed on the plaintexts. By utilizing homomorphic encryption, Li *et al.* [207] designed cloud-aided privacy-preserving frequent item set mining scheme for vertically partitioned databases, which can be exploited to achieve privacy-preserving association rule mining. Zhang *et al.* [208] employed fully homomorphic encryption scheme to achieve human-to-human infection analysis based on social data and health data collected by mobile devices. Liu *et al.* [209] designed a hybrid privacy-preserving clinical decision support system in fog and cloud computing environment, from Paillier encryption with threshold decryption, neural network and other building block to securely monitor patients' health condition in real-time. Differential privacy [206] is a privacy-preserving technique that utilizes random noise to maximize the accuracy of queries from statistical databases, while minimizing the chances of identifying individual records. Abadi *et al.* [210] proposed a privacy-preserving deep learning algorithm within the framework of differential privacy. Zhang and Zhu [211] studied regularized empirical risk minimization machine learning problems, and proposed dual variable perturbation and primal variable perturbation to realize dynamic differential privacy. Further, local differential privacy [212], [213] technique is essential to support distributed data collection and centralized data analysis without disclosing any information about the collected data. Erlingsson *et al.* [212] used local differential privacy to design a randomized aggregate privacy-preserving ordinal response for crowdsourcing statistics with strong privacy guarantees. Qin *et al.* [213] presented a systematic study of heavy hitter mining under local differential privacy, and proposed a two-phase mechanism with local differential privacy for obtaining accurate heavy hitters, which is more efficient budget-wise than the approach obtaining the heavy hitters directly from the whole dataset.

Summary and Insights: Fully homomorphic encryption has its inherit drawback on computational overhead, which is intolerable for both resource-restricted IoT devices in data encryption and fog nodes in function evaluation, especially on large volumes of data sets. In terms of differential privacy, the existing schemes are constructed on centralized data storage. Nevertheless, the architecture of fog computing is decentralized and the data may come from various services and different users. How to design decentralized big data analysis with differential privacy in fog computing is pretty critical and challenging.

In summary, a variety of state-of-art schemes have been proposed to solve the security and privacy issues in cloud-based IoT applications, including smart grid, e-healthcare systems, VANET and mobile crowdsensing. These schemes cannot be adopted to resolve the security and privacy challenges in fog-assisted IoT applications immediately, due to the

TABLE IV
SECURITY CHALLENGES AND SOLUTIONS IN FOG COMPUTING

Roles	Security Challenges	Security Solutions
Real-Time Services	Identity Authentication	Identity Authentication [83]–[88], Cooperative Authentication [89], [90], Anonymous Authentication [91]–[93].
	Access Control	Role-based Access Control Policy [95], Attribute-based Access Control Policy [96], Device and Key Management [97]–[99].
	Lightweight Protocols Design	Lightweight Cryptographic Schemes [100], [101], Lightweight Elliptic Curve Cryptosystem [102].
	Intrusion Detection	Host-based Intrusion Detection Systems [103], [104], Network-based Intrusion Detection System [105], [106], Distributed Intrusion Detection Systems [107]–[110].
	Resilience to Sybil Attacks	Mobile Sybil Defense [111], [112], Cryptography-based Sybil Defense [113], [114].
	Trust Management	Evidence-based Trust Model [115], Monitoring-based Trust Model [117]–[120], Reputation Management [121]–[124].
Transient Storage	Sensitive Data Identification and Protection	Symmetric Encryption [217], Asymmetric Encryption [218].
	Data Integrity Protection	Provable Data Possession [126]–[128].
	Secure Data Sharing	Proxy Re-encryption [129]–[132], Attribute-based Encryption [133]–[141], Key-Aggregate Encryption [142]–[144].
Data Dissemination	Privacy-preserving Data Aggregation	Homomorphic Encryption [145]–[154], One-way Trapdoor Permutation [155], Key Distribution and Key Agreement [156], [157], Homomorphic Signature [158]–[164].
	Secure Data Search	Symmetric Searchable Encryption[165]–[170],[176], [177], Asymmetric Searchable Encryption [171]–[175].
	Secure Content Distribution	Secure Service Discovery [178], [179], Broadcast Encryption [180], Key Management Mechanism [181], [182], Anonymous Broadcast Encryption [183].
	Privacy-preserving Packet Forwarding	Privacy-preserving Packet Forwarding [184]–[186].
Decentralized Computation	Verifiable Computation	Privately Verifiable Computation [187], [189], [192], [193], Publicly Verifiable Computation [154], [190], [191], [194]
	Secure Aided Computation	Server-aided Exponentiation [196], [197], Server-aided Verification [198], [199], [204], Server-aided Encryption [201], Server-aided Function Evaluation [202], Server-aided Key Exchange [203].
	Secure Big Data Analysis	Fully Homomorphic Encryption [152], [153], [207]–[209], Differential Privacy [206], [210]–[213].

unique features of fog computing. As fundamental approaches, they point out the directions we could focus on for addressing these issues. Table IV shows the security and privacy challenges in fog computing and the possible solutions in fog-assisted IoT applications. Fig. 5 illustrates our insights on the reasons why the existing solutions cannot directly leveraged to solve the security and privacy challenges in fog computing. We analyze the unique features of fog computing and demonstrate the obstacles on solving the security and privacy problems. The potential challenges on security and privacy in fog computing deserve further investigation and efforts.

VI. ISSUES, CHALLENGES AND FUTURE RESEARCH DIRECTIONS

In this section, we discuss several open problems in fog computing, as shown in Fig. 5, and expect to trigger more research efforts on these issues.

A. Location Privacy Preservation

The location awareness and geographic distribution of fog computing contribute to local real-time services, local data management and local content distribution. For example, Huo *et al.* [78] proposed a location difference-based proximity detection protocol to allow a user to discover the friends in the coverage area of the same fog node. Yang *et al.* [79] proposed the location verification of fog nodes to prevent an attacker from maliciously claiming a forged position for service access. Therefore, fog computing makes many location-based services and functions more attractive than ever. Unfortunately, due to the feature of localization, users' locations are obliviously exposed in fog computing. For example, a wearable device uploads its collected data to a fog node and the fog node further deliveries the data or its summary to the cloud. During these processes, both the fog node and the cloud can learn the location of the wearable device carried by the user. Even if a reliable anonymity technique is utilized to effectively prevent the fog node and cloud from identifying the user, the location is still exposed via positioning techniques if the devices connect multiple fog nodes, such as three-point positioning. With the leakage of precise location information, the user has a high risk to be identified according to the fact that an anonymous user can be re-identified in the probability of 90% with four spatial-temporal points [205]. Further, the cloud is

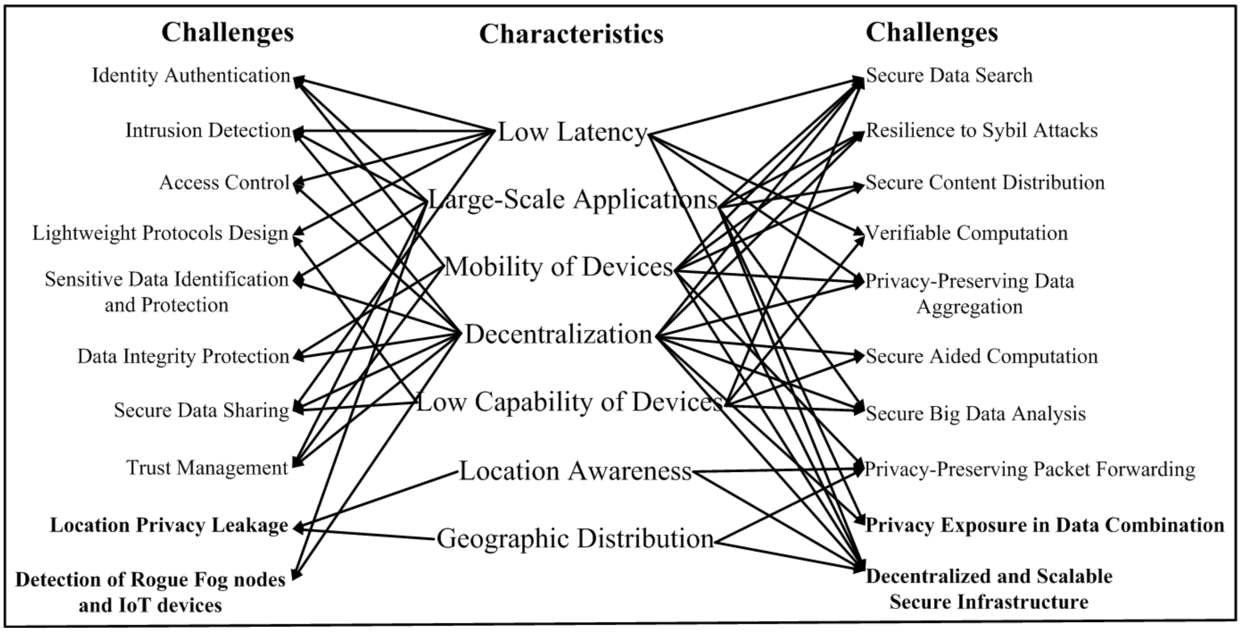

Challenges
Characteristics
Challenges
Identity Authentication
Intrusion Detection
Access Control
Lightweight Protocols Design
Sensitive Data Identification and Protection
Data Integrity Protection
Secure Data Sharing
Trust Management
Location Privacy Leakage
Detection of Rogue Fog nodes and IoT devices
Low Latency
Large-Scale Applications
Mobility of Devices
Decentralization
Low Capability of Devices
Location Awareness
Geographic Distribution
Secure Data Search
Resilience to Sybil Attacks
Secure Content Distribution
Verifiable Computation
Privacy-Preserving Data Aggregation
Secure Aided Computation
Secure Big Data Analysis
Privacy-Preserving Packet Forwarding
Privacy Exposure in Data Combination
Decentralized and Scalable Secure Infrastructure

Fig. 5. Correlation between Features and Security Issues in Fog Computing.

able to learn the rough area of the user based on the location of the fog node, and thereby learning the user's activity region. Therefore, due to the involvement of fog nodes, the global location privacy of an anonymous user is reduced to local location privacy. As a result, it is of difficulty to protect users' location privacy, as the curious cloud can always learn the knowledge about the rough regions of users according to the geographic locations of fog nodes.

B. Detection of Rogue Fog Nodes and IoT Devices

As fog computing architecture is vulnerable to a large variety of cyber attacks as mentioned in Section IV, both fog nodes and IoT devices have huge risks to be compromised. The corrupted fog nodes and IoT devices may pretend to be legitimate and coaxes users to connect to them. Roman *et al.* [12] demonstrated that fog nodes are vulnerable to several DoS attacks, such as distributed DoS attacks and wireless jamming, as fog nodes have less resource than the cloud. Stojmenovic *et al.* [214] showed the feasibility of man-in-the-middle attacks in fog computing under the condition that the gateway has been compromised or replaced by a fake one. A user's secret key can be extracted from the digital certificate if the device is compromised or hacked. Moreover, even if the fog nodes and IoT devices are not compromised, they may turn to rogue nodes due to their personal incentives. For example, a rogue fog node may be deployed to broadcast rumors and fraud to the driving-through vehicles. Some rogue IoT devices may collude to manipulate the results of mobile crowdsensing. The fake, compromised or rogue fog nodes and IoT devices would be big threats to the security and privacy of users' data. Unfortunately, it is difficult to detect them in fog computing due to the following reasons. Firstly, trust models are various in different applications, which require distinct trust management schemes to deal with them separately. Secondly, the dynamic and distributed environment makes it hard to maintain a blacklist of rogue fog nodes and IoT devices. One important method is to build reliable and strong anomaly intrusion detection systems for each fog node and IoT device. It is possible to reuse various intrusion detection and protection systems developed for cloud computing. However, these systems only can detect external attacks with a certain probability. It is hard to discover the rogue or corrupted fog nodes and IoT devices. Therefore, the study on the method to detect the rogue and corrupted fog nodes and IoT devices in fog computing is worthy to focus on.

C. Privacy Exposure in Data Combination

In IoT applications, the devices act as data producers to generate and process data of various levels of sensitivity. Some may be inherently sensitive, such as the data generated by a person's heart-rate sensor, but the others might be benign. However, even the collected data seems to be non-sensitive, the application of data in combination can trigger serious security and privacy concerns [215]. This problem is exacerbated by the use of fog computing for IoT, as one of the motivations for fog computing is to enable the collaboration among fog nodes explicitly that can aggregate and process the data cross a large number of IoT devices. For example, a patient buys some pills at a pharmacy and pay with a credit card. The sensitive information of the patient the pharmacy has is limited if it does not have the personal information of the patient, even it remembers the face and the credit card number of the patient. Nevertheless, if the customers' information in local credit card center combines with the purchasing information of patients in the pharmacy, the pharmacy is able to link the health status with the patient's identity. As a result, the sensitive information about the patient is disclosed to the pharmacy, including identity, health status, living address and telephone number.

This simple example also indicates the importance of identity information protection, as benign information may become sensitive, combining with the identity information. The fusion of raw data from different sources can enhance the potential values of these data. For instance, the spread of infectious diseases, e.g., flu, Ebola and SARS, is difficult to control as they can rapidly infect from human to human in a short period through close contact. Susceptible patient isolation is an effective way to prevent a patient from spreading virus to others, but how to find the people who have contacts with a person carrying virus after he/she is isolated is a big challenge. Social network data can play a vital role in the discovery of susceptible patients. Some social applications on mobile devices, e.g., Wechat, can find people in the physical proximity, record social interactions and detect if some people cough or sneeze via speech recognition. Therefore, the combination of social data and health data is a novel approach to enhance infection analysis. However, health data and social data are privacy-sensitive and they are not in the same trusted domain. To perform infection analysis, health data and social data should be combined and shared with others, which inevitably exposes sensitive information. Nevertheless, people are not willing to excessively reveal this sensitive information to an untrusted party. This issue raises to the tradeoff between the functional benefits of combining data, and the privacy perspective. To balance this tradeoff, several techniques should be used to prevent the privacy risks in data combination, such as differential privacy and fully homomorphic encryption. However, these techniques are still in their infancy stage and more efforts should be paid. In addition, during the data analysis, it is hard to anticipate all possible information leakage that arises from data combination. Therefore, there is a clear need to define some levels of privacy protection in data combination and design efficient and effective privacy-preserving methods to protect users' privacy in fog computing.

D. Decentralized and Scalable Secure Infrastructure

Fog computing is a decentralized, scalable and dynamic paradigm, in which both fog nodes and IoT devices can arbitrarily join and leave the architecture. Because of the lack of centralized server, it is challenging to build a secure infrastructure in such a distributed framework. First of all, it is impossible to find a trusted leader to determine the trustworthiness of fog nodes. If there is no party to regulate network operations, network stability and service reliability are hard to be achieved because of the corruption of fog nodes. Secondly, traditional security mechanisms are quite inefficient in the decentralized and dynamic framework of fog computing. For example, each fog node has to authenticate the users' identities and delegate access right to each user before allowing them to access services. If each fog node stores a copy of authentication credentials, the storage efficiency is low, and if the credentials are kept on one powerful fog node, the communication overhead is heavy. It is of difficulty to find an efficient approach to achieve rapid authentication and delegation in fog computing. Thirdly, distributed computation can be performed on the multiple fog nodes, but it is hard to guarantee the correctness of computation results, since not all fog nodes is fully trusted. Even if these problems can be addressed separately, the compatibility of these solutions may be another critical issue. To build a secure infrastructure in fog computing, one promising technique can be used is blockchain [216], which is distributed database that maintains a continuously-growing list of records. The blockchain is first used in bitcoin [216] to achieve secure online payments without the trusted administrator. In fog computing, the blockchain format can maintain the log files in a distributed way, which records the events occur in fog nodes and messages exchanged among the cloud, fog nodes and IoT devices. As a result, if a fog node misbehaves to cheat users, the cloud or other fog nodes, the honest fog nodes can discover the misbehavior based on the log files of the misbehaving fog node. Further, the blockchain can be used to build the authentication and reputation framework for all IoT devices and fog nodes. If a secure layer can be built by blockchain between the cloud and IoT devices, the fog computing would become a reliable, trustworthy and powerful architecture. Although the advantages of building a secure middle layer in Cloud-Fog-IoT framework is obvious, the overhead on a single fog node would be heavy due to its restricted storage resource and overhead on the blockchain management. In summary, how to build a scalable, efficient and decentralized secure infrastructure is challenging but important for the healthy development of fog computing.

VII. Conclusion

Fog computing is a new decentralized architecture that revolutionizes the cloud computing by extending storage, computing and networking resources to the network edge for supporting extremely large-scale IoT applications. However, it is also confronted with traditional security threats, which raise various new security and privacy challenges towards users. In this article, we have provided a comprehensive survey of securing fog computing for IoT applications. We have first reviewed the architecture and the features of fog computing. We have also discussed the roles of fog nodes in IoT applications, including real-time services, transient storage, data dissemination and decentralized computation, and examined several promising IoT applications according to different roles of fog nodes. We have presented the security and privacy threats in fog computing, including a series of security attacks and privacy exposure risks. Moveover, we have demonstrated the security and privacy challenges, reviewed the state-of-art solutions to secure fog computing in IoT applications, and showed our insights on the leaving problems to encourage further research on the security and privacy issues. Finally, we have identified several open research issues which should hold the most premise in terms of security and privacy issues in fog computing.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[2] J. Gubbia, R. Buyyab, S. Marusica, and M. Palaniswamia, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] D. Evans, "The Internet of Things: How the next evolution of the Internet is changing everything," Cisco, San Jose, CA, USA, White Paper, 2011.

[4] J. Camhi, *Former Cisco CEO John Chambers Predicts 500 Billion Connected Devices by 2025*, Bus. Insider, New York, NY, USA, 2015.

[5] *Cisco Global Cloud Index: Forecast and Methodology, 2014–2019*, Cisco, San Jose, CA, USA, White Paper, 2014.

[6] C. MacGillivray *et al.*, *IDC FutureScape: Worldwide Internet of Things 2016 Predictions*, document US40755816, IDC FutureScape, Framingham, MA, USA, 2015.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[8] "Fog computing and the Internet of Things: Extend the cloud to where the things are," Cisco, San Jose, CA, USA, White Paper, 2015.

[9] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, 2014.

[10] T. Zhang, *Fog Boosts Capabilities to Add More Things Securely to the Internet*, Cisco Blogs, San Jose, CA, USA, 2016.

[11] Wikipedia. (2016). *Fog Computing*. [Online]. Available: https://en.wikipedia.org/wiki/Fog-computing

[12] R. Roman, J. Lopez, and M. Manbo, "Mobile edge computing, fog *et al.*: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[13] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 2015.

[14] K. Zhang, X. Liang, R. Lu, K. Yang, and X. S. Shen, "Exploiting mobile social behaviors for Sybil detection," in *Proc. INFOCOM*, Hong Kong, 2015, pp. 271–279.

[15] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.

[16] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Mobidata*, Hangzhou, China, 2015, pp. 37–42.

[17] S. Sarkar and S. Misra, "Theoretical modelling of fog computing: A green computing paradigm to support IoT applications," *IET Netw.*, vol. 5, no. 2, pp. 23–29, Mar. 2016.

[18] B. Varghese, N. Wang, D. S. Nikolopoulos, and R. Buyya, "Feasibility of fog computing," ArXiv Preprint, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1701.05451.pdf

[19] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, to be published.

[20] K. Dantu, S. Y. Ko, and L. Ziarek, "RAINA: Reliability and adaptability in android for fog computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 41–45, Apr. 2017.

[21] P. Varshney and Y. Simmhan, "Demystifying fog computing: Characterizing architectures, applications and abstractions," ArXiv Preprint, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1702.06331.pdf

[22] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surveys*, vol. 50, no. 3, 2017, Art. no. 32.

[23] M. Tao, K. Ota, and M. Dong, "Foud: Integrating fog and cloud for 5G-enabled V2G networks," *IEEE Netw.*, vol. 31, no. 2, pp. 8–13, Mar./Apr. 2017.

[24] E. K. Markakis *et al.*, "EXEGESIS: Extreme edge resource harvesting for a virtualized fog environment," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 173–179, Jul. 2017.

[25] H. Dubey *et al.*, "Fog computing in medical Internet-of-Things: Architecture, implementation, and applications," ArXiv Preprint, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1706.08012.pdf

[26] A. M. Rahmani *et al.*, "Exploiting smart E-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.

[27] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, "Fog computing in healthcare—A review and discussion," *IEEE Access*, vol. 5, pp. 9206–9222, 2017.

[28] B. Hayes, "Cloud Computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, 2008.

[29] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 404–413, Apr. 2017.

[30] N. C. Luong *et al.*, "Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2546–2590, 4th Quart., 2016.

[31] A. Botta, W. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[32] H. Li, M. Dong, and K. Ota, "Radio access network virtualization for the social Internet of Things," *IEEE Cloud Comput.*, vol. 2, no. 6, pp. 42–50, Nov./Dec. 2015.

[33] F. Bonomi, R. Milito, J. Zhu, and S. Addepali, "Fog computing and its role in the Internet of Things," in *Proc. MCC Workshop*, Helsinki, Finland, 2012, pp. 13–16.

[34] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing IoT applications in the fog: A distributed dataflow approach," in *Proc. IOT*, Seoul, South Korea, 2015, pp. 155–162.

[35] X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan, and G.-J. Ren, "Foggy clouds and cloudy fogs: A real need for coordinated management of fog-to-cloud computing systems," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 120–128, Oct. 2016.

[36] *Automatic Identification and Data Capture Techniques—Data Structures—Digital Signature Meta Structure*, ISO/IEC Standard CD 20248, 2016.

[37] K. Liang, L. Zhao, X. Chu, and H.-H. Chen, "An integrated architecture for software defined and virtualized radio access networks with fog computing," *IEEE Netw.*, vol. 31, no. 1, pp. 80–87, Jan./Feb. 2017.

[38] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. Leung, "Fog radio access networks: Mobility management, interference mitigation and resource optimization," ArXiv Preprint, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1707.06892.pdf

[39] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. S. Shen, "Catalyzing cloud-fog interoperation in 5G wireless networks: An SDN approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14–20, Sep. 2017.

[40] Y. Cao, P. Hou, D. Brown, J. Wang, and S. Chen, "Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing," in *Proc. Mobidata*, Hangzhou, China, 2015, pp. 43–48.

[41] M. A. A. Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 161–169, Apr. 2016.

[42] N. Kamiyama *et al.*, "Priority control based on website categories in edge computing," in *Proc. IEEE INFOCOM WKSHPS*, San Francisco, CA, USA, 2016, pp. 776–781.

[43] W. Shi and S. Dustdar, "The promise of edge computing," *Computers*, vol. 49, no. 5, pp. 78–81, May 2016.

[44] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[45] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases & future directions," *IEEE Commun. Surveys Tuts.*, to be published.

[46] M. Chiang, S. Ha, I. Chih-Lin, F. Risso, and T. Zhang, "Clarifying fog computing and networking: 10 questions and answers," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 18–20, Apr. 2017.

[47] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[48] V. K. Sehgal, A. Patrick, A. Soni, and L. Rajput, "Smart human security framework using Internet of Things, cloud and fog computing," in *Proc. Intell. Distrib. Comput.*, New Delhi, India, 2015, pp. 251–263.

[49] V. Gazis, "A survey of standards for machine-to-machine and the Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 482–511, 1st Quart., 2017.

[50] J. Ni, X. Lin, K. Zhang, and X. Shen, "Privacy-preserving real-time navigation system using vehicular crowdsourcing," in *Proc. IEEE VTC Fall*, Montreal, QC, Canada, 2016, pp. 1–5.

[51] S. Chen, M. Li, K. Ren, and C. Qiao, "Crowd map: Accurate reconstruction of indoor floor plans from crowdsourced sensor-rich videos," in *Proc. ICDCS*, Columbus, OH, USA, 2015, pp. 1–10.

[52] J. Shropshire, "Extending the cloud with fog: Security challenges and opportunities," in *Proc. Inf. Syst.*, Savannah, GA, USA, 2014, pp. 1–10.

[53] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[54] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and P. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[55] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[56] A. Munir, P. Kansakar, and S. U. Khan, "IFCIoT: Integrated fog cloud IoT architectural paradigm for future Internet of Things," ArXiv Preprint, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1701.08474.pdf

[57] T. H. Luan *et al.*, "Fog computing: Focusing on mobile users at the edge," ArXiv Preprint, Tech. Rep., 2016. [Online]. Available: https://arxiv.org/pdf/1502.01815.pdf

[58] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computers*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[59] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao, "Rise of the indoor crowd: Reconstruction of building interior view via mobile crowdsourcing," in *Proc. ACM SenSys*, Seoul, South Korea, 2015, pp. 59–71.

[60] H. Gupta, S. B. Nath, S. Chakraborty, and S. K. Ghosh, "SDFog: A software defined computing architecture for QoS aware service orchestration over edge devices," ArXiv Preprint, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/pdf/1609.01190.pdf

[61] R. Lu, X. Lin, H. Zhu, and X. Shen, "An intelligent secure and privacy-preserving parking scheme through vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 6, pp. 2772–2784, Jul. 2010.

[62] J. Ni, K. Zhang, X. Lin, Y. Yu, and X. Shen, "Cloud-based privacy-preserving parking navigation through vehicular communications," in *Proc. SecureComm*, Guangzhou, China, 2016, pp. 85–103.

[63] T. Zhang, *Malware Defense Is a Team Sport, Best Played by Collaborative Fog Nodes*, Cisco Blogs, San Jose, CA, USA, 2016.

[64] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4924–4938, Aug. 2017.

[65] Y. Wang, M. Sheng, X. Wang, L. Wang, and J. Li, "Mobile-edge computing: Partial computation offloading using dynamic voltage scaling," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4268–4282, Oct. 2016.

[66] S. S. M. Chow, M. H. Au, and W. Susilo, "Server-aided signatures verification secure against collusion attack," *Inf. Security Tech. Rep.*, vol. 17, no. 3, pp. 46–57, 2013.

[67] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.

[68] K. Zhang *et al.*, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 122–129, Jan. 2017.

[69] *Smart Home Devices Used As Weapons in Website Attack*, BBC News, London, U.K., Oct. 2016. [Online]. Available: www.bbc.com/news/technology-37738823

[70] Kaspersky Lab. (2015). *Top 7 Mobile Security Threats: Smart Phones, Tablets, & Mobile Internet Devices—What the Future Has in Store*. [Online]. Available: https://usa.kaspersky.com/

[71] O. T. T. Kim, N. D. Tri, V. D. Nguyen, N. H. Tran, and C. S. Hong, "A shared parking model in vehicular network using fog and cloud environment," in *Proc. APNOMS*, Busan, South Korea, 2015, pp. 1–6.

[72] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[73] S. Gao, Z. Peng, B. Xiao, Q. Xiao, and Y. Song, "SCoP: Smartphone energy saving by merging push services in fog computing," in *Proc. IWQoS*, 2017, pp. 1–10.

[74] T. N. Gia *et al.*, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. CIT/IUCC/DASC/PICOM*, Liverpool, U.K., 2015, pp. 1–8.

[75] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 146–152, Jun. 2017.

[76] M. Aazam and E.-N. Huh, "E-HAMC: Leveraging fog computing for emergency alert service," in *Proc. PerCom Workshops*, St. Louis, MO, USA, 2015, pp. 1–6.

[77] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, "Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1174–1184, Oct. 2017.

[78] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: A location difference-based proximity detection protocol for fog computing," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1117–1124, Oct. 2017.

[79] R. Yang *et al.*, "Position based cryptography with location privacy: A step for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 799–806, Jan. 2018.

[80] X. Chen and L. Wang, "Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—PEPA," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 745–748, Apr. 2017.

[81] J. Zhou, Z. Cao, X. Dong, X. Lin, and A. V. Vasilakos, "Securing m-healthcare social networks: Challenges, countermeasures and future directions," *IEEE Wireless Commun.*, vol. 20, no. 4, pp. 12–21, Aug. 2013.

[82] H. Wang *et al.*, "Resource-aware secure ECG healthcare monitoring through body sensor networks," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 12–19, Feb. 2010.

[83] S. Chandrasekhar and M. Singhal, "Efficient and scalable query authentication for cloud-based storage systems with multiple data sources," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 520–533, Jul./Aug. 2017.

[84] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, to be published.

[85] A. Wasef and X. Shen, "EMAP: Expedite message authentication protocol for vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 78–89, Jan. 2013.

[86] Q. Jiang *et al.*, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 76, pp. 37–48, Dec. 2016.

[87] P. Hu *et al.*, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1143–1155, Oct. 2017.

[88] C. Li, Z. Qin, E. Novak, and Q. Li, "Securing SDN infrastructure of IoT–fog networks from MitM attacks," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1156–1164, Oct. 2017.

[89] X. Lin and X. Li, "Achieving efficient cooperative message authentication in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3339–3348, Sep. 2013.

[90] J. Zhou, X. Lin, X. Dong, and Z. Cao, "PSMPA: Patient self-controllable and multi-level privacy-preserving cooperative authentication in distributed m-healthcare cloud computing system," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 6, pp. 1693–1703, Jun. 2015.

[91] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in VANETs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 86–96, Jan. 2012.

[92] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Proc. CT RSA*, San Francisco, CA, USA, 2016, pp. 111–126.

[93] K. Lefevre, D. J. Dewitt, and R. Ramakrishnan, "Incognito: Efficient full-domain K-anonymity," in *Proc. SIGMOD*, Baltimore, MD, USA, 2005, pp. 49–60.

[94] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. IEEE S P*, Berkeley, CA, USA, 2009, pp. 173–187.

[95] S. Salonikias, I. Mavridis, and D. Gritzalis, "Access control issues in utilizing fog computing for transport infrastructure," in *Proc. CRITIS*, 2015, pp. 15–26.

[96] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT*, Tallinn, Estonia, 2011, pp. 568–588.

[97] K. Hess, *10 BYOD Mobile Device Management Suites You Need to Know*, ZDNet, San Francisco, CA, USA, 2012.

[98] P. Steiner, "Going beyond mobile device management," *Comput. Fraud Security*, vol. 2014, no. 4, pp. 19–20, 2014.

[99] J. Ni, X. Lin, K. Zhang, Y. Yu, and X. S. Shen, "Device-invisible two-factor authenticated key agreement protocol for BYOD," in *Proc. ICCC*, Chengdu, China, 2016, pp. 1–6.

[100] J. Zhang, Q. Li, X. Wang, B. Feng, and D. Guo, "Towards fast and lightweight spam account detection in mobile social networks through tog computing," *Peer Peer Netw. Appl.*, 2017.

[101] M. Katagi and S. Moriai. (2011). *Lightweight Cryptography for the Internet of Things*. [Online]. Available: https://www.iab.org/wp-content/IAB-uploads/2011/03/Kaftan.pdf

[102] "Final lightweight cryptography status report," Eur. Netw. Excellence Cryptol. II, Tech. Rep., 2012. [Online]. Available: http://www.ecrypt.eu.org/ecrypt2/documents/D.SYM.12.pdf

[103] K. Vieira, A. Schulter, C. Westphall, and C. Westphall, "Intrusion detection for grid and cloud computing," *IEEE IT Prof. Mag.*, vol. 12, no. 4, pp. 38–43, Jul./Aug. 2010.

[104] J. Xu, J. Arshad, and P. Townend, "An abstract model for integrated intrusion detection and severity analysis for clouds," *Int. J. Cloud Appl. Comput.*, vol. 1, no. 1, pp. 1–17, 2011.

[105] H. Hamad and M. A. Hoby, "Managing intrusion detection as a service in cloud networks," *Int. J. Comput. Appl.*, vol. 41, no. 1, pp. 35–40, 2012.

[106] A. Houmansadr, S. A. Zonouz, and R. Berthier, "A cloud-based intrusion detection and response system for mobile phones," in *Proc. DSNW*, 2011, pp. 31–32.

[107] A. V. Dastjerdi, K. A. Bakar, and S. G. H. Tabatabaei, "Distributed intrusion detection in clouds using mobile agents," in *Proc. ADVCOMP*, Sliema, Malta, 2009, pp. 175–180.

[108] S. Ram, "Secure cloud computing based on mutual intrusion detection system," *Int. J. Comput. Appl.*, vol. 2, no. 1, pp. 57–67, 2012.

[109] J. Liu *et al.*, "Secure intelligent traffic light control using fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 817–824, Jan. 2018.

[110] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proc. ACM CCS*, Chicago, IL, USA, 2009, pp. 21–32.

[111] D. Quercia and S. Hailes, "Sybil attacks against mobile users: Friends and foes to the rescue," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010, pp. 336–340.

[112] S. Park, B. Aslam, D. Turgut, and C. C. Zou, "Defense against Sybil attack in the initial deployment stage of vehicular ad hoc network based on roadside unit support," *Security Commun. Netw.*, vol. 6, no. 4, pp. 523–538, 2013.

[113] X. Lin, "LSR: Mitigating zero-day Sybil vulnerability in privacy-preserving vehicular peer-to-peer networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 237–246, Sep. 2013.

[114] X. Liang, X. Lin, and X. S. Shen, "Enabling trustworthy service evaluation in service-oriented mobile social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 310–320, Feb. 2014.

[115] H. Li and M. Singhal, "Trust management in distributed systems," *IEEE Comput.*, vol. 40, no. 2, pp. 45–53, Feb. 2007.

[116] H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser, "A survey of multi-agent trust management systems," *IEEE Access*, vol. 1, pp. 35–50, 2013.

[117] Z. Wei, H. Tang, F. R. Yu, M. Wang, and P. Mason, "Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4647–4658, Nov. 2014.

[118] Z. Su *et al.*, "Toward architectural and protocol-level foundation for end-to-end trustworthiness in cloud/fog computing," *IEEE Trans. Big Data*, to be published.

[119] J. Weng, C. Miao, and A. Goh, "An entropy-based approach to protecting rating systems from unfair testimonies," *IEICE Trans. Inf. Syst.*, vol. E89-D, no. 9, pp. 2502–2511, Sep. 2006.

[120] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social Internet of Things," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1253–1266, May 2014.

[121] J.-H. Cho, A. Swami, and I.-R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 562–583, 4th Quart., 2011.

[122] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, "A reputation-based announcement scheme for VANETs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 4095–4108, Nov. 2012.

[123] I.-R. Chen, F. Bao, M. J. Chang, and J.-H. Cho, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 5, pp. 1200–1210, May 2014.

[124] J. Ren, Y. Zhang, K. Zhang, and X. Shen, "SACRM: Social aware crowdsourcing with reputation management in mobile sensing," *Comput. Commun.*, vol. 65, pp. 55–65, Jul. 2015.

[125] J. Singh, T. Pasquier, J. Bacon, H. Ko, and D. Eyers, "Twenty security considerations for cloud-supported Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 269–284, Jun. 2015.

[126] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Trans. Services Comput.*, vol. 5, no. 2, pp. 220–232, Apr./Jun. 2012.

[127] Y. Yu *et al.*, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 767–778, Apr. 2017.

[128] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2231–2244, Dec. 2011.

[129] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. EUROCRYPT*, Espoo, Finland, 1998, pp. 127–144.

[130] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Security*, vol. 9, no. 1, pp. 1–30, 2006.

[131] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," in *Proc. ACM CCS*, Alexandria, VA, USA, 2007, pp. 185–194.

[132] J. Weng, R. H. Deng, X. Ding, C.-K. Chu, and J. Lai, "Conditional proxy re-encryption secure against chosen-ciphertext attack," in *Proc. ACM ASIACCS*, Sydney, NSW, Australia, 2009, pp. 322–332.

[133] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. EUROCRYPT*, Aarhus, Denmark, 2005, pp. 457–473.

[134] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE S P*, 2007, pp. 321–334.

[135] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM CCS*, Alexandria, VA, USA, 2006, pp. 89–98.

[136] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. ACM CCS*, Chicago, IL, USA, 2010, pp. 735–737.

[137] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.

[138] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, and X. Cheng, "An attribute-based encryption scheme to secure fog communications," *IEEE Access*, vol. 5, pp. 9131–9138, 2017.

[139] A. Alotaibi, A. Barnawi, and M. Buhari, "Attribute-based secure data sharing with efficient revocation in fog computing," *Int. J. Inf. Security*, vol. 8, no. 3, pp. 203–222, 2017.

[140] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 720–729, Jan. 2018.

[141] Z. Yu, M. H. Au, Q. Xu, R. Yang, and J. Han, "Towards leakage-resilient fine-grained access control in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 763–777, Jan. 2018.

[142] C.-K. Chu, S. S. M. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 468–477, Feb. 2014.

[143] B. Cui, Z. Liu, and L. Wang, "Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2374–2385, Aug. 2014.

[144] C.-C. Lee, C.-T. Li, S.-T. Chiu, and S.-D. Chen, "Time-bound key-aggregate encryption for cloud storage," *Security Commun. Netw.*, vol. 9, no. 13, pp. 2059–2069, 2016.

[145] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, Prague, Czech Republic, 1999, pp. 223–238.

[146] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. TCC*, Cambridge, MA, USA, 2005, pp. 325–341.

[147] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *Proc. INFOCOM*, Orlando, FL, USA, 2012, pp. 1969–1977.

[148] X. Liang *et al.*, "Fully anonymous profile matching in mobile social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 641–655, Sep. 2013.

[149] J. Ni *et al.*, "Differentially private smart metering with fault tolerance and range-based filtering," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2483–2493, Sep. 2017.

[150] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[151] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 712–719, Jan. 2018.

[152] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, Bethesda, MD, USA, 2009, pp. 169–178.

[153] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Proc. EUROCRYPT*, 2010, pp. 24–43.

[154] G. Zhuo, Q. Jia, L. Guo, M. Li, and P. Li, "Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, 2016, pp. 1–9.

[155] J. Zhou, Z. Cao, X. Dong, and X. Lin, "Security and privacy in cloud-assisted wireless wearable communications: Challenges, solutions, and future directions," *IEEE Wireless Commun.*, vol. 22, no. 2, pp. 136–144, Apr. 2015.

[156] J. Xia and Y. Wang, "Secure key distribution for the smart grid," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1437–1443, Sep. 2012.

[157] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "EDAT: Efficient data aggregation without TTP for privacy-assured smart metering," in *Proc. ICC*, Kuala Lumpur, Malaysia, 2016, pp. 1–6.

[158] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proc. EUROCRYPT*, Warsaw, Poland, 2003, pp. 416–432.

[159] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," *Comput. Commun.*, vol. 32, no. 6, pp. 1079–1085, 2009.

[160] M. Bellare and G. Neven, "Identity-based multi-signatures from RSA," in *Proc. CT RSA*, San Francisco, CA, USA, 2006, pp. 145–162.

[161] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters, "Sequential aggregate signatures and multisignatures without random Oracles," in *Proc. EUROCRYPT*, St. Petersburg, Russia, 2006, pp. 465–485.

[162] D. Derler and D. Slamanig, "Key-homomorphic signatures and applications to multiparty signatures," Cryptol. ePrint Arch., Tech. Rep. 792, 2016. [Online]. Available: https://eprint.iacr.org/2016/792.pdf

[163] J. Ni, X. Lin, K. Zhang, and Y. Yu, "Secure and deduplicated spatial crowdsourcing: A fog-based approach," in *Proc. Globecom*, Washington, DC, USA, 2016, pp. 1–6.

[164] J. H. Ahn, M. Green, and S. Hohenberger, "Synchronized aggregate signatures: New definitions, constructions and applications," in *Proc. ACM CCS*, Chicago, IL, USA, 2010, pp. 473–484.

[165] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE S P*, Berkeley, CA, USA, 2000, pp. 44–55.

[166] J. Li *et al.*, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010, pp. 441–445.

[167] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. ICDCS*, Genoa, Italy, 2010, pp. 253–262.

[168] P. Rizomiliotis and S. Gritzalis, "ORAM based forward privacy preserving dynamic searchable symmetric encryption schemes," in *Proc. CCSW*, Denver, CO, USA, 2015, pp. 65–76.

[169] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage," in *Proc. IEEE S P*, Berkeley, CA, USA, 2014, pp. 639–654.

[170] X. Yang, F. Yin, and X. Tang, "A fine-grained and privacy-preserving query scheme for fog computing-enhanced location-based service," *Sensors*, vol. 17, no. 7, p. 1611, 2017.

[171] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. EUROCRYPT*, Interlaken, Switzerland, 2004, pp. 506–522.

[172] H. S. Rhee, J. H. Park, W. Susilo, and D. H. Lee, "Improved searchable public key encryption with designated tester," in *Proc. ASIACCS*, Sydney, NSW, Australia, 2009, pp. 376–379.

[173] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proc. Pairing*, Tokyo, Japan, 2007, pp. 2–22.

[174] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proc. EUROCRYPT*, Istanbul, Turkey, 2008, pp. 146–162.

[175] V. Iovino and G. Persiano, "Hidden-vector encryption with groups of prime order," in *Proc. Pairing*, Beijing, China, 2008, pp. 75–88.

[176] M. S. Islam, M. Kuzu, and M. Kantarcioglu, "Access pattern disclosure on searchable encryption: Ramification, attack and mitigation," in *Proc. NDSS*, San Diego, CA, USA, 2012, pp. 1–15.

[177] F. Hahn and F. Kerschbaum, "Searchable encryption with secure and efficient updates," in *Proc. ACM CCS*, Scottsdale, AZ, USA, 2014, pp. 310–320.

[178] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz, "An architecture for a secure service discovery service," in *Proc. MobiCom*, Seattle, WA, USA, 1999, pp. 24–35.

[179] D. J. Wu, A. Taly, A. Shankar, and D. Boneh, "Privacy, discovery, and authentication for the Internet of Things," in *Proc. ESORICS*, Oslo, Norway, 2016, pp. 301–319.

[180] J. H. Park, H. J. Kim, M. H. Sung, and D. H. Lee, "Public key broadcast encryption schemes with shorter transmissions," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 401–411, Sep. 2008.

[181] M. Abdalla, Y. Shavitt, and A. Wool, "Key management for restricted multicast using broadcast encryption," *IEEE/ACM Trans. Netw.*, vol. 8, no. 4, pp. 443–454, Aug. 2000.

[182] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2005, pp. 258–275.

[183] B. Libert, K. G. Paterson, and E. A. Quaglia, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," in *Proc. PKC*, Darmstadt, Germany, 2012, pp. 206–224.

[184] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "SAGE: A strong privacy-preserving scheme against global eavesdropping for eHealth systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 4, pp. 365–378, May 2009.

[185] R. Lu, X. Lin, and X. Shen, "SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, 2010, pp. 1–9.

[186] X. Lin, R. Lu, X. Liang, and X. Shen, "STAP: A social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in VANETs," in *Proc. IEEE INFOCOM*, Shanghai, China, 2011, pp. 2147–2155.

[187] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2010, pp. 465–482.

[188] A. C. Yao, "Protocols for secure computations," in *Proc. SFCS*, Chicago, IL, USA, 1982, pp. 160–164.

[189] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Proc. CRYPTO*, Santa Barbara, CA, USA, 2010, pp. 483–501.

[190] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in *Proc. TCC*, Taormina, Italy, 2012, pp. 422–439.

[191] C. Papamanthou, E. Shi, and R. Tamassia, "Signatures of correct computation," in *Proc. TCC*, Tokyo, Japan, 2013, pp. 222–242.

[192] S. G. Choi, J. Katz, R. Kumaresan, and C. Cid, "Multi-client non-interactive verifiable computation," in *Proc. TCC*, Tokyo, Japan, 2013, pp. 499–518.

[193] S. D. Gordon, J. Katz, F.-H. Liu, E. Shi, and H.-S. Zhou, "Multi-client verifiable computation with stronger security guarantees," in *Proc. TCC*, Warsaw, Poland, 2015, pp. 144–168.

[194] K. Elkhiyaoui, M. Önen, M. Azraoui, and R. Molva, "Efficient techniques for publicly verifiable delegation of computation," in *Proc. ACM CCS*, Xi'an, China, 2016, pp. 119–128.

[195] T. Matsumoto, K. Kato, and H. Iami, "Speeding up secret computations with insecure auxiliary devices," in *Proc. Crypto*, Santa Barbara, CA, USA, 1988, pp. 497–506.

[196] S. Kawamura and A. Shimbo, "Fast server-aided secret computation protocols for modular exponentiation," *IEEE J. Sel. Areas Commun.*, vol. 11, no. 5, pp. 778–784, Jun. 1993.

[197] B. Cavallo, G. D. Crescenzo, D. Kahrobaei, and V. Shpilrain, "Efficient and secure delegation of group exponentiation to a single server," in *Proc. RFIDSec*, New York, NY, USA, 2015, pp. 156–173.

[198] M. Girault and D. Lefranc, "Server-aided verification: Theory and practice," in *Proc. ASIACRYPT*, Chennai, India, 2005, pp. 605–623.

[199] W. Wu, Y. Mu, W. Susilo, and X. Huang, "Server-aided verification signatures: Definitions and new constructions," in *Proc. ProvSec*, Nanjing, China, 2008, pp. 141–155.

[200] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, 2004.

[201] M. Bellare, S. Keelveedhi, and T. Ristenpart, "DupLESS: Server-aided encryption for deduplicated storage," in *Proc. Usenix Security*, Washington, DC, USA, 2014, pp. 179–194.

[202] S. Kamara, P. Mohassel, and B. Riva, "Salus: A system for server-aided secure function evaluation," in *Proc. ACM CCS*, Raleigh, NC, USA, 2012, pp. 797–808.

[203] Y. Cliff, Y. S. T. Tin, and C. Boyd, "Password based server aided key exchange," in *Proc. ACNS*, Singapore, 2006, pp. 146–161.

[204] T. Wang *et al.*, "Trajectory privacy preservation based on a fog structure for cloud location services," *IEEE Access*, vol. 5, pp. 7692–7701, 2017.

[205] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE Access*, vol. 4, pp. 2751–2763, 2016.

[206] C. Dwork, "Differential privacy," in *Proc. ICALP*, Venice, Italy, 2006, pp. 1–12.

[207] L. Li, R. Lu, K.-K. R. Choo, A. Datta, and J. Shao, "Privacy-preserving-outsourced association rule mining on vertically partitioned databases," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1847–1861, Aug. 2016.

[208] K. Zhang, X. Liang, J. Ni, K. Yang, and X. Shen, "Exploiting social network to enhance human-to-human infection analysis without privacy leakage," *IEEE Trans. Depend. Secure Comput.*, to be published.

[209] X. Liu, R. H. Deng, Y. Yang, H. N. Tran, and S. Zhong, "Hybrid privacy-preserving clinical decision support system in fog—Cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 825–837, Jan. 2018.

[210] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM CCS*, Vienna, Austria, 2016, pp. 308–318.

[211] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[212] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM CCS*, Scottsdale, AZ, USA, 2014, pp. 1054–1067.

[213] Z. Qin *et al.*, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proc. ACM CCS*, Vienna, Austria, 2016, pp. 192–203.

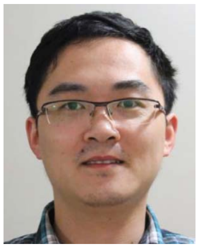
[214] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput. Pract. Exp.*, vol. 28, no. 10, pp. 2991–3005, 2016.

[215] A. Narayanan and V. Shmatikov, "Myths and fallacies of 'personally identifiable information,"' *Commun. ACM*, vol. 53, no. 6, pp. 24–26, 2010.

[216] J. Bonneau *et al.*, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE S P*, San Jose, CA, USA, 2015, pp. 104–121.

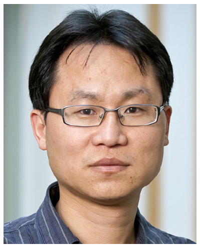
[217] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 5, pp. 877–882, 2012.

[218] B. Kaliski, "A survey of encryption standards," *IEEE Micro*, vol. 13, no. 6, pp. 74–81, Dec. 1993.



Jianbing Ni (S'16) received the B.E. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests are applied cryptography and information security, with current focus on cloud computing, smart grid, and Internet of Things.

Kuan Zhang (S'13–M'17) received the B.Sc. degree in communication engineering and the M.Sc. degree in computer applied technology from Northeastern University, China, in 2009 and 2011, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2016, where he was also a Post-Doctoral fellow with the Broadband Communications Research Group. Since 2017, he has been an Assistant Professor with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, USA. His research interests include security and privacy for mobile social networks, e-healthcare systems, cloud computing, and cyber physical systems.



Xiaodong Lin (M'09–SM'12–F'17) received the Ph.D. degree in information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, and the second Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada. He was an Associate Professor of information security with the faculty of Business and Information Technology, University of Ontario Institute of Technology, Canada. He is currently an Associate Professor with the Department of Physics and Computer Science, Wilfrid Laurier University, Waterloo. His research interests include wireless network security, applied cryptography, computer forensics, software security, and wireless networking and mobile computing. He was a recipient of the Outstanding Achievement in Graduate Studies Award from the University of Waterloo.

Xuemin (Sherman) Shen (M'97–SM'02–F'09) received the B.Sc. degree from Dalian Maritime University, China, in 1982, and the M.Sc. and Ph.D. degrees from Rutgers University, Camden, NJ, USA, in 1987 and 1990, respectively, all in electrical engineering. He is a University Professor from the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He was a recipient of the Excellent Graduate Supervision Award in 2006, the Outstanding Performance Award from the University of Waterloo, in 2004, 2007, 2010, and 2014, the Premier's Research Excellence Award from the Province of Ontario, Canada, in 2003, the Distinguished Performance Award from the Faculty of Engineering, University of Waterloo, in 2002 and 2007, and the Joseph LoCicero Award from the IEEE Communications Society. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom'16 and Infocom'14, the IEEE VTC'10 Fall, and Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring and the IEEE ICC'08, the General Co-Chair for ACM Mobihoc'15, Chinacom'07, and QShine'06, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, the IEEE NETWORK, *Peer-to-Peer Networking and Application*, and *IET Communications*, a Founding Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Computer Networks*, and *ACM/Wireless Networks*. He is a Registered Professional Engineer of ON, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.