Liu, Dan; Yan, Zheng; Ding, Wenxiu; Atiquzzaman, Mohammed

A survey on secure data analytics in edge computing

Published in:
IEEE Internet of Things Journal

DOI:
10.1109/JIOT.2019.2897619

Published: 01/06/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Liu, D., Yan, Z., Ding, W., & Atiquzzaman, M. (2019). A survey on secure data analytics in edge computing. *IEEE Internet of Things Journal*, *6*(3), 4946-4967. Article 8634892. https://doi.org/10.1109/JIOT.2019.2897619

A Survey on Secure Data Analytics in Edge Computing

Dan Liu, Zheng Yan, *Senior Member, IEEE,* Wenxiu Ding, *IEEE Member,* and Mohammed Atiquzzaman *Senior Member, IEEE*

***Abstract*—Internet of Things (IoT) is gaining increasing popularity. Overwhelming volumes of data are generated by IoT devices. Those data after analytics provide significant information that could greatly benefit IoT applications. Different from traditional applications, IoT applications such as environmental monitoring, smart navigation and smart healthcare come with new requirements such as mobility, real-time response, and location awareness. However, traditional cloud computing paradigm cannot satisfy these demands due to centralized processing and being far away from local devices. Hence, edge computing was introduced to perform data processing and storage in the edge of networks, which is closer to data sources than cloud computing, thus efficient and location-aware. Unfortunately, edge computing brings new security and privacy challenges when applied to data analytics. The literature still lacks a thorough review on the recent advances in secure data analytics in edge computing. In this paper, we first introduce the concept and features of edge computing, and then propose a number of requirements for its secure data analytics by analyzing potential security threats in edge computing. Furthermore, we give a comprehensive review on the pros and cons of the existing works on data analytics in edge computing based on our proposed requirements. Based on our literature survey, we highlight current open issues and propose future research directions.**

***Index Terms*—edge computing, data analytics, security, privacy preservation**

I. Introduction

RAPID development of Internet of Things (IoT) enables everything to communicate with each other, which has changed our life, work and study thoroughly. Many IoT applications, such as wearable devices, smart environment monitoring, smart healthcare and so on, have been widely applied in our daily life and offered great convenience. In order to benefit from IoT, an overwhelming number of sensors or devices are employed. Cisco has forecasted based on current trends that there will be 50 billion devices which are connected to the Internet by 2020 [1]. Besides, Cisco Global Cloud Index predicted that the data generated by things, human, machines would exceed 500 Zettabytes (ZB) by 2020, but the IP traffic of the global data center will only reach 10.4 ZB at that time [2]. Furthermore, Cisco's CEO predicted that about 500 billion user devices will join the Internet by 2025 [3]. As we all know, smart devices are not powerful enough to process these data efficiently due to limited computation and storage capacity. Hence, it becomes a serious issue to process the rapidly increasing volume of data to alleviate the heavy burden of networks.

Cloud computing was originally regarded as a promising computing infrastructure to mitigate the heavy burden on edge devices, since it can provide various services (such as computation, storage, and networking) for individuals, organizations and enterprises. The advantage of cloud computing is that cloud servers have abundant computing and storage resources to allow a very large quantity of users to access the services provided by cloud [4]. However, cloud computing is not efficient enough to support such distributed IoT environment due to three reasons. First, some IoT applications need to support real-time response, location awareness, context awareness and mobility, but cloud computing cannot satisfy these demands owing to centralization and being far away from user devices which will be discussed in detail in Section II.C. Second, if cloud computing is used to handle very large amount of raw data, the bandwidth of the current network could become a bottleneck, mainly owing to inevitable queuing delay. Third, the burden on cloud servers will increase and become a bottleneck for the increasing amount of service requests.

In order to overcome these issues, edge computing [5], [6] was introduced to extend cloud computing to the edge of networks. Edge computing is a new decentralized paradigm that can also provide data computation, storage and application services to end users, while it offers several advantages, such as real-time response, location awareness and mobility due to its proximity to terminal devices. It is suited for various scenarios, such as smart gird [7], smart traffic lights [8],

This work is sponsored by National Postdoctoral Program for Innovative Talents (grant BX20180238), the NSFC (grants 61672410, 61802293 and U1536202), the Academy of Finland (grant 308087), the National Key Research and Development Program of China (grant 2016YFB0800704), the Project Supported by Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2016ZDJC-06), the Key Lab of Information Network Security, Ministry of Public Security under grant No. C18614, the Project funded by China Postdoctoral Science Foundation (grant 2018M633461), and the 111 project (grants B16037 and B08038). *(Corresponding author: Wenxiu Ding.)*

D. Liu is with State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an, China (email: 15596173220@163.com).

Z. Yan is with the State Key Lab on Integrated Services Networks, School of Cyber Engineering, Xidian University, No.2 South Taibai Road, Xi'an, China, 710071; and the Department of Communications and Networking, Aalto University, Konemiehentie 2, P.O.Box 15400, Espoo, Finland (e-mail: zyan@xidian.edu.cn; zheng.yan@aalto.fi).

W. Ding is with the School of Cyber Engineering, Xidian University, Xi'an, China (email: wxding@xidian.edu.cn).

M. Atiquzzaman is with the Telecommunications and Networks Research Lab, School of Computer Science, University of Oklahoma, Norman, OK 73019-6151 USA (e-mail: atiq@ou.edu).

augmented reality applications [9], video streaming [10] and so on. Tang *et al.* [11] showed that edge computing can help heighten the efficiency and quality of services.

Though edge computing brings many benefits, it also faces a variety of security and privacy threats. On one hand, since edge computing is considered as an extension of cloud computing, it inherits some security issues from cloud computing. On the other hand, edge computing also faces security and privacy challenges because of its distinctive features, such as geographic distribution, heterogeneity, and low latency. For achieving secure data analytics, deploying security mechanisms is indispensable. Unfortunately, due to restricted resources of edge devices, typical security mechanisms proposed in cloud framework are not suitable for edge framework. Therefore, it is important to develop security solutions in edge computing to support reliable and efficient edge computing-based IoT applications.

Though there exist several surveys on security and privacy issues in edge computing [4], [7], [9], [12]–[20], it still lacks a comprehensive survey on security and privacy of data analytics in edge computing. Stojmenovic *et al.* [7], [15] only focused on man-in-the-middle attacks. Yi *et al.* [9] identified various issues when designing and implementing fog computing, but did not provided a comprehensive discussion on security issues of fog computing. Wang *et al.* [19] only highlighted the issues of fog forensics. Zhang *et al.* [20] only overviewed access control of user data in fog computing. Some researchers surveyed the security and privacy challenges in fog computing [4], [12], [13], but they did not discuss the exiting security solutions about fog computing. Although [18] and [14] analyzed the security problems and surveyed recent advance including secure and privacy-preserving schemes in fog computing. However, they did not focus on the data analytics in fog computing. A detailed comparison of our survey with other existing surveys on security and privacy issues in edge computing is presented in Table I. Obviously, the literature still lacks a thorough review on the recent advance of secure data analytics in edge computing. The objective of this paper is to provide an in-depth analysis of security threats in existing edge computing, and provide a framework to compare and contrast the effectiveness of existing security mechanisms for data analytics in edge computing.

In this paper, we provide a comprehensive overview of the existing efforts on secure data analytics in edge computing. We introduce the concept and features of edge computing, and then summarize the common security and privacy mechanisms for outsourcing data analytics. Besides, we first propose a number of requirements for secure data analytics by summarizing the security threats of data analytics in edge computing. Furthermore, we thoroughly review the existing works in edge computing by employing our proposed requirements as a measure to discuss their pros and cons. Finally, we point out some open issues and propose future research directions. The main contributions of this paper can be summarized as follows:

- We analyze the security threats of data analytics in edge computing and propose a number of security and performance requirements.
- We use the proposed security and performance requirements as a measure to comprehensively review and discuss existing data analytics schemes in edge computing.
- We highlight a number of open issues and further propose future research directions towards secure and privacy-preserving data analytics in edge computing.

The remainder of this paper is organized as follows. Section II presents the concept, architecture and features of edge computing, followed by existing main security and privacy mechanisms for outsourcing data analytics in Section III . In Section IV, we analyze the security threats of data analytics in edge computing and propose a number of requirements for evaluating the performance of secure data analytics. We provide a thorough literature review on secure data analytics in edge computing in Section V. We further highlight open issues and propose future research directions in Section VI. Finally, we conclude the paper in the last section.

II. Overview of Edge Computing

This section introduces the basic concepts related to edge computing, its three-layer architecture and features.

A. Basic Concept

Edge Computing: Edge computing is considered as a method of moving some of the cloud processing closer to user devices which require real-time interaction to make the best use of untapped computational capabilities in the edge of networks [21], [22]. In [23], edge computing refers to place application services, data and processing at extremes of a network instead of placing them centrally. Herein, "Edge" [6] is defined as any computing and network resources which are on the path between data sources and cloud service center. And edge computing requires that computing happens at the vicinity of data sources. Although edge computing and fog computing have some differences in concept [23], in fact, they are interchangeable in academia and industry. Thereby, we do not distinguish between these two terms in this paper.

Edge Node: Edge nodes are facilities or infrastructures that have computation and storage capabilities at the edge of network. They can be resource-limited devices, like set-top-boxes, road-side units, WiFi access points, gateways, routers, end devices, etc. They can also be resource-rich devices that usually possess powerful CPU and abundant storage spaces, such as cloudlet.

In recent years, some similar terms have emerged, such as Mobile Edge Computing (MEC), Mobile Cloud Computing (MCC) and fog computing. In [24], MEC is defined as an emergent model where cloud computing platform extends to the mobile base stations at the vicinity of mobile subscribers to support delay-sensitive and context-aware applications. MCC refers to an architecture where mobile users offload data processing and data storage to cloud computing [25]. Fog computing is regarded as a scenario in which a large number of heterogeneous and decentralized fog nodes can communicate and cooperate with each other and perform data storage and data processing tasks without the involvement of third parties [9]. All these paradigms are proposed to bring the capabilities

TABLE I
COMPARISON OF OUR SURVEY WITH OTHER EXISTING SURVEYS

Covered Topic	[4]	[7], [15]	[9]	[12]	[13]	[14]	[16]	[17]	[18]	[19]	[20]	Our Survey
Give a comprehensive review of security and privacy issues in edge computing	Y	N	N	Y	Y	Y	Y	Y	Y	N	N	Y
Compare cloud computing and edge computing in detail	Y	Y	N	Y	N	Y	Y	N	N	N	N	Y
Summarize mainstream attacks	Y	N	N	Y	N	N	N	N	Y	N	N	Y
Propose security and performance requirements of secure data analytics	N	N	N	N	N	N	N	N	N	N	N	Y
Compare the computational complexity of existing mechanisms	N	N	N	N	N	Y	N	N	Y	N	N	Y
Review secure data analytic	N	N	N	N	N	N	N	N	N	N	N	Y

Y: discussed; N: not discussed

of cloud servers to the edge of network, but there are some differences between them. With regards to architecture, data computing is still at a cloud server in MCC, while MEC and fog computing process data at the edge of networks. MEC treats mobile base stations as edge nodes to serve mobile subscribers. Fog computing is a generalized concept where fog nodes are not only base stations but also access points, routers, etc. Towards service delivery, MEC and fog computing can serve for such IoT applications that have strict requirements on low latency, location awareness, mobility and context awareness. However, MCC just makes use of the cloud to serve for mobile devices and does not consider these specific requirements.

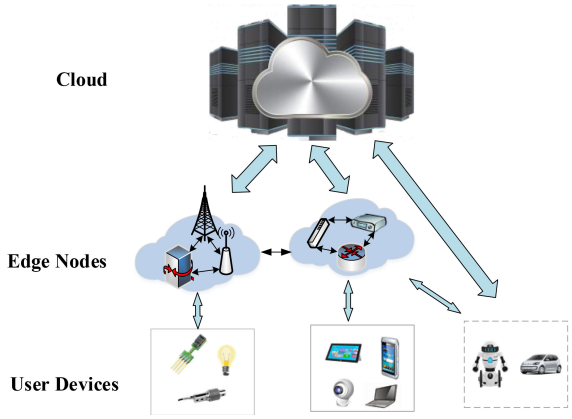

Fig. 1. Architecture of edge computing

B. Edge Computing Architecture

According to the aforementioned definitions, the architecture of edge computing is shown in Fig. 1. This framework can be divided into two categories. First, user devices with some computational power act as edge nodes to preprocess raw data and then pass them to the cloud server for further processing. However, when the computation task is so big that the user devices cannot handle it, the users will offload their computational tasks to adjacent edge nodes. For example, Shi *et al.* [6] proposed a case study where a lost child can be found via video analysis. The cloud sends the request of searching the child to all cameras in a targeted area. Then the cameras perform the search mission and return search results to the cloud. In this case, a variety of cameras work as edge nodes to execute the search request, which can release the burden of cloud and save search time compared to the method which relies on the cloud to perform search analysis. Second, user devices offload some computational tasks to adjacent edge nodes to do preprocessing (such as data compression and data fusion) and then the cloud does the final analysis. Herein, we mainly concentrate on the second situation. To achieve sound interactions among different layers (i.e., user device layer, edge node layer and cloud server layer), hybrid communication technologies are applied in edge computing, including wired communications (such as Ethernet, and optical fiber) and wireless communications (such as ZigBee, WiFi, and LTE) [26]. The communication between cloud server and edge nodes and the communication among edge nodes are usually supported by wired communication technologies, while edge nodes and IoT devices communicate with each other with wireless communication technologies. The architecture of edge computing is described as below.

The lowest layer is user device layer which is constituted by a large number of IoT devices, such as sensors, smart phones, smart wearable devices, and so on. Some of these devices are mobile IoT objects, and others are fixed IoT objects. Raw data can be generated or perceived by them and sent to a higher layer device to further process.

The middle layer is edge node layer. The edge nodes consist of the devices that possess some computing capability, such as base stations, routers, set-top boxes, switches, etc. The edge nodes can provide services to users, including computing offloading, transient data storage, content caching, and delivery services from the cloud to the users. Moreover, they can offload some computational tasks from the cloud in order to alleviate its burden. In addition, the edge nodes can cooperate with each other to provide collaborative services for users. For example, Shi *et al.* [6] introduced a use case of connected

health in which hospitals, pharmacies, logistics companies, governments, and insurance companies form a collaborative edge to provide health-care services.

The highest layer is cloud server layer. It conducts further data processing on preprocessed data from edge nodes, and it can also delegate computational tasks to the edge nodes. There are two situations where the cloud server requires to perform further data processing after edge node processing. First, when coordination among edge nodes is required, the cloud server can assist them to establish communications. Second, when data analytics are very large-scale (e.g., city-wide) or long-term (e.g., over years), the edge nodes normally send data to the cloud server for analysis [11].

C. Features of Edge Computing

The involvement of the edge node layer makes the edge computing different from the cloud computing in several aspects. The detailed comparison between the cloud computing and the edge computing is summarized as below and also shown in Table II.

TABLE II
FEATURE COMPARISON BETWEEN CLOUD COMPUTING AND EDGE COMPUTING

Features	Edge Computing	Cloud Computing
Location Awareness	Yes	No
Geographic Position	Fixed Positions	Various Positions
Latency	Low	High
Large-Scale IoT Application Support	Yes	No
Network Architecture	Decentralized	Centralized
Hardware	Heterogeneous Devices	General Devices
Mobility	Yes	No

Location Awareness: Location awareness refers to the ability to determine the geographical location of a user device. Location awareness can be used for targeted advertisement and entertainment. The cloud computing does not offer location awareness services. When a cloud server needs to know the location of users, Location-Based Service (LBS) can be offered. In this service, users have to send their location information to the cloud server, which could incur expensive communication overload. Furthermore, it causes location privacy leakage of users [16]. In contrast, in the edge computing, an edge node is aware of user devices in its own coverage area and the users do not need to send their local information to a remote third party, like the cloud server.

Geographic Distribution: In the cloud computing, a cloud server processes data in central cloud servers, which are deployed at some fixed places. However, edge nodes are deployed at various positions, like highways, roadways, supermarkets, museum floors, etc. Due to geographic distribution, the edge nodes can acquire high-quality data streams from IoT devices and provide real-time response to users.

Latency: Generally, the services provided by the cloud server are far away from IoT devices, which leads to long data transmission latency. This is tolerable for non-real-time applications (e.g., offline games) but intolerable for real-time applications (e.g., augmented reality and health emergency). On the contrary, edge nodes are closer to IoT devices, data transmission between them takes short time.

Large-Scale IoT Application Support: Due to heavy management and computational overhead, the cloud computing cannot provide services for large-scale IoT applications. For example, in a wide range of environment monitoring system, an overwhelming volume of data are produced by massive sensors. If these sensors are managed and the data processing is performed in the central cloud, the burden of the cloud server could be huge. However, in the edge computing, the edge nodes have power and autonomy to manage these IoT devices in their own areas, thus erase the shortcoming of the cloud computing in terms of large-scale IoT application support.

Network Architecture: In the cloud computing, there is a centralized server to manage computation and storage resources. Nevertheless, the edge computing paradigm is a decentralized framework since each edge node self-organizes to offer real-time application services to users.

Hardware: In the paradigm of cloud computing, data is produced by some enterprises and there are only a few central cloud servers to offer services. In the edge computing, however, hardware devices are heterogeneous. Heterogeneity is a distinct feature of the edge computing. This reflects in three aspects. First, data producers are heterogeneous, that is, data are generated by heterogeneous IoT devices with various formats. Second, data transmission is heterogenous, that is, collected data is transmitted by using different communication technologies. Third, edge nodes are heterogenous. This means that services are deployed in multiple types of edge nodes, including end-user devices, access points, routers, switches, and so on.

Mobility: In the edge computing, IoT devices that have high mobility are usually data producers, while in the cloud computing data is often generated by companies and enterprises, such as YouTube, Facebook, etc. Therefore, compared with the cloud computing, mobility support is essentially required in the edge computing because IoT devices are easy to move from one area to another area covered by edge nodes.

III. SECURITY AND PRIVACY MECHANISMS FOR OUTSOURCING DATA ANALYTICS

In this section, we briefly introduce main security and privacy mechanisms for outsourcing data analytics, which include secure data collection methods, secure data processing methods and secure data storage methods.

A. Secure Data Collection Methods

The first step in data analysis is to collect data from user devices. The collected data fundamentally affect quality and accuracy of data analytics. Therefore, we will first overview traditional security and privacy methods for secure data collection.

1) Authentication Mechanism: User authentication mechanism in outsourcing data analytics is a critical requirement to ensure the reliability of data source. It is able to validates the identity of user to guarantee that the user is legitimate to access cloud server or edge node [27]. Herein, we summarize a number of common authentication methods as below.

Password-based Authentication Methods (PAM). Password authentication is the simplest and the most convenient authentication mechanism [28]. In registration phase, each user sends a remote server his/her ID and password. The remote server maintains a password table which is used for storing user IDs and passwords. In authentication phase, the remote server verifies the legitimacy of the user with the password table. However, PAM suffers from the following drawbacks: i) passwords are easily leaked, since many users often set passwords to meaningful characters to prevent them from forgetting their passwords, such as their own or family members' birthdays, phone numbers, names, etc. ii) passwords are static, thus they are easily eavesdropped and stolen during transmission. iii) the password table is highly susceptible to be tampered by intruders [29].

Smart Card-based Authentication Methods (SCAM) [29] . Smart card is a kind of non-reproducible card, which contains data related to user identity. In registration phase, the system gives a user a smart card; In authentication phase, a special card reader reads the information carried by the smart card to verify the legitimacy of the user. The advantage of SCAM is that it does not need to maintain a password table. However, since the data read from the smart card are still static, the intruder may obtain the identity information of the user in the card reader by means of memory scanning.

Dynamic Password-based Authentication Method (DPAM). DPAM is a technique that allows a user's password to change continuously according to time or number of uses. Dynamic password is usually generated by dedicated hardware in user side, and the server uses the same algorithm to calculate the valid password. If only two passwords match, the authenticate succeeds. Dynamic password authentication adopts a one-time method to avoid the risk of stolen password. However, if the number of times the client and the server program are not synchronized, the authentication failure may occur.

Biometric-based Authentication Methods (BAM). Biometric authentication is a technology that takes advantage of the inherent biological or behavioral characteristics of an individual to verify his/her identity [30], such as fingerprint, voice, face, DNA, keystrokes, etc. Biometric authentication is more reliable than other authentication methods, since physical human characteristics and behavioral feature are very difficult to forge [27]. However, BAM may not be applicable in many application scenarios where human-beings are not involved in data collection. Moreover, BAM need long execution time and their security level is always constrained by time complexity, especially when high security level is needed.

2) Trust Management Mechanism: Although authentication can determine the authenticity of a device, the user does not guarantee that the service provider behaves well. Trust management mechanism refers to utilizing some effective approaches to achieve the measurement and computation of trust value in order to choose a more dependable service provider [31]. According to [32], [33], trust management mechanisms can be divided into two types: reputation-based and policy-based trust management.

Reputation-based Trust Management (RTM). RTM concerns a trust measurement method where utilizes numerical and computational mechanisms to obtain trust values. For example, in a social network, the trust value of each user is calculated by collecting and aggregating the reputation value obtained according to the opinion of others about it.

Policy-based Trust Management (PTM). PTM is an objective trust assessment method where logical rules and verifiable attributes are encoded in signed credentials to decide the access to data. The PTM method usually makes a binary decision based on whether a requestee allows access request [34]. PTW has less flexibility by reason of the binary nature of trust assessment.

B. Secure Data Processing Methods

In outsourcing data analytics, user devices can rely on computing and storage resources in cloud or edge servers to perform computationally intensive operations to obtain various services. Once data are outsourced to the cloud or edge servers, data owner loses the control of the data. And then the personal data of users can be revealed to the service provider or malicious intruders. Many works were conducted to solve this issue. Herein, we list main secure data processing methods as follows.

1) Homomorphic Encryption (HE): HE is a cryptographic technology that allows arbitrary data computation to be executed over ciphertexts and generates an encrypted computation result [35], [36] . When decrypted, this result is the same as the result of operations performed on the plaintext. HE can be divided into Full Homomorphic Encryption (FHE) and Partial Homomorphic Encryption (PHE). FHE (such as BGN encryption [37]) is designed to support mixed data computations over ciphertexts, but its computation overhead is much higher than PHE. PHE (such as Paillier encryption [38]) just supports one or two types of operations, thus it can only support restrained application scenarios. Therefore, appropriate algorithms should be selected according to the practical needs of application scenarios.

2) Differential Privacy (DP): DP is a privacy-preserving technique that provides strong and provable privacy guarantee for users by adding a random noise to user data [39]. In outsourcing data process, DP is mainly used to achieve privacy-preserving data aggregation which refers to a statistical data analysis, such as average, minimum, maximum, sum, and count over a given time period [40]. There are two ways to apply differential privacy in data aggregation. i) The first way is to add random noise to the result of the data aggregation, which can maximize the accuracy of aggregation to defend against differential attack and ensure [41]. ii) The second way is that each data owner adds a random noise to his/her data to prevent privacy leakage, and then the cloud or edge server aggregates these perturbed data. Although doing so brings some statistical error, when the amount of data is large enough,

it can still protect user privacy while completing data analysis [42], [43]. But the second way only supports summation aggregation.

3) Pseudonym Technology (PT): PT allows users to request the services offered by the cloud or edge server anonymously, using pseudonyms [44]. The pseudonym management is carried out by a centralized cloud server or lots of edge nodes. This technology is used to protect location privacy or identity privacy of users. Since edge nodes or cloud servers do not know the true identity of a user, user private data user private data cannot be associated with himself/herself. However, there are many methods of de-anonymization attack based on modeling and analysis of users' behaviors [45]–[47]. Therefore, the security of PT is lower than other methods.

C. Secure Data Storage Methods

To relieve the cost of storing data locally, some results of data processing are typically stored to edge nodes or cloud servers. In order to prevent the computational results from being tempered or thieved, these data are usually encrypted. However, encryption will hinder the operations (such as access, search, deduplication) of data results. Herein, we introduce some key mechanisms to protect data privacy without affecting the use of data.

1) Access Control Mechanism: During data storage, user needs to access computational results. Access control is a policy or procedure that only allows authorized user to access the data [48]. In what follows, we summarize several common access control models.

Role-based Access Control (RBAC). In RBRC, the access policies are related to roles and the authorization of users is achieved by assigning them corresponding roles. By mapping data owners to roles and roles to privileges on data objects [49], RBRC provides flexible access control and management. In cloud/edge computing, data are stored in cloud/edge data center due to the limited storage capacity of user devices, and the cloud/edge server acts as an administrator to manage data and access policies for data, which could lead to privacy disclosure. Therefore, RBRC must be combined with other security mechanisms to achieve more secure access control.

Proxy Re-Encryption (PRE). PRE-based access control model allows a proxy to transform a ciphertext encrypted with Alice's public key into one that can be decrypted by Bob's private key. PRE is designed to achieve secure data sharing in outsourcing data storage. A general PRE-based data sharing scheme is illustrated in Fig. 2. Each data owner may generate an arbitrary number of re-encryption keys based on his/her own private key and the recipients' public keys and then an access control list with re-encryption key for each recipient are upload to a semi-trust proxy server deployed in the cloud/edge server. When someone wants to access data stored in cloud server, data will be re-encrypted and send to the recipient by the proxy server.

Attribute-based Access Control (ABAC). Attribute-based encryption (ABE) is a cryptographic technology where the secret key of a data owner and the ciphertext are relied on attributes of recipients. Based on it, ABAC can offer fine-grained access

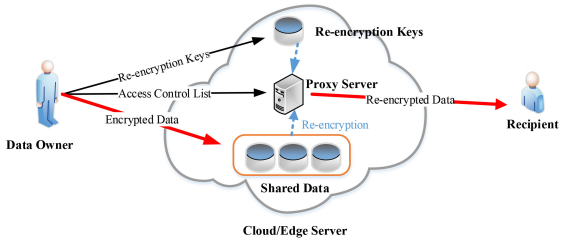

Fig. 2. PRE-based secure data sharing scheme

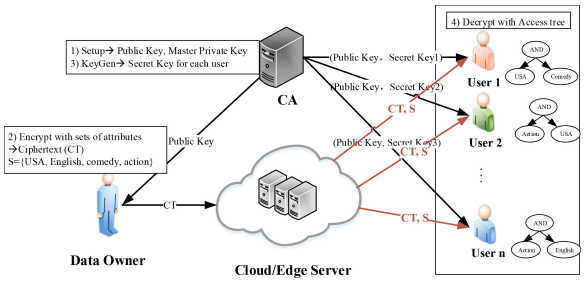

Fig. 3. KP-ABE access control scheme

control, since it can grant the data owner the ability to set the access policy in a very fine-grained way to preserve the private data of data owner. The recipients are able to encrypt data only when his/her attributes match the specified access policy. ABE includes two main types: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE access control, as displayed in Fig. 3, the ciphertexts are labeled with a set of attributes and private key is associated with access structures that can control which ciphertext a user is able to decrypt. CP-ABE access control is shown in Fig. 4. Private key is labeled with a set of attributes and ciphertext is associated with access structures that control which user is able to decrypt the ciphertext. However, decryption of ABAC requires to operate multiple bilinear pairings, which incurs high computational overhead.

2) Searchable Encryption: In outsourcing computing paradigm, data is typically stored in ciphertext onin the cloud/edge server, which disrupts search functionality. Song *et al.* [50] first proposed searchable encryption whichthat not only can achieve data encryption but also support keyword search over ciphertext. The two main branches of searchable encryption are symmetric searchable encryption (SSE) and

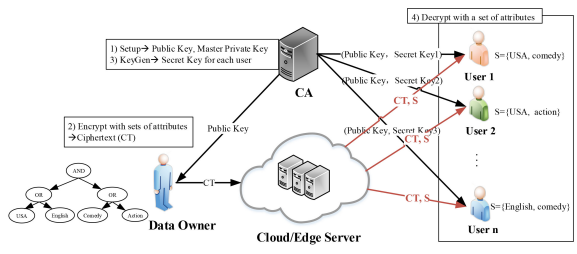

Fig. 4. CP-ABE access control scheme

asymmetric searchable encryption (ASE) [51]. In SSE scheme, data owner uses his/her symmetric key to encrypt data index and create search trapdoors. ASE enables many users who hold the public key to encrypt data but only allows the data owner to create search trapdoors. And then the trapdoors are sent to the cloud service, which executes the search algorithm and returns search results to the data users. Search can also be based on certain rules (e.g. returning the most matching of the first n related results).

IV. Security Threats and Requirements of Data Analytics in Edge Computing

This section summarizes the fundamental requirements of data analytics in edge computing by analyzing potential security threats with regards to IoT devices, communication networks and edge-cloud devices.

A. Security Threats of Data Analytics in Edge Computing

Edge computing pioneers a new computing model that brings great convenience to solve network congestion and latency. However, with intellectualization of local edge devices, it also encounters various security and privacy challenges. As an extension of cloud computing, the edge computing not only inherits the secure threats from the cloud computing, but also introduces new security risks due to its own features. In this subsection, we will summarize the potential security and privacy threats existing in data analytics from three aspects including IoT devices, communication networks, and edge-cloud devices.

1) Security Threats on IoT Devices: The IoT devices are very important components in the framework of edge computing. On one hand, they generate data and submit raw data to edge nodes or cloud servers. On the other hand, they can also participate in the provision of services. Therefore, it is vital to ensure the security of IoT devices and the reliability of their collected data. In what follows, we list some common security attacks on IoT devices.

Physical Attack: In the edge computing, most IoT devices are deployed in unattended or outdoor environments, such as tunnels, subways, factories, etc. Therefore, they are highly susceptible to physical attacks [52], including nature disaster, energy supply shortage, device damage, etc.

Injection of Information Attack [4]: An adversary can control and reprogram a device to distribute fake information. For example, malicious users provide fake data to a crowdsourcing service.

Service Manipulation Attack [4]: A device might participate in the provision of services and manipulate the outcome of the service. For example, a cluster of devices can act as an edge node to control a service [53].

2) Security Threats on Networks: As aforementioned in Section II, most IoT devices interact with edge nodes or cloud servers through wireless communications. Hence, the data collection in edge computing also faces various network security challenges related to wireless communications. In what follows, we summarize security attacks on wireless networks.

Man-in-the-Middle Attack: Edge nodes like gateways may be compromised and private communications could be intercepted by a fake node once an attacker exerts control over normal edge nodes [7].

Sybil Attack: A Sybil attacker claims a large number of client identities or impersonate some legal nodes to control or compromise the whole edge computing framework. For example, in edge computing-based crowdsensing, Sybil attackers could send incorrect reports to an edge node, which will aggravate the influence of false data injection and impact the accuracy of data analytics.

Sinkhole Attack [54]: A rogues router performs routing and attracts surrounding nodes using unfaithful routing information. Then the router (i.e., the attacker) may perform selective forwarding or alters the data passing through it.

Eavesdropping Attack: Due to the broadcast nature in wireless communications, the edge computing is vulnerable to eavesdropping attacks. An attacker might monitor wireless channels to snatch data packets to obtain private communication contents. This type of attack is difficult to be detected, thus some encryption measures should be implemented to guarantee data confidentiality.

Jamming Channel Attack: An attacker purposely sends a huge number of forged messages to exhaust communication channels or computing resources, which makes legal users unable to communication with each other.

Forgery Attack: Forgery is a common attack in wireless communications. Malicious attackers compromise the whole system by forging valid messages and configuration files. Moreover, these fake messages might consume network bandwidth and storage resources of edge nodes, thus further affect the accuracy of data analytics.

Tampering Attack [55]: A tempering adversary can maliciously delay, drop or even alter transmitting data to undermine edge computing and degrade the efficiency of edge nodes. Since the condition of wireless channels and user mobility may also lead to transmission delay or failure, it is difficult to detect tampering behaviors.

3) Security Threats on Edge-Cloud Devices: Edge devices and cloud servers are the core part of edge computing. They host virtualized servers and provide outsourced data computation and data storage services for user devices. Thus, external attackers normally try to disrupt the services provided by the edge devices and cloud servers in various ways. Herein, we list several security attacks on edge-cloud devices.

Distributed Denial of Service (DDoS) Attack: DDoS attack is a severe attack in edge computing. Adversaries deliberately utilize the drawback of network protocol or directly run out of the resources of targeted entity, and make targeted edge nodes, cloud servers, or network fail to provide services or access to resources. There are generally three kinds of DDoS attacks. First, many adversaries send a mass of data packets to jam the bandwidth of a server to make its channel disabled. The second case is to consume CPU memory resources by sending specific request packets, e.g., TCP/IP request packets. Third, when connection is built, adversary sends a good deal of data packets to consume service resources in edge devices or cloud servers.

Insider Attack: The data theft attack raised by an internal adversary is one of serious attacks in cloud computing and edge computing. Malicious insiders usually abuse their privileges and utilize their knowledge to steal user private data [14].

SQL Injection: In an SQL injection attack, an attack inserts a malicious piece of code into an SQL code. Thus, the malicious code is erroneously executed in database backend.

Privacy Leakage: Edge devices are located in the edge of the network near data sources. Compared with the cloud computing where a data center is located in the core network, the edge node devices can collect more high-value and sensitive data from mobile users. Four types of privacy leakage could occur in the edge computing, including data privacy, location privacy, usage privacy, and identity privacy.

- *Data privacy:* In the edge computing, edge nodes usually collect user data from sensors and end devices for analysis or computing. The edge nodes could be curious even though honest. They might snoop on user information driven by pecuniary benefits. However, some of the collected data is sensitive, such as the drug purchase records of users, the health status of a patient. Thus, data privacy could be leaked at the edge nodes.
- *Location privacy:* Two reasons account for user location privacy leakage. On the one hand, an edge node requires to be constantly aware of users in its own coverage area and the users usually choose to access the nearest edge node, which results in disclosing user location information. On the other hand, location privacy might be exposed from transmission. For example, if a user device usually delegates its task to the nearest edge node, it is possible to reveal its location information to an eavesdropper even without knowing the specific content of transmission. Moreover, if this user device is moving from its original region to another region and access the services of multiple edge nodes, the eavesdropper might know its trajectory information.
- *Usage privacy [13]:* Usage privacy is extremely crucial in edge computing. It primarily refers to the usage patterns with which a client leverages the services provided by edge nodes. For instance, in smart grid, an eavesdropper or a curious edge node can acquire a lot of information of a family. It can analyze the readings of a smart meter to infer family private information, such as when the family is or is not at home, and when the family members go to sleep. Obviously, resident privacy is leaked.
- *Identity privacy:* User identity links to a special user. The identity information includes identity number, name, address, telephone number, public-key certificate, etc. When a client accesses services offered by an edge node, it would be susceptible to identity privacy leakage.

B. Requirements of Data Analytics Based on Edge Computing

Since there exists a growing number of security threats and privacy leakage problems in edge computing, it is essential to deploy some security mechanisms to resist those external and internal threats as mentioned in Section III.A. In this subsection, we propose a number of essential requirements that should be satisfied in edge computing in the services of data analytics. The detailed requirements are classified into two parts: security and privacy requirements and performance requirements, as summarized below.

1) Security and Privacy Requirements: First, security mechanisms should consider the following security requirements in order to enhance the security of services provided by edge nodes.

Authenticity (Au): In order to confirm the identity of involved edge nodes and edge devices, authenticity should be guaranteed before participating in edge network for data analysis. Mutual authentication enables involved edge and user devices to authenticate each other, which is an effective method to provide authenticity.

Trustworthiness (Tu): Besides authenticity of edge nodes and user devices, the trustworthiness of networked devices is of great importance. Authenticity helps establishing an initial and secure relationship between user devices and edge nodes, but it cannot guarantee the honesty of their subsequent actions. They may act dishonestly and even be compromised by attackers. Trust management becomes a good manner to monitor the nodes and devices, and figure out their trustworthiness.

Confidentiality (Cn): Due to the eavesdropping attack, data confidentiality also needs to be satisfied, which is important for data security and user privacy.

Integrity (I): Data integrity should be ensured to prevent original data from being tampered by attackers during transmission or even by the semi-trusted edge nodes.

Location Privacy (LP): As mentioned in Section III.A, user location may be disclosed when enjoying the services provided by edge computing, but it is extremely sensitive in some scenarios. For example, in a fog-based parking navigation application [56], a moving vehicle uploads the videos or photos of vacant parking spaces to nearby fog nodes. However, location privacy of drivers is disclosed due to location awareness of fog nodes. If location information of the drivers is not protected, no drivers would like to participate in such a parking navigation system. Hence, location privacy should be preserved.

Usage Privacy (UP): Usage patterns indicate the habits of users to consume the services from edge nodes. Once the usage patterns of users are disclosed, an adversary may know the details of the routine life and activities of the users. Hence, a secure solution is highly expected to protect the usage privacy of users.

Identity Privacy (IP): Identity privacy plays an important role in ensuring user privacy. An identity can be easily linked to a user, and hence it causes big threats to user private information, such as name, address, etc. If identity privacy is not guaranteed, users would be unwilling to access services provided by the edge computing. Anonymity may be a countermeasure to protect user identity.

Traceability (Ta): Since edge computing introduces an intermediate edge node layer, users lose the control of their raw data. Therefore, it is likely that incorrect results are gained due to improper operation of one edge node or faked data updated from users. Furthermore, due to the mobility and

decentralization of edge nodes, the errors caused by one edge node might spread to other computation tasks and lead to more incorrect final results. In order to improve the correctness of data analysis and reduce the influence of faked data and malicious users, it is of great significance to realize data provenance for tracking back to invalid data inputs.

2) Performance Requirements: Different from cloud computing, edge nodes are highly distributed and have lower computation capabilities and resources. Hence, performance should also be seriously considered when judging the quality of data analysis schemes based on edge computing. Generally, the following criteria related to performance are considered.

Correctness (Cr): As mentioned above, there exist some attacks on data analytics, which may lead to incorrect analytical results and serious consequences. Hence, correctness of data analysis is an essential quality attribute of performance.

Scalability (S): Scalability means that edge nodes can run normally when adding a new device or removing a device. The edge nodes and user devices join or leave frequently in an edge computing framework. Hence, scalability is highly expected to be supported.

Mobility (Mo): As users might pass through several coverage areas of edge nodes in a high speed, mobility becomes an important factor to consider for securing data analytics in edge computing. For example, edge nodes can be deployed in self-driving cars to monitor real-time traffic conditions by analyzing data collected from various sensors.

Efficiency (E): As the resources of user devices and edge nodes are much less than those of cloud servers, efficiency is paid more attention in the edge computing than the cloud computing. Moreover, the edge computing aims to gain the advantage of low latency over the cloud computing. Usually, we need to consider communication and computation cost in the judgement of the performance of a scheme. However, the communication cost of a scheme refers to the size of transmitted packets and the number of communication interactions, which is not easy to uniformly measure. Therefore, we mainly focus on efficiency in our review and show the communication overhead of all involved entities in edge computing. As the fixed times of basic computations (such as multiplication and addition) are very efficient, we merely consider the complex operations (such as bilinear pairing, exponentiation operation, modular addition operation, and modular multiplication operation) in our review and analysis.

V. Secure Data Analytics in Edge Computing

In this section, we review and discuss recent advance of data analytics schemes in edge computing by employing our proposed requirements in Section IV.B as a measure to comment their pros and cons. In this paper, we focus on the related works in fog computing and edge computing with regards to secure data collection, secure data processing and secure data storage. Before going into the details, we first list the notations used in computational overhead evaluation in Table III.

TABLE III
Notation Description

Symbol	Description
p_u	The number of participants in crowdsensing application
p_f	The number of edge nodes involved in crowdsensing application
m	The number of the dimensions of a sensing vector
l	The bit length of m
g	The maximum links per node in [53]
$u^{'}$	The number of users in [53]
A	The simulation area in [53]
u	The number of aggregated users per edge node
f	The number of aggregated edge nodes per cloud server
t	The number of the attribute universe in attribute-based encryption (ABE)
$t^{'}$	The number of slots in [57]
a	The number of decrypted attributes in ABE
$a^{'}$	The number of newly decrypted attributes in ABE after preprocessing at an edge node
p	The number of attribute authorities in [58]
n	The number of edges of a proximity polygon area that is given by a requester
x	The accuracy of proximity detection
n_1	The number of blocks in an outsourced file

A. Secure Data Collection in Edge Computing

As the first step of data analytics, data collection seriously influences the performance of data analytics, especially its accuracy [55]. Hence, we first review the literature on secure and privacy-preserving data collection.

1) Authentication in Edge Computing: Fake data injection attack seriously affects the reliability of data sources, while authentication can help verify the identity of each entity involved in edge computing. Hence, authentication can be applied to prevent malicious data injection and improve the quality of collected data. In this part, we review existing authentication schemes in edge computing. When discussing scheme efficiency, we only consider the computational overhead of authentication.

Ibrahim *et al.* [59] proposed a mutual authentication scheme in a hierarchical fog framework, which allows any user device to authenticate with any fog node in a fog area. In details, each fog user holds only one long-lived master secret key that can generate the session key between any fog server and itself. This design decreases the storage overhead in user device side. During authentication, a session key is securely shared by both participants, which provides data confidentiality. However, the identity of the user device is transferred in plaintext, which makes it easy for adversary to know which user and which fog node are communicating and discloses identify privacy of users. Moreover, mutual authentication can only be provided in one fog area. Hence, it results in moderate scalability and mobility. The computational overhead of this scheme includes one hash and two symmetric encryption operations in user side and two symmetric encryption operations in edge node side.

In order to solve the issue of limited user mobility support in [59], Amor *et al.* [60] designed an anonymous mutual authentication scheme by further revising the work in [59]. Identity privacy is preserved by pseudonym-based cryptography and high mobility is supported by considering three cases, namely,

authentication of a single fog node, intra-fog authentication, and inter-fog authentication. Different from [59], the session key between a fog user and a fog node is generated by using bilinear pairing, which provides data confidentiality. However, both [59] and [60] just consider the situation where a new fog node joins the fog layer but does not consider the departure of old fog nodes, thus its support on mobility is still constrained. Moreover, they need the fog nodes in one fog to store IDs of all fog users, which introduces high storage overhead in fog node side. Besides, data integrity is not ensured in these works. With regards to computational overhead, we analyze the most complicated case, inter-fog authentication. This scheme [60] includes one elliptic curve point multiplication operation, one hash operation and one bilinear pairing in user side, one elliptic curve point multiplication operation, one bilinear pairing in edge nodes, and two elliptic curve point multiplication operations in cloud side.

Currently, biometric authentication is becoming popular [30], as it automatically recognizes and verifies the identity of a living person. However, biometric identification takes relatively long execution time. To improve the efficiency of authentication, some researchers tended to combine biometric authentication with edge computing. Hu *et al.* [61] explored a face identification and resolution authentication framework base on fog computing between IoT devices and the cloud, where some complex operations of face feature extraction are offloaded to fog nodes. Moreover, it realizes data confidentiality, data integrity and mutual authentication among cloud servers by leveraging multiple cryptography techniques. However, this work does not implement secure interactions between user devices and fog nodes, since it reveals face information to the fog node. Furthermore, scalability and mobility were not considered as well. In addition, several cloud servers need to frequently interact with each other, which incurs high communication overhead. Regarding computational overhead, the edge node needs to do one hash operation, two symmetric encryption operations, two modular exponentiation operations and three elliptic curve point multiplication operations, while the cloud undertakes ten modular exponentiation operations, 15 elliptic curve point multiplication operations, 12 symmetric encryption operations and nine hash operations.

2) Secure Data Transmission: Owing to the existence of security and privacy issues in the whole edge networks as mentioned in Section III.A, data confidentiality and integrity should be ensured during data transmission. Hence, it is necessary to explore a secure data transmission scheme to prevent data from being tampered and stolen. In this part, we review secure data transmission schemes in edge computing.

With the purpose of achieving the secure communications between fog nodes and cloud server, Alrawais *et al.* [62] designed an encrypted key exchange protocol by combining ciphertext-policy attribute-based encryption (CP-ABE) with digital signature to offer data confidentiality and data integrity. Meanwhile, the system prevents an active attacker from learning or changing the transmitted data. The computational overhead of this protocol executes one hash operation,$(2a+1)$ bilinear pairings, a modular mutiplication operations and a hash operations in fog node side, and $(4a+1)$ modular exponentiation operations and $2a$ hash operations in cloud side. Due to high computational overhead, this protocol cannot be used in delay-sensitive scenarios. Moreover, it neglects scalability and mobility of user devices and fog nodes in design.

Though the previous work ensures secure communication, it ignores efficiency. Signcrytion technology was designed as an efficient public-key primitive to simultaneously perform both digital signature and data encryption [63]. Basudan *et al.* [64] designed a certificateless aggregate signcryption (CLASC) protocol for road surface condition monitoring based on vehicular crowdsensing in fog framework, which achieved mutual authentication, data condfidentiality and integrity. Moreover, it introduces a pseudo identity generated from a real identity for vehicular devices to communicate with fog nodes, thus this protocol preserves identify privacy. In addition, it combines mix-zone technique with pseudonym technique to realize location privacy protection. Mobility was supported in this work, but scalability was missed. With regards to computational overhead, the vehicular device undertakes three elliptic curve point multiplication, one elliptic curve point addition, and three hash operations. The computational overhead at the fog node includes $6u$ elliptic curve point multiplications, u elliptic curve point addition, $3u$ hash operations and $3u$ bilinear pairings. In addition, fog node first aggregates all messages about the same road event from different vehicles and performs batch signature verfication, thus the computational overhead of this protocol is less than [65], [66].

However , the protocol in [64] needs to do many expensive bilinear pairings, the computational overhead of this protocol is still a bit high. Moreover, this protocol has been demonstrated to be vulnerable to forgery attack by Chen *et al.* [67], where an attacker may forge the signature of arbitrary data. To solve these problems, Chen *et al.* [67] designed a light-weight and anonymous aggregate signcryption for a fog-enabled vehicle-to-infrastructure scenario, which can guarantee data confidentiality and integrity. The advantage of this scheme is that the full secret key of mobile sensor cannot be obtained in any case, thus unforgeability of signature is ensured. Moreover, the real identity of each user device cannot be retrieved from the road condition report generated by each mobile sensor about road event to protect the identity privacy of users. However, moblity and scalability were not considered. In this scheme, each mobile sensor needs to do three elliptic curve point multiplication, one elliptic curve point addition, five hash operations, five modular mutiplication operations and four modular addition operations. Each fog node aggregates signatures of all mobile sensors in its coverage area to do batch verfication, which needs $2u$ elliptic curve point multiplications, $5u$ modular addition operations, $2u$ hash operations and $3u$ modular mutiplication operations.

3) Trust Management in Edge Computing: Authentication can only check the authenticity of entities before connection is established, but it is difficult to ensure whether they behave in a satisfactory way during service. User devices and edge nodes frequently join or leave the system of edge computing and they need to continually interact with unfamiliar objects. Hence, trust management becomes an especially crucial manner in

edge computing to determine whether to set up cooperation with the unfamiliar objects. For example, in a driverless automobile application based on edge computing, the users prefer to choose a trusted edge node that offers reliable services. Such a decision can reduce occurrence of traffic accidents and protect user personal safety. In this part, we overview trust management schemes in edge computing.

Su *et al.* [68] proposed a policy-based end-to-end trustworthiness governance scheme where fog nodes could serve for user devices only when the security attributes of fog nodes satisfy the demands of processed data. Since this work can support multi-organizational trust evaluation, it guarantees the trustworthiness of fog servers. However, scalability and mobility of users were not considered in this scheme design. Since no specific algorithm was given, its computational overhead is hard to judge.

Furthermore, Sharma *et al.* [53] developed an entropy-based trust relaying and privacy preservation system by using edge-crowd integration in social Internet of things (S-IoT), which provides trustworthy interactions among user devices. This scheme utilizes available IoT devices as mini-edge servers to release the deployment of edge servers near user-site and reduces the cost and complexity of system network. Moreover, for leveraging fission computing, it supports distributed trust management without a centralized reputation system and can detect fake sources. Also, a user movement model was proposed to support mobility. However, it is specifically designed for an S-IoT environment, which limits its applicability in other scenarios. The computational overhead for each user includes ignorable addition operations and $gh + 4g + 2n + 4A$ multiplication operations.

Ma *et al.* [69] proposed two privacy-preserving reputation management mechanisms for edge computing-based mobile crowdsensing to deal with malicious participants. The first scheme is more efficient than the second one, but it discloses the deviations of each participant. The second scheme updates the reputation values by utilizing the rank of deviations, but it increases the computational overhead. Since the nearest edge node aggregates the encrypted sensing data of all participants by applying somewhat-homomorphic encryption, data confidentiality is guaranteed. However, the location privacy of participants is revealed to edge nodes. Furthermore, each user device (both participant and inquirer) can join edge network but the departure of user is not considered, thus its scalability is moderate. Moreover, mobility and traceability are missed as well. In addition, this proposed scheme just explores the trustworthiness of participants, but neglects the trustworthiness of edge nodes. Regarding computational overhead, we herein just analyze the second scheme since it is more secure, which includes three modular multiplication operations and three modular addition operations in participant side, two modular multiplication operations and one modular addition operation in querier, $\frac{p_u}{2}(17p_u + 3p_u l - 9 - 3l)$ modular multiplication operations, l modular exponentiation operarions and $\frac{p_u}{2}(14p_u + 3lp_u - 6p_u - 3l)$ modular addition operations in edge node side and $2p_u$ exponentiation operations in cloud side.

B. Secure Data Processing in Edge Computing

Data analytics/processing is a vital part of various services based on edge computing. It helps in digging out significant information and further improving related services. In this subsection, we review the existing works about secure data processing and analytics in edge computing by classifying them into two categories: privacy-preserving data computation and privacy-preserving data aggregation.

1) Privacy-preserving Data Computation in Edge Computing: Edge nodes and cloud servers provide the capability to deal with complex operations for resource-constrained devices. However, once data is outsourced to the edge nodes or the cloud servers, data owners will lose full control on their data. Meanwhile, as honest-but-curious entities, the cloud servers and the edge nodes may disclose user personal data and invade user privacy. Even worse, the scope of privacy breach could be expanded owing to the mobility of users. Therefore, it is mostly important to achieve large-scale data computation and preserve user privacy simultaneously.

a) Data Computation Based on Homomorphic Encryption Technology: Homomorphic encryption is an efficient technology that enables arbitrary data computation to be executed over ciphertexts [35], [36]. It is widely used to realize privacy-preserving outsourced data computation. Liu *et al.* [70] designed a hybrid clinical decision system in fog-cloud network to monitor patients' physical conditions in real time by combining data mining with Paillier homomorphic encryption. This system achieves lightweight and real-time data processing in fog nodes, while high accuracy disease decision algorithms are implemented in the cloud. The advantage of this work is that it supports various computations over encrypted non-integer data. However, authenticity, data integrity, identity privacy, scalability and mobility are not considered. We take one data packet for each user as an example for efficiency analysis. The computation at user side contains one modular exponentiation operation, one modular addition operation and one modular multiplication operation. In edge node side, a real-time process algorithm includes five modular exponentiation operations, two modular addition operations and six modular multiplication operations.

Similarly, Huo *et al.* [71] utilized Paillier homomorphic cryptosystem and decision-tree theory to implement a location difference-based proximity detection (LoDPD) system in fog computing. This system protects location privacy of users and ensures data confidentiality. The advantage of this scheme is that its communication cost and CPU cost are lower than traditional private proximity detection (PPD) methods. However, the friend information of users is sent to a local fog node in plaintext, which results in leaking private data. Moreover, it does not consider scalability, mobility, and authenticity of users. In terms of computational overhead, we assume that there is only one friend near a requester. In this case, the requester needs to perform two modular exponentiation operations and one modular multiplication operation. Every friend of the requestor undertakes two modular exponentiation operations and four modular multiplication operations. And in fog node side, it includes $4n(x + 4)$ modular exponentiation

operations and $2n(x+3)$ modular multiplication operations.

b) Data Computation Based on Pseudonym Technology: Apart from homomorphic encryption, pseudonym technology is also deemed as an appropriate technique to protect personal privacy during data computation [44]. Since edge nodes or cloud servers do not know the true identity of a user, user private data cannot be associated with himself. Kang *et al.* [72] took advantages of edge resources to develop a fog-assisted pseudonym management scheme. It protects location privacy and identify privacy for vehicles. The proposed scheme achieves timely pseudonym distribution and reduces management and communication overhead by deploying local authority in fog nodes. Moreover, the authors designed a context-aware pseudonym changing game to dynamically change pseudonym with context awareness. By leveraging digital signature and public key encryption during data transmission, the authenticity of all entities, data integrity and data confidentiality are fulfilled. Furthermore, since this scheme also offers cross-region pseudonym requesting and changing services, mobility is supported. However, edge nodes and cloud server are regarded as honest and they are responsible for pseudonym generation. Scheme scalability is missed. Since the encryption and signature algorithms were not detailed, the computational overhead of the scheme cannot be evaluated.

Also, Wang *et al.* [73] leveraged randomizing anonymous credentials to achieve a privacy-preserving crowdsourcing-based navigation scheme in a fog network. In this scheme, fog nodes generate and publish crowdsensing tasks, and then compute an optimal route with the traffic information collected by vehicles. Combining multiple cryptographic technology, like group signatures, Advanced Encryption Standard (AES) and Elgamal encryption algorithm, the scheme fulfills the security requirements of authenticity, confidentiality, identity privacy and location privacy. However, data integrity is missed. Moreover, the correctness of calculation is not validated and traceability, scalability and mobility are not considered. The computational overhead of this scheme includes four modular exponentiation operations, one modular multiplication operation, one symmetric encryption operation, one bilinear pairing and one hash operation in querier side, two modular exponentiation operations, one modular multiplication operation, one symmetric encryption operation, one bilinear pairing, three hash operations and one modular addition operation in participant side, 12 modular exponentiation operations, seven modular multiplication operations, two symmetric encryption operations, six bilinear pairings, two hash operations in edge node side.

c) Data Computation Based on Other Technology: Chaff-based technology that allows a user device to generate additional tasks to hide real tasks can also be used to implement privacy-preserving data analysis. He *et al.* [57] proposed a trajectory privacy protection mechanism by utilizing chaff services to safely offload tasks to distrusted edge nodes. In this mechanism, a user generates many additional chaffs to confuse eavesdroppers so that the eavesdroppers cannot detect the trajectory of the user. The proposed algorithm assumes that the eavesdroppers detect the user's location information by maximizing likelihood detection, and then designs a series of chaff control strategies to minimize tracking accuracy, e.g., impersonation strategy, maximum likelihood strategy, optimal offline strategy, and optimal online strategy. Each user designs the trajectories for chaffs through $t^{'}$ multiplication operations, which can be ignored in terms of computational overhead. However, each chaff consumes the computing resources of edge nodes, thus this scheme introduces high computational overhead in edge node side.

There are also other secure data calculation methods which were only designed to protect some specific privacy. For example, Yang *et al.* [74] first explored the new definition of secure positioning protocol in fog computing to solve the issue of location privacy leakage in a bounded retrieval model [75]. In this work, a prover proves that it is in a certain region in order to hide its exact position by considering two scenarios, one-dimension scenario and three-dimension scenario. However, the disadvantage of this work is the authors assume that the cost of reading bits and performing computation at a device is zero, which is not practical. This scheme takes advantage of the time interval between the two verifiers and the prover to determine the location area of the prover, which just needs several addition operations. Therefore, its computational cost can be ignored.

Dang and Hoang [76] designed a data protection model for edge computing to guarantee data security and support mobility. It consists of a Region-Based Trust-Aware (RBTA) model to achieve trust establishment between two regions, Fog-based Privacy-aware Role Based Access Control (FPRBAC) and a mobility management service. This scheme fulfills the requirements on trustworthiness, mobility and scalability. However, data confidentiality and data integrity were not considered. In addition, location privacy and identity privacy were not protected. Since no specific algorithm was described in the scheme, its computational overhead cannot be evaluated.

2) Privacy-preserving Data Aggregation in Edge Computing: Besides aforementioned complex analysis, data aggregation is one of the simplest but important computations for providing vital services in edge computing. For example, in crowdsensing-based traffic monitoring applications, edge nodes (e.g. road units) collect and preprocess traffic flow data from multiple vehicles and then send them to a cloud server. Owing to privacy concern, homomorphic encryption [77]–[79] and differential privacy [42], [80] were applied to achieve privacy-preserving data aggregation in edge computing.

Lu *et al.* [77] employed homomorphic encryption to develop a lightweight data aggregation scheme for heterogeneous IoT devices in fog computing, which provides data confidentiality and data integrity. However, identity privacy, traceability, scalability and mobility were not considered. In terms of computational overhead, each user device needs to do one symmetric encryption, one hash operation, five modular multiplication operations and three modular addition operations. Each edge node performs one symmetric encryption, five hash operations, one modular exponentiation operation, and $u+1$ modular multiplication operations. A cloud server undertakes one hash operation, one modular exponentiation operation, two modular multiplication operations and one modular addition operation.

Lyu *et al.* [42] also presented a privacy-preserving fog-

assisted data aggregation scheme for smart grid by using differential privacy and secret sharing. Specifically, this scheme utilizes Gaussian distribution noise to perturb private data to ensure differential privacy of aggregate statistic. Moreover, two-layer aggregation can alleviate privacy leakage and maintain data utility. This work uses public-key cryptography to realize authentication and considers the join and leave of nodes in order to provide high scalability. However, since data aggregation service is always offered by the nearest fog node, the location privacy of users might be disclosed to the fog node. In addition, data integrity, identity privacy, traceability, mobility were not considered. But this scheme is efficient and suitable for resource-constrained devices. It only executes one modular addition operation in user side, $u+1$ modular addition operations in fog node side and f modular addition operations in cloud side.

The above two data aggregation schemes are fault-tolerance. But identity privacy protection was not considered. In order to tackle this problem, Wang *et al.* [78] introduced an anonymous data aggregation scheme in a fog environment by employing pseudonyms technology. The cloud authenticates with fog nodes and user devices in registration phase, which guarantees the authenticity of all entities engaged in the fog computing. This scheme not only protects identity privacy of terminal users but also guarantees data confidentiality via homomorphic encryption. In this scheme, the fog node and cloud verify received messages, which ensures data integrity. Besides, revocation of terminal devices and fog nodes was considered. However, mobility was left out of consideration. This scheme executes three exponentiation operations, two hash operations, one modular exponentiation operation, two modular multiplication operations and bilinear pairings in user side, one hash operation, two bilinear pairings, two modular multiplication operations and $2(u-1)$ elliptic curve point addition operations in fog node side, one hash operation, one modular exponentiation operation and two bilinear pairings in cloud side. Different from [77], both [78] and [42] mainly perform single-modality data aggregation and do not consider the heterogeneity of data in edge computing.

For the purpose of preserving both data privacy and identity privacy, Guan *et al.* [79] combined pseudonym certificate with Paillier homomorphic encryption to achieve secure data aggregation in a fog-enhanced IoT environment. In this scheme, each fog area owns a local certification authority (LCA) and a trusted certificate authority (TCA). It works with user devices to generate and update the pseudonym certificate, which prevents certificate forgery. Furthermore, all entities can verify data integrity with digest during data transmissions. In addition, both pseudonym certificate update and revocation were considered, thus scalability can be satisfied to some extent. However, user mobility among fogs and location privacy were missed. As for computational complexity, each user device undertakes eight modular exponentiation operations, two hash operations and one modular multiplication operation for pseudonym generation and secure data processing. For the generation of each user's pseudonyms, LCA needs to perform six modular exponentiation operations, one hash operation and two modular multiplication operations and TCA does two modular exponentiation operations. Besides, each fog node takes $(u+3)$ hash operations and $(u+2)$ modular multiplication operations for secure data aggregation. Finally, the aggregated result uploads to the cloud server and is verified by the cloud, which needs one modular exponentiation operation, one hash operation and one modular multiplication operation. All the above-mentioned data aggregation schemes do not support traceability and cannot verify the correctness of aggregated results.

C. Secure Data Storage in Edge Computing

Due to the limited storage capacity of user devices, some results of data processing are typically stored in edge nodes or cloud servers. Secure data storage is also important to prevent computational results from being tempered or thieved. In this subsection, we overview the existing efforts on security and privacy in data storage in edge computing. The review is classified into three parts: data access control, secure data search, and secure data deduplication.

1) Data Access Control in Edge Computing: After data processing, user needs to access computational results stored in edge nodes or cloud servers [81]. If no proper security mechanism is deployed, any unauthorized user can arbitrarily access resources of other users, which obviously intrudes personal privacy. Access control is an efficient manner to authorize user devices to access distinctive resources in edge computing [20]. Therefore, it is crucial to explore access control in edge computing. In the following part, we review the existing access control schemes in edge computing. Regarding scheme efficiency, we just analyze the computational overhead of user decryption in access control.

a) Access Control Based on Attribute-Based Encryption Technology: Attribute-Based Encryption (ABE) is an efficient technique to achieve secure and fine-grained access control [82], as it not only protects private data, but also grants data owners the ability to directly set access policies. Zuo *et al.* [83] proposed an attributed-based encryption with outsourced decryption (OD-ABE) scheme in a fog computing environment. It achieves chosen ciphertext attack (CCA) security. There are two ciphertexts in this algorithm: one is from data owner and the other is from fog node; the latter ciphertexts are decrypted using a shorter private key, which reduces the computational overhead of decryption and saves the storage of IoT devices. Moreover, through CHK and FO transformation, a decryptor can check the validity of data, thus the integrity of data is ensured. However, since the shorter private key is generated by a data owner, the outsourced data cannot be decrypted except the data owner. Moreover, traceability, scalability and mobility were not considered. The computational overhead of decrypting the ciphertexts is only two modular exponentiation operations in user side. Its computational overhead is lower than traditional cloud computing.

Fan *et al.* [58] also designed a CP-ABE-based verifiable multi-authority outsourced access control scheme in a fog-cloud network. This scheme offloads most encryption and decryption computations to fog devices in order to reduce computational overhead of user side. Computation results can

be verified by data owners. Meanwhile, user involvement and revocation were considered, thus scalability was considered. Besides, attribute revocation are also supported to address revocation issues. We can see that data confidentiality and correctness of computation were realized. However, the disadvantage of this scheme is that each attribute needs an attribute authority to manage it, thus a great number of devices should be deployed. Moreover, identity privacy is disclosed during service access. Mobility, data integrity, user authenticity as well as traceability were not considered. With regards to computational overhead, this scheme executes one modular exponentiation operation and one modular multiplication operation in user side and $(3a + 1)p$ bilinear pairings in edge node side.

However, encryption and decryption are performed in user devices [58] and [83]. Due to limited resources of end devices, the schemes bring huge computing cost to user devices with the increase of attributes. By taking advantage of fog node's computing power, Zhang *et al.* [84] proposed an improved CP-ABE-based access control scheme, which ensures data confidentiality. This work offloads the generation of access control structure in encryption phase and bilinear pairing operations in decryption phase to fog nodes to reduce the computational burden of user equipment. Moreover, an efficient attribute update method without sending re-encrypting messages was proposed in this scheme. However, the introduction of edge nodes brings more communication overhead. Besides, data integrity, location privacy, identity privacy, mobility and scalability were not considered. Concerning computational cost, there are one bilinear pairing operation and two modular multiplication operations in user side and $(a+2)$ bilinear pairing operations and $(2a+2)$ modular multiplication operations in edge node.

In fact, the ABE-based access control schemes can raise a key-delegation abuse issue. In order to tackle this issue, Jiang *et al.* [85] proposed a CP-ABE-based access control scheme against key-delegation abuse in fog computing. Obviously, the data confidentiality was considered. Compared with [83], it realizes traceable CP-ABE access control to track any users who want to illegally share their privacy, thus this scheme provides traceability. However, authenticity, data integrity, identity privacy, scalability and mobility were missed in the scheme design. Its computational overhead is mainly in user side and includes t bilinear pairings. Hence, this scheme requires that the user device should have some computational power.

b) Access Control Based on Proxy Re-Encryption Technology: Proxy Re-Encryption (PRE) was also utilized to solve access control issue in edge computing, as it can translate a ciphertext with one key into another ciphertext with different key by applying a proxy. By modifying Green *et al.*'s work [86] to resist key exposures from side-channel attack, Wang [87] proposed an ID-based proxy re-encryption scheme to implement access control in fog computing. Users encrypt files using symmetric keys and upload the ciphertexts to a cloud server, and then these keys are encrypted by a public master key and stored in a fog server. When an end user wants to access the files in the cloud, the fog node re-encrypts the user's symmetric key from the public master key to his own key to achieve data access while ensuring data confidentiality. However, it needs a fully trusted private key generator (PKG), which is extremely difficult to achieve in a practical IoT system. Moreover, user revocation was not considered. Besides, authenticity, data integrity, identity privacy preservation, scalability and mobility were missed in this work. In terms of computational overhead, the user only needs to do one hash operation, $2d$ bilinear pairings and two modular multiplication operations.

In particular, Tang *et al.* [88] combined CP-ABE with PRE to implement fine-grained data sharing for big health data in fog computing, where profile information and health information are encrypted with different encryption algorithms, CP-ABE and public key encryption, respectively. After a fog node preprocesses the health data, the data is re-encrypted with a new access policy to achieve secure data sharing. Hence, data confidentiality is ensured in this work. However, health information and new access policy are disclosed to fog nodes. Moreover, authentication, data integrity, identity privacy protection, correctness of calculation, scalability and mobility were not considered. The computational overhead of this scheme includes $2(a+a^{'}+1)$ bilinear pairings, $2(a+a^{'})$ modular exponentiation operations and $a + a^{'} + 1$ modular multiplication operations in user side.

c) Access Control Based on Other Technology: Other ways were explored to implement access control in edge computing. For secure data outsourcing and access, Zahra *et al.* [89] focused on adding the Shibboleth protocol in a fog-IoT network to achieve cross domain data access control between user devices and fog nodes. The authors detailly introduced the workflow of the Shibboleth system and further demonstrated its correctness. By leveraging the metadata file in the Shibboleth protocol, this system protects data integrity. Moreover, the Shibboleth system can authenticate users before issuing access rights, which guarantees user authenticity. Identity privacy was achieved by generating a unique ID to hide the user's original identity. However, data confidentiality, mobility and scalability were not considered. In this scheme, all the security mechanisms, such as authentication, identity privacy protection, and data integrity verification are realized by directly calling the components of the Shibboleth protocol. The article does not provide specific details. Therefore, the computational complexity cannot be evaluated.

Zaghdoudi *et al.* [90] took advantage of Distributed Hash Table (DHT) to design a generic access control system for ad hoc mobile cloud computing (MCC) and fog computing. The authentication of access node and data integrity are verified by the cloud server This proposed model is suitable for spontaneous networks that is temporarily created in a pervasive mobile infrastructure and needs to respond to MCC access control demands. However, confidentiality, scalability and mobility were not discussed. Access control is implemented through an access control list achieved by DHT, thus the computational complexity depends on querying and inserting of DHT, which are not described in this work.

Yu *et al.* [91] provided a fine-grained access control scheme in fog computing by using leakage-resilient functional encryption against side channel attack. This scheme first defined the notion of leakage-resilient pair encodings and achieved

the transformation from pair encodings to leakage-resilient founction encryption in order to improve the security of access control. Data confidentiality is fulfilled with the consideration on attribute update. However, authenticity, data integrity, identity privacy preservation, scalability and mobility were not discussed. Its computational overhead includes one bilinear pairing and one modular multiplication operation in user side.

2) Secure Data Search in Edge Computing: To better protect user privacy, user data are usually encrypted before being uploaded to edge nodes or cloud servers, which sets up an obstacle for data utilization, such as data search and retrieve. In many situations, the users just need parts of data rather than the whole data. Therefore, searching over encrypted data [92] becomes a significant research topic for protecting personal privacy. In what follows, we review the state-of-art of searchable encryption in edge computing. In terms of efficiency, we herein only discuss the computational overhead of ciphertext search.

Fu *et al.* [93] designed a fog-assisted privacy-preserving cloud data storage and retrieval system for industrial Internet of Things (IIoT). Specifically, not only can data users search over encrypted data with identifiers, they can also search over encrypted data based on monitored objects with certain features. The first data search manner was achieved by constructing an ID-AVL (Adelson-Velsky and Landis) tree for hash values. In the second retrieval manner, an encrypted Retrieval Feature (RF) tree was designed by utilizing k-Nearest Neighbor (kNN) algorithm to support efficient and privacy-preserving data search. This system ensures data confidentiality. It also considers the addition and deletion of features of monitored objects, thus it satisfies the requirement on scalability. However, edge node was assumed to be honest, which is not practical. The system design neglects authenticity of user devices. Computational overhead is hard to be evaluated since it does not provide specific operations.

In order to simultaneously realize keyword search over encrypted files and access control, Miao *et al.* [94] first presented a Lightweight Fine-Grained ciphertexts Search (LFGS) system in fog computing by using CP-ABE-based keyword search. This system ensures data confidentiality. By offloading partial computational tasks of end users to a fog node, the designed algorithm lightens the computational and storage burden of end users. However, this work cannot support attribute update and conjunctive keywords search. Moreover, mobility, scalability, identity privacy and data integrity were not discussed. This system executes one modular addition operation, $2t + 1$ modular exponentiation operations and one hash operation in user side, one modular addition operation, one modular multiplication operation, $2t + 2$ modular exponentiation operations and $t+2$ bilinear pairingings in fog node side and two modular exponentiation operations, one modular multiplication operation, two symmetric encryption operations and $2t + 1$ bilinear pairings in cloud server side.

3) Secure Data Deduplication in Edge Computing: Deduplication is a technique for automatically eliminating coarse-grained and unrelated duplicate data. In addition to access control and secure data search, data deduplication is growing in importance in data storage. There are two reasons to deploy data deduplication mechanisms in edge computing. On one hand, an edge server usually collects sensing data generated by IoT devices, thus it is unavoidable to get replicated data, which leads to a high communication cost. On the other hand, because the data from different user devices is outsourced and flooded to the edge server, it becomes necessary to save storage cost. However, the data is usually encrypted before uploaded to the edge server, so secure data deduplication over encrypted data becomes a critical research topic. In what follows, we review secure data deduplication in edge computing.

Deduplication is divided into two categories based on the location where it occurs: server-side and client-side. Server-side deduplication needs data owners to upload their data to a remote server, and then the server checks data duplication and eliminates duplicated data. In the latter, the data owner only needs to upload data if they are not stored in the server. Regarding client-side deduplication, Koo and Hur [95] proposed a privacy-preserving cross-user data deduplication over encrypted data scheme in fog computing. Through efficient user-level key management and data update, this proposed scheme achieves fine-grained access control and data confidentiality. The advantage of this scheme is that the number of keys of data owners is constant regardless of the number of outsourced files. However, it does not consider data integrity during data transmission and deduplication. Moreover, mobility and scalability were missed. Besides, since the data owner always sends a request to the nearest fog node, location privacy is disclosed. In terms of computational overhead, we consider initial data upload, subsequent data upload and data decryption. The initial upload executes one hash operation, three bilinear pairings, five modular exponentiation operations and (n_1+5) modular multiplication operations in user side. In the subsequent upload, the user undertakes one hash operation, three bilinear pairings, one modular exponentiation operations and (n_1+3) modular multiplication operations. The decryption includes two bilinear pairings and $(n_1 + 2)$ modular multiplication operations in user side.

Different from [95], Ni *et al.* [96] presented a Fog-assisted Server-side Deduplication (Fo-SDD) scheme for mobile crowdsensing to prevent replicate data collection and reduce communication cost. Based on AES and full-domain hash function, fog nodes detect and eliminate repeated data in a sensing report but learn nothing about the report, thus the scheme realizes data confidentiality. Moreover, key homomorphic signature was leveraged to allow fog nodes to aggregate the signature of replicate data to achieve contribution claim. In addition, by utilizing blind signature, an extended Fo-SDD scheme protects identity privacy and location privacy of mobile users and realizes data integrity. However, traceability, mobility and scalability were not considered. Except the computational overhead of service setup and data reading phase, the scheme executes two modular exponentiation operations, one bilinear pairing, one hash operation, one modular multiplication operation and one symmetric encryption operation in customer side, three symmetric encryption operations, three bilinear pairings, two modular multiplication operations, four hash operations and seven modular exponentiation operations in initial reporter side, two symmetric encryption operations,

three bilinear pairings, two modular multiplication operations, three hash operations and five modular exponentiation operations in replicate reporter side.

By combining client-side with server-side deduplication, Koo *et al.* [97] proposed a hybrid data deduplication protocol in fog storage to achieve best-effort bandwidth. The server-side data deduplication is adopted in a user-fog network to prohibit malicious users from learning side information. The client-side deduplication is applied in a cloud-fog network. This protocol takes advantage of identity-based encryption to achieve data confidentiality and data integrity. However, an encryption key is generated based on data deduplication, which is achieved by comparing received ciphertext and the ciphertext stored in cloud/fog server, thus key management in user side is complex and the storage overhead will increase with the number of uploaded files. Moreover, scalability and mobility were not discussed. Apart from the computational overhead in setup and decryption phases, a user needs to do two symmetric encryption operations, two hash operations, one bilinear pairing and six modular exponentiation operations, and both the cloud server and the edge node need to undertake two bilinear pairings.

Owing to the feature of geographical distribution of edge computing, data are temporarily stored in lots of edge nodes in different positions, which incurs the new challenge for data query in edge computing. However, in some IoT applications, users need to query the data stored in edge nodes. Thus, it is vital to achieve a distributed and secure storage model for edge computing. He *et al.* [98] designed a secure data storage model for fog computing to improve the security of data storage. A three-layer architecture is proposed in this work, including control, authentication service and data storage layer. In the authentication service layer, a credible hierarchical deployment strategy was adopted to achieve user authentication, which ensures authenticity of user. In particular, a cooperative working mechanism was proposed to achieve efficient data query services. Moreover, a data synchronization mechanism was devised to support the storage state of the edge node changes. Since data is encrypted during transmission, data confidentiality is considered. However, mobility, scalability, data integrity, identity privacy and location privacy were missed. Since this scheme only provides a model of distributed fog storage and does not give a specific algorithm, there is no way to evaluate its computational complexity.

In cloud storage, since private data of users is outsourced to a cloud server, the users lose control on their data. This introduces various and sophisticated cyber threats, such as insider attack, data theft attack, and malicious modification. Edge computing is a promising paradigm to address these issues due to its unique features. There are some schemes that focus on cloud data storage by applying edge computing to fight against data theft attack [99] and insider attack [100], [101].

Based on a decoy technique, Hamid *et al.* [99] presented a fog-assisted cloud storage scheme to resist attacks on healthcare private data. Applying both an authenticated key agreement protocol and a photo encryption algorithm guarantee authenticity of all entities and data confidentiality. The proposed scheme undertakes one elliptic curve point multiplication, one elliptic curve point addition, one bilinear pairing and three hash operations in user side, fog node and cloud side, respectively. Due to the deployment of a decoy database, this scheme introduces some additional communication overhead. The disadvantage of this scheme is that user devices need to upload additional user photos to the decoy database in a fog node, which results in high storage overhead. Mobility, scalability, data integrity and identity privacy were not considered in this study.

To resist inside attack in a cloud server, Wang *et al.* [100] designed a three-layer hierarchical storage scheme in fog computing, which divides user data into three parts of different size with a Hash-Solomon code algorithm and stores them in local device, fog node and cloud server, respectively. Allocating the different ratio of user data stored in different devices makes insider attackers impossible to recover the real user data even if they get all data in a certain device. However, user revocation was not considered. Authenticity, trustworthiness, data integrity, identity privacy, scalability and mobility were not discussed, either.

In order to protect private data of users, Wang *et al.* [101] proposed an improved fog-based storage scheme by further revising the work in [100]. In this scheme, additional mechanisms, including a malicious modification detection algorithm and a reputation evaluation algorithm, were designed to ensure data integrity and trustworthiness. However, authenticity, scalability and mobility were not considered. With respect to computational overhead, [100] and [101] just need to perform several basic operations (including addition, multiplication, and division) to get the ratio of stored data, which is negligible.

D. Summary and Comparison of Aforementioned Existing Work

Finally, we summarize and compare in Table IV all above reviewed works by applying the following criteria:

1) Scalability (S):

- H: The work considers the join and departure of devices.
- M: The work considers the join or departure of devices
- L: The work does not consider scalability.

2) Mobility (Mo):

- H: The work supports mobility.
- M: The work only supports limited mobility.
- L: The work does not consider mobility.

3) Efficiency (E): For efficiency analysis, we consider the following operations to present computational overhead:

- Ha= Hash operation
- SE= Symmetric encryption operation
- PM= Elliptic curve point multiplication operation
- AM= Elliptic curve point addition operation
- BP= Bilinear pairing
- E= Exponentiation operation
- MA= Modular addition operation
- MM= Modular multiplication operation
- ME= Modular exponentiation operation
- Mu= Multiplication operation
- Ad= Addition operation

For ease of presentation, we ignore the fixed number of multiplication and addition in the consideration of computational overhead, as these operations are much more efficient than other complex computations.

4) Others: Besides the three aspects above, other criteria including Au, Tu, Cn, I, LP, DP, UP, IP, Ta, and Cr, have the same meanings regarding to the following marks:

- Y: It is considered in the work.
- N: It is not considered in the work.
- –: It is not mentioned in the work.

VI. Open Issues and Future Research Directions

A. Open Issues

Based on the analysis and comparison on the existing works in Section V, we outline several open issues on secure data analytics in edge computing as follows.

First of all, how to balance security and efficiency is still an open problem. From Table IV, we can observe that most works need user devices or edge nodes to perform some complex operations, which incurs high computation cost. In fact, some IoT applications in edge computing have a high demand on real-time response. The complex computation obviously impacts efficiency, especially for resource-constrained user devices. Therefore, making a trade-off between security and efficiency becomes necessary in many practical situations.

Secondly, the trustworthiness of networked devices in highly distributed network still need further researches. Though there are a few studies [53], [68], [69] on trustworthiness of networked devices, they still have some disadvantages. For example, the works in [53], [69] consider only single application scenario and the study in [68] does not support mobility and scalability. However, device trustworthiness is an important demand to ensure the QoS of edge nodes and prevent a device from malfunction. Thus, an effective trust model for edge computing should be proposed.

Thirdly, usage privacy is ignored in all reviewed works. However, it is very necessary to protect usage privacy in edge computing. Since edge computing is designed specifically for IoT applications, the collected data contain user behavior information and living habit information. An eavesdropper or an edge node could easily obtain usage patterns even if data are encrypted before uploading to an edge server. For example, an eavesdropper can predict when a user is at home through the changes of the readings of a smart meter.

Fourthly, the literature still lacks an effective solution to verify the correctness of data computation when utilizing edge nodes to do data analytics. The correctness of calculation remains high importance in outsourced data analytics [102], [103]. In edge computing, neither cloud servers nor edge nodes can be fully trusted, which makes it hard to ensure the correctness of data computation, processing and analytics. If there is no security solution to guarantee the above correctness, end users will be reluctant to use the services provided by edge computing.

Fifthly, mobility and scalability of mobile devices cannot be well supported in the current literature. Most existing works neglect the mobility and scalability of user devices when designing security schemes for edge computing. However, user devices and edge nodes might frequently migrate from one place to another. At the same time, user devices could quickly join or leave an edge network. Thus, a security scheme should support mobility and scalability in the context of edge computing, however, this is still an open issue.

B. Future Research Directions

Besides the above indicated open issues, we further propose a number of promising research directions in order to guide future research.

First, a flexible and self-adaptive data analytics is expected in edge computing. Usually, not all user data are sensitive. Some data are considered as private data, e.g., location information, health status and social relation information, while some are not, e.g., social events, environmental status information. How to automatically identify the sensitivity of user data and flexibly deal with these data becomes a key issue for achieving efficient and secure data analytics in edge computing.

Second, a lightweight and secure data analytics scheme is highly expected in edge computing. Due to limited capacities and resources, an edge node cannot perform too many complex operations (such as bilinear pairing and modular exponential operation), which could incur high latency. Especially for the applications with high real-time requirements, efficiency becomes a crucial issue in secure data analytics. Therefore, it is urgent to devise a lightweight method to accomplish secure data processing.

Third, trust management in edge computing is an interesting and significant research topic. Compared with cloud computing, edge computing makes it more troublesome in trust management due to three reasons. First of all, the decentralization of edge computing puts huge obstacles on collecting and managing evidence information about edge nodes to evaluate their trust values. Moreover, due the subjectivity of trust, different entities may have distinct security requirements on the same edge node, facing a variety of applications and services. This introduces additional challenges in trust management for edge computing. Finally, an edge node might frequently move from one area to another. For example, a vehicle equipped with a computer can work as a moving edge server. Thus, designing a universal trust model that can support both mobility and scalability become a hard problem in edge computing.

Fourth, usage privacy preservation becomes essential and vital to research in in edge computing. One direct solution is that the user device creates dummy tasks and delegates them to multiple edge nodes to hide its real tasks among the multiple tasks. However, this solution requires the user to pay for multiple tasks and wastes resources and energy. Designing an efficient and lightweight scheme to protect usage privacy is a challenging research topic.

Fifth, verifiable computation [104], [105] is expected to guarantee the correctness of data analytics edge computing. However, it is more difficult to implement verifiable computation in edge computing than cloud computing. On one hand, verifiable computation might bring high latency in edge

TABLE IV
COMPARISON OF EXISTING SECURITY WORKS IN DATA ANALYTICS BASE ON EDGE COMPUTING

Ref	Security Requirements											Performance Requirements		
	Au	Tu	Cn	I	LP	UP	IP	Ta	Cr	S	Mo	***E***		
												User side	**Edge node side**	**Cloud side**
[59]	Y	–	Y	N	–	–	N	–	–	M	M	$1*Ha+2*SE$	$2*SE$	–
[60]	Y	–	Y	N	–	–	Y	–	–	M	H	$1*Ha+1*BP+1*PM$	$1*PM+1*BP$	$2*PM$
[61]	Y	–	Y	Y	–	–	N	–	–	L	L	–	$2*ME+3*PM+2*SE+1*Ha$	$10*ME+15*PM+12*SE+9*Ha$
[62]	–	–	Y	Y	–	–	–	–	–	L	L	–	$1*Ha+(2a+1)*BP+a*MM$	$(4a+1)*ME+2a*Ha$
[64]	Y	–	Y	Y	Y	–	Y	N	–	L	H	$1*Ha+3*PM+1*AM$	$6u*PM+u*AM+3u*Ha+3u*BP$	–
[67]	Y	–	Y	Y	Y	–	Y	N	–	L	L	$3*PM+1*AM+5*Ha+5*MM+4*MA$	$2u*PM+5u*MA+2u*Ha+3u*MM$	–
[68]	–	Y	–	–	–	–	–	–	–	L	L	–	–	–
[53]	–	Y	–	–	–	–	–	–	–	L	H	$(g*h+4*g+2*n+4*A)*Mu$	–	–
[69]	–	Y	Y	–	N	–	–	N	–	M	L	$Participant$: $3*MM+3*MA$ $Querier$: $2*MM+1*MA$	$\frac{p_u}{2}(17p_u+3p_ul-9-3l)*MM+\frac{p_u}{2}(14p_u+3lp_u-6p_u-3l)*MA+l*ME$	$2p_u*E$
[70]	N	–	Y	N	–	–	N	–	–	L	L	$1*ME+1*MM+1*MA$	$5*ME+6*M+2*MA$	–
[71]	N	–	Y	–	Y	–	–	–	–	L	L	$Requester$: $2*ME+1*MM$ $Friend$: $2*ME+4*MM$	$4n(x+4)*ME+2n(x+3)*MM$	–
[72]	Y	–	Y	Y	Y	–	Y	Y	–	L	H	–	–	–
[73]	Y	–	Y	N	Y	–	Y	N	N	L	L	$Querier$: $4*ME+1*MM+1*SE+1*BP+1*Ha$ $Participant$: $2*ME+1*MM+1*SE+1*BP+3*Ha+1*MA$	$12*ME+7*MM+2*SE+6*BP+2*Ha$	–
[57]	–	–	–	–	Y	–	–	–	–	L	L	$t^{'}*Mu$	–	–
[74]	–	–	–	–	Y	–	–	–	–	L	L	–	–	–
[76]	–	Y	N	N	N	–	N	–	–	H	H	–	–	–
[77]	–	–	Y	Y	–	–	N	N	N	L	L	$1*SE+1*Ha+5*MM+3*MA$	$5*Ha+1*SE+1*ME+(u+1)*MM$	$1*Ha+1*ME+2*MM+1*MA$
[78]	Y	–	Y	Y	–	–	Y	N	N	H	L	$3*E+1*ME+2*Ha+2*MM+2*BP$	$1*Ha+2*B+2*MM+2(u-1)*AM$	$1*Ha+1*ME+2*BP$
[42]	Y	–	Y	N	N	–	N	N	N	H	L	$1*MA$	$(u+1)*MA$	$f*MA$
[79]	Y	–	Y	Y	N	–	Y	N	N	H	L	$8*MM+2*Ha+1*MM$	$Fog\ Node$: $(u+3)*Ha+(u+2)*MM$ TCA : $2*ME$ LCA : $6*ME+2*MM+1*Ha$	$1*MM+1*Ha+1*MM$

(Continued)

(Continued)

[83]	N	–	Y	Y	–	–	N	N	–	L	L	$2 * ME$	–	–
[58]	N	–	Y	N	–	–	N	N	Y	L	L	$1 * ME + 1 * MM$	$(3a + 1)p * BP$	–
[84]	N	–	Y	N	N	–	N	N	N	L	L	$1 * BP + 2 * MM$	$(a + 2) * BP + (2a + 2) * MM$	–
[85]	N	–	Y	N	–	–	N	Y	–	L	L	$t * BP$	–	–
[87]	N	–	Y	N	–	–	N	–	–	L	L	$1 * Ha + 2d * BP + 2 * MM$	–	–
[88]	N	–	Y	N	–	–	N	–	N	L	L	$2(a + a^{'} + 1) * BP + 2(a + a^{'}) * ME + (a + a^{'} + 1) * MM$	–	–
[89]	Y	–	N	Y	–	–	Y	–	–	L	L	–	–	–
[90]	Y	–	N	Y	–	–	N	–	–	L	L	–	–	–
[91]	N	–	Y	N	–	–	N	–	–	L	L	$1 * BP + 1 * MM$	–	–
[93]	N	–	Y	–	–	–	–	–	–	H	–	–	–	–
[94]	–	–	Y	N	–	–	N	–	–	L	L	$1 * MA + (2t + 1) * ME + 1 * Ha$	$1 * MA + 1 * MM + 2(t + 1) * ME + (t + 2) * BP$	$2 * ME + 1 * MM + 1 * SE + (2t + 1) * BP$
[95]	–	–	Y	N	N	–	–	–	–	L	L	$Initialupload$: $1 * Ha + 3 * BP + 5 * ME + (n_1 + 5) * MM$ $Subsequentupload$: $1 * Ha + 3 * BP + 1 * ME + (n_1 + 3) * MM$ $Decryption$: $2 * BP + (n_1 + 2) * MM$	–	–
[96]	–	–	Y	Y	Y	–	Y	N	–	L	L	$Customer$: $2 * ME + 1 * BP + 1 * Ha + 1 * MM + 1 * SE$ $Initialreporter$: $3 * SE + 3 * BP + 2 * MM + 4 * Ha + 7 * ME$ $Replicatereporter$: $2 * SE + 3 * BP + 2 * MM + 3 * Ha + 5 * ME$	$(2 + 2p_u) * ME + (1 + 2p_u) * MM + (1 + p_u) * SE + (1 + p_u) * Ha$	$p_f * ME + p_f * BP$
[97]	–	–	Y	Y	–	–	–	–	–	L	L	$2 * SE + 2 * Ha + 1 * BP + 6 * ME$	$2 * BP$	$2 * BP$
[98]	Y	–	Y	N	N	–	N	–	–	L	L	–	–	–
[99]	Y	–	Y	N	–	–	N	–	–	L	L	$3 * Ha + 1 * PM + 1 * BP + 1 * AM$	$3 * Ha + 1 * PM + 1 * BP + 1 * AM$	$3 * Ha + 1 * PM + 1 * BP + 1 * AM$
[100]	N	N	Y	N	–	–	N	–	–	L	L	–	–	–
[101]	N	Y	N	Y	–	–	N	–	–	L	L	–	–	–

computing. On the other hand, a user might continuously travel from one region to another, which incurs multiple edge nodes in different regions to work together to serve for users. As long as any one of these edge nodes makes any mistakes, the final result will be incorrect. In particular, it is extremely important to track back and remove dishonest edge nodes for the purpose of saving computational and communication overhead. Currently, provenance management is a promising technique to achieve such traceable verifiable computation [106], which will greatly help finding the origins of mistakes. However, it might bring high communication overhead. Thus, how to design an efficient verifiable computation based on provenance management is a promising topic.

Sixth, mobility, scalability and privacy protection should be jointly considered in secure data analytics. In edge computing, edge nodes and user devices have high mobility. Currently, several works focus on supporting mobility in fog computing by applying such approaches as a SDN-based method [107] and a mathematical method [108]. However, in these approaches, location privacy cannot be ensured due to location disclosure to edge nodes. Therefore, supporting high mobility and scalability and simultaneously ensuring user privacy becomes a valuable research topic in secure data analytics.

VII. Conclusion

Edge computing is regarded as a revolutionary technology to extend cloud computing to the edge of a network for supporting various IoT applications. While benefiting from the edge computing, we are still facing many security and privacy challenges. In this survey, we introduced the basic concept and features of edge computing and compared them with cloud computing. Then, we analyzed its potential security and privacy threats in order to propose a number of security requirements and performance requirements. By employing these requirements as evaluation criteria, we thoroughly reviewed and commented on the state-of-art of secure data analytics in edge computing. Based on our survey, we finally highlight out a number of open issues and proposed a number of interesting research problems to motivate future research directions in secure data analytics in edge computing.

References

[1] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, pp. 1-11, 2011.

[2] C. V. Networking, "Cisco Global Cloud Index: Forecast and Methodology, 2015-2020. White paper," *Cisco Public, San Jose,* 2016.

[3] J. Camhi, "Former Cisco CEO John Chambers predicts 500 billion connected devices by 2025," *Business Insider*, 2015.

[4] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680-698, 2018.

[5] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, pp. 30-39, 2017.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, pp. 637-646, 2016.

[7] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *2014 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2014, pp. 1-8.

[8] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, N. Bessis and C. Dobre, Eds., ed Cham: Springer International Publishing, 2014, pp. 169-186.

[9] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, 2015, pp. 37-42.

[10] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," *IEEE Access*, vol. 5, pp. 8284-8300, 2017.

[11] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in *Proceedings of the ASE BigData & SocialInformatics 2015*, 2015, p. 28.

[12] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, pp. 601-628, 2017.

[13] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," *in International Conference on Wireless Algorithms, Systems, and Applications*, 2015, pp. 685-695.

[14] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, et al., "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293-19304, 2017.

[15] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency and Computation: Practice and Experience*, vol. 28, pp. 2991-3005, 2016.

[16] Y. Guan, J. Shao, G. Wei, and M. Xie, "Data Security and Privacy in Fog Computing," *IEEE Network*, vol. 99, pp. 1-6, 2018.

[17] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, pp. 34-42, 2017.

[18] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data Security and Privacy-Preserving in Edge Computing Paradigm: Survey and Open Issues," *IEEE Access*, vol. 6, pp. 18209-18237, 2018.

[19] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *2015 IEEE 39th Annual Computer Software and Applications Conference (COMPSAC)*, pp. 53-59, 2015.

[20] P. Zhang, J. K. Liu, F. R. Yu, M. Sookhak, M. H. Au, and X. Luo, "A Survey on Access Control in Fog Computing," **IEEE Communications Magazine**, vol. 56, pp. 144-149, 2018.

[21] R. Want, B. N. Schilit, and S. Jenson, "Enabling the internet of things," *Computer*, vol. 48, pp. 28-35, 2015.

[22] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Challenges and opportunities in edge computing," in *IEEE International Conference on Smart Cloud (SmartCloud)*, 2016, pp. 20-26.

[23] O. C. A. W. Group, "Openfog architecture overview," *White Paper*, February, 2016.

[24] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, pp. 450-465, 2018.

[25] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, pp. 1587-1611, 2013.

[26] V. K. Sehgal, A. Patrick, A. Soni, and L. Rajput, "Smart human security framework using internet of things, cloud and fog computing," in *Intelligent distributed computing, ed: Springer*, 2015, pp. 251-263.

[27] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, and K. Sakurai, "Authentication in mobile cloud computing: A survey," Journal of Network and Computer Applications, vol. 61, pp. 59-80, 2016.

[28] C.-S. Tsai, C.-C. Lee, and M.-S. Hwang, "Password authentication schemes: Current status and key issues," *IJ Network Security*, vol. 3, pp. 101-115, 2006.

[29] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on consumer Electronics*, vol. 46, pp. 28-30, 2000.

[30] J. Wayman, A. Jain, D. Maltoni, and D. Maio, "An introduction to biometric authentication systems," in *Biometric Systems, ed: Springer*, pp. 1-20, 2005.

[31] G. Xu and Z. Yan, "A survey on trust evaluation in mobile ad hoc networks," in *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*, pp. 140-148, 2016.

[32] F. Yunfang, "Adaptive trust management in MANET," in *International Conference on Computational Intelligence and Security*, pp. 804-808, 2007.

[33] P. Bonatti, C. Duma, D. Olmedilla, and N. Shahmehri, "An integration of reputation-based and policy-based trust management," *networks*, vol. 2, p. 10, 2007.

[34] J.-H. Cho, A. Swami, and R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 13, pp. 562-583, 2011.

[35] C. Gentry and D. Boneh, "A fully homomorphic encryption scheme," vol. 20: Stanford University Stanford, 2009.

[36] W. Ding, Z. Yan, and R. H. Deng, "Encrypted data processing with homomorphic re-encryption," *Information Sciences*, vol. 409, pp. 35-55, 2017.

[37] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography Conference*, pp. 325-341, 2005.

[38] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223-238, 1999.

[39] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, pp. 211-407, 2014.

[40] L. Chen, R. Lu, Z. Cao, K. AlHarbi, and X. Lin, "MuDA: Multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-peer networking and applications*, vol. 8, pp. 777-792, 2015.

[41] S. Han, S. Zhao, Q. Li, C.-H. Ju, and W. Zhou, "PPM-HDA: Privacy-Preserving and Multifunctional Health Data Aggregation With Fault Tolerance," *IEEE Transactions Information Forensics and Security*, vol. 11, pp. 1940-1955, 2016.

[42] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo and M. Palaniswami, "PPFA: Privacy Preserving Fog-Enabled Aggregation in Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 3733-3744, 2018.

[43] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 735-746, 2010.

[44] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *International Workshop on Selected Areas in Cryptography*, pp. 184-199, 1999.

[45] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE Symposium on Security and Privacy*, pp. 111-125, 2008.

[46] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "De-anonymization attack on geolocated data," *Journal of Computer and System Sciences*, vol. 80, pp. 1597-1614, 2014.

[47] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks," in *Proceedings of the 2014 acm sigsac conference on computer and communications security*, pp. 537-548, 2014.

[48] A. R. Khan, "Access control in cloud computing environment," *ARPN Journal of Engineering and Applied Sciences*, vol. 7, pp. 613-615, 2012.

[49] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE transactions on information forensics and security*, vol. 8, pp. 1947-1960, 2013.

[50] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *IEEE Symposium on Security and Privacy*, pp. 44-55, 2000.

[51] Y. Wang, J. Wang, and X. Chen, "Secure searchable encryption: a survey," *Journal of communications and information networks*, vol. 1, pp. 52-65, 2016.

[52] M. Abomhara and G. M. Køien, "Security and privacy in the Internet of Things: Current status and open issues," in *International Conference on Privacy and Security in Mobile Systems*, pp. 1-8, 2014.

[53] V. Sharma, I. You, D. N. K. Jayakody, and M. Atiquzzaman, "Cooperative trust relaying and privacy preservation via edge-crowdsourcing in social Internet of Things," *Future Generation Computer Systems*, 2017.

[54] K. Vinuthna and S. Sobharani, "A Survey on Internet of Things-Fog Secure Data Inprocessing Health Services," 2017.

[55] X. Jing, Z. Yan, and W. Pedrycz, "Security Data Collection and Data Analytics in the Internet: A Survey," *IEEE Communications Surveys & Tutorials*, 2018.

[56] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, pp. 146-152, 2017.

[57] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location privacy in mobile edge clouds: A chaff-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 2625-2636, 2017.

[58] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, "A Secure and Verifiable Outsourced Access Control Scheme in Fog-Cloud Computing," *Sensors*, vol. 17, p. 1695, 2017.

[59] M. H. Ibrahim, "Octopus: An Edge-fog Mutual Authentication Scheme," *IJ Network Security*, vol. 18, pp. 1089-1101, 2016.

[60] A. B. Amor, M. Abid, and A. Meddeb, "A Privacy-Preserving Authentication Scheme in an Edge-Fog Environment," in *IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017, pp. 1225-1231.

[61] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things," *IEEE Internet of Things Journal*, vol. 4, pp. 1143-1155, 2017.

[62] 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.

[63] J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption," in *International Workshop on Public Key Cryptography*, 2002, pp. 80-98.

[64] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, pp. 772-782, 2017.

[65] Z. Eslami and N. Pakniat, "Certificateless aggregate signcryption: Security model and a concrete construction secure in the random oracle model," *Journal of King Saud University - Computer and Information Sciences*, vol. 26, pp. 276-286, 2014.

[66] H. Lu and Q. Xie, "An efficient certificateless aggregate signcryption scheme from pairings," in *International Conference on Electronics, Communications and Control*, 2011, pp. 132-135.

[67] Y. Chen, Z. Lu, H. Xiong, and W. Xu, "Privacy-Preserving Data Aggregation Protocol for Fog Computing-Assisted Vehicle-to-Infrastructure Scenario," *Security and Communication Networks*, vol. 2018, 2018.

[68] Z. Su, F. Biennier, Z. Lv, Y. Peng, H. Song, and J. Miao, "Toward architectural and protocol-level foundation for end-to-end trustworthiness in Cloud/Fog computing," *IEEE Transactions on Big Data*, 2017.

[69] L. Ma, X. Liu, Q. Pei, and Y. Xiang, "Privacy-Preserving Reputation Management for Edge Computing Enhanced Mobile Crowdsensing,"*IEEE Transactions on Services Computing*, 2018.

[70] 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 Generation Computer Systems*, vol. 78, pp. 825-837, 2018.

[71] Y. Huo, C. Hu, X. Qi, and T. Jing, "LoDPD: A location difference-based proximity detection protocol for fog computing," *IEEE Internet of Things Journal*, vol. 4, pp. 1117-1124, 2017.

[72] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved Pseudonym Scheme for Fog Computing Supported Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, pp. 2627-2637, 2018.

[73] L. Wang, G. Liu, and L. Sun, "A Secure and Privacy-Preserving Navigation Scheme Using Spatial Crowdsourcing in Fog-Based VANETs," *Sensors*, vol. 17, p. 668, 2017.

[74] R. Yang, Q. Xu, M. H. Au, Z. Yu, H. Wang, and L. Zhou, "Position based cryptography with location privacy: A step for Fog Computing," *Future Generation Computer Systems*, vol. 78, pp. 799-806, 2018.

[75] N. Chandran, V. Goyal, R. Moriarty, and R. Ostrovsky, "Position Based Cryptography," in *International Cryptology Conference on Advances in Cryptology*, 2009, pp. 391-407.

[76] T. D. Dang and D. Hoang, "A data protection model for fog computing," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 32-38.

[77] 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.

[78] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, pp. 712-719, 2018.

[79] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, et al., "APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82-92, 2019.

[80] M. Yang, T. Zhu, B. Liu, Y. Xiang, and W. Zhou, "Machine Learning Differential Privacy With Multifunctional Aggregation in a Fog Computing Architecture," *IEEE Access*, vol. 6, pp. 17119-17129, 2018.

[81] W. Ding, Z. Yan, and R. Deng, "Privacy-preserving data processing with flexible access control," IEEE Transactions on Dependable and Secure Computing, 2017.

[82] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89-98.

[83] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Generation Computer Systems*, vol. 78, pp. 730-738, 2018.

[84] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, "An efficient access control scheme with outsourcing capability and attribute update for fog computing," *Future Generation Computer Systems*, vol. 78, pp. 753-762, 2018.
[85] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 720-729, 2018.
[86] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Applied Cryptography and Network Security*, 2007, pp. 288-306.
[87] Z. Wang, "Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing," *Future Generation Computer Systems*, 2017.
[88] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. Shen, "Lightweight and Privacy-Preserving Fog-Assisted Information Sharing Scheme for Health Big Data," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, 2017, pp. 1-6.
[89] S. Zahra, M. Alam, Q. Javaid, A. Wahid, N. Javaid, S. U. R. Malik, et al., "Fog Computing Over IoT: A Secure Deployment and Formal Verification," *IEEE Access*, vol. 5, pp. 27132-27144, 2017.
[90] B. Zaghdoudi, H. K.-B. Ayed, and W. Harizi, "Generic Access Control System for Ad Hoc MCC and Fog Computing," in *International Conference on Cryptology and Network Security*, 2016, pp. 400-415.
[91] Z. Yu, M. H. Au, Q. Xu, R. Yang, and J. Han, "Towards leakage-resilient fine-grained access control in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 763-777, 2018.
[92] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Annual International Cryptology Conference*, 2007, pp. 535-552.
[93] J. Fu, Y. Liu, H.-C. Chao, B. Bhargava, and Z. Zhang, "Secure Data Storage and Searching for Industrial IoT by Integrating Fog Computing and Cloud Computing," *IEEE Transactions on Industrial Informatics*, 2018.
[94] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight Fine-Grained Search over Encrypted Data in Fog Computing," *IEEE Transactions on Services Computing*, pp. 1-1, 2018.
[95] D. Koo and J. Hur, "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 739-752, 2018.
[96] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing," *IEEE Transactions on Dependable and Secure Computing*, 2018. doi: 10.1109/TDSC.2018.2791432
[97] D. Koo, Y. Shin, J. Yun, and J. Hur, "A Hybrid Deduplication for Secure and Efficient Data Outsourcing in Fog Computing," in Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on, 2016, pp. 285-293.
[97] D. Koo, Y. Shin, J. Yun, and J. Hur, "A Hybrid Deduplication for Secure and Efficient Data Outsourcing in Fog Computing," in *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 285-293, 2016.
[98] S. He, B. Cheng, H. Wang, X. Xiao, Y. Cao, and J. Chen, "Data security storage model for fog computing in large-scale iot application," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 39-44, 2018.
[99] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A Security Model for Preserving the Privacy of Medical Big Data in a Healthcare Cloud Using a Fog Computing Facility With Pairing-Based Cryptography," *IEEE Access*, vol. 5, pp. 22313-22328, 2017.
[100] T. Wang, J. Zhou, X. Chen, G. Wang, A. Liu, and Y. Liu, "A Three-Layer Privacy Preserving Cloud Storage Scheme Based on Computational Intelligence in Fog Computing," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, pp. 3-12, 2018.
[101] T. Wang, J. Zhou, M. Huang, M. Z. A. Bhuiyan, A. Liu, W. Xu, et al., "Fog-based storage technology to fight with cyber threat," *Future Generation Computer Systems*, 2018.
[102] X. Yu, Z. Yan, and R. Zhang, "Verifiable Outsourced Computation over Encrypted Data," *Information Sciences*, 2018.
[103] Z. Yan, X. Yu, and W. Ding, "Context-aware verifiable cloud computing," *IEEE Access*, vol. 5, pp. 2211-2227, 2017.
[104] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Annual Cryptology Conference*, pp. 465-482, 2010.
[105] X. Yu, Z. Yan, and A. V. Vasilakos, "A survey of verifiable computation," *Mobile Networks and Applications*, vol. 22, pp. 438-453, 2017.
[106] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *ACM Sigmod Record*, vol. 34, pp. 31-36, 2005.
[107] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility Support for Fog Computing: An SDN Approach," *IEEE Communications Magazine*, vol. 56, pp. 53-59, 2018.
[108] P. J. Roig, S. Alcaraz, K. Gilly, and C. Juiz, "Study on mobility and migration in a fog computing environment," in *International Conference Electronics*, 2018, pp. 1-6.

Dan Liu received her B.Eng. degree in communication engineering from Jilin University, Changchun, China in 2017. She is currently pursuing the master's degree with the State Key Laboratory on Integrated Services Networks, Xidian University. Her research interests are in data analytics, privacy preservation and edge computing.

Zheng Yan received the BEng degree in electrical engineering and the MEng degree in computer science and engineering from the Xi'an Jiaotong University, Xi'an, China in 1994 and 1997, respectively, the second MEng degree in information security from the National University of Singapore, Singapore in 2000, and the licentiate of science and the doctor of science in technology in electrical engineering from Helsinki University of Technology, Helsinki, Finland. She is currently a professor at the Xidian University, Xi'an, China and a visiting professor at the Aalto University, Espoo, Finland. Her research interests are in trust, security, privacy, and security-related data analytics. Prof. Yan serves as a general or program chair for 30+ international conferences and workshops. She is a steering committee co-chair of IEEE Blockchain international conference. She is also an associate editor of many reputable journals, e.g., IEEE Internet of Things Journal, Information Sciences, Information Fusion, JNCA, IEEE Access, SCN, etc.

Wenxiu Ding received her B.Eng. degree and PhD degree in information security from Xidian University, Xi'an, China in 2012 and 2017. She was the research assistant at the School of Information Systems, Singapore Management University from 2015 to 2016. Now, she is a lecturer at the School of Cyber Engineering at Xidian University. Her research interests include RFID authentication, privacy preservation, data mining and trust management.

Mohammed Atiquzzaman received the MS and PhD degrees in electrical engineering and electronics from the University of Manchester, United Kingdom. He currently holds the Edith Kinney Gaylord Presidential professorship in the School of Computer Science at the University of Oklahoma. He is the editor-in-chief of Journal of Networks and Computer Applications, founding editor-in-chief of Vehicular Communications and has served/serving on the editorial boards of various IEEE journals and co-chaired numerous IEEE international conferences including IEEE Globecom. His research interests are in communications switching, transport protocols, wireless and mobile networks, satellite networks, and optical communications. His research has been funded by National Science Foundation, NASA, US Air Force, Cisco, Honeywell and other funding agencies.