Privacy Issues in Edge Computing

Qi Xia, Zeyi Tao, and Qun Li

1 Introduction

With the quick development of Internet of Things (IoT), massive data is produced every day. For example, photographers can easily take 100 Mb photos per day; a surveillance camera can easily take 20 Gb video record per day. Considering the numerous amount of IoT devices, the total amount of data is beyond imagination. The computational limitation of an IoT device makes it almost impossible to process and analyse surveillance camera videos and photos in real time. With the help of cloud computing, a centralized server with sufficient computational power is able to process this data. However, limited by the low bandwidth and high latency, cloud computing is not efficient enough to deal with this large amount of data in real time. Therefore, edge computing has emerged as an effective technology to reach high bandwidth and low latency [1–4]. By offloading some of the computational power and storage to the edge of the network, edge computing is capable to deliver new services and applications to billions of IoT devices, such as augmented reality, video analytics, smart home, smart hospital, Internet of vehicles, etc.

Figure 1 shows a simple structure of cloud edge infrastructure. Cloud server, which has sufficient computational resources and storage space, is usually in a data center and far away from most of end users. At the edge of the network, edge servers are geographically close to end devices to ensure high bandwidth and low latency. Edge servers usually have considerable computational resources and storage space than end devices, but not as many as cloud server. The end device usually communicates with the edge server to get a quick response.

Q. Xia (✉) · Z. Tao · Q. Li
The College of William and Mary, Williamsburg, VA, USA
e-mail: qxia@cs.wm.edu; ztao@cs.wm.edu; liqun@cs.wm.edu

W. Chang, J. Wu (eds.), *Fog/Edge Computing For Security, Privacy, and Applications*, Advances in Information Security 83,
https://doi.org/10.1007/978-3-030-57328-7_6

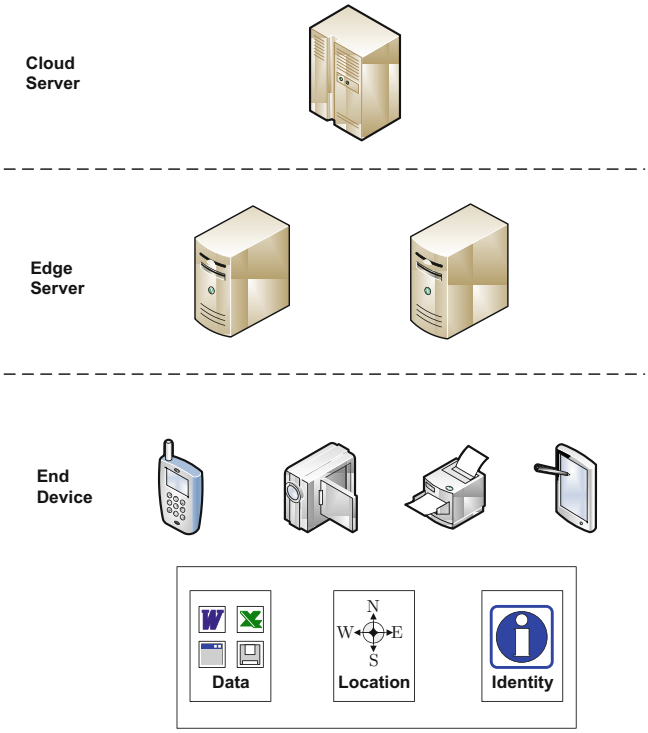

Fig. 1 Cloud edge infrastructure

However, privacy is an important issue of edge computing [5, 6]. In summary, there are three kinds of privacy concerns in edge computing.

- **Data Privacy** Because of the high bandwidth of edge computing, more data are transmitted between end devices and edge servers. This allows more private information to be transmitted. On the other hand, unlike cloud computing, which has a central data center and is usually strictly supervised, edge servers are hard to control. Therefore there may be some edge servers that are curious about the sensitive data of end users. While the end device is usually connected to the nearest edge server and may be migrated from one edge server to another edge server for a better quality of experience, it is easy to leak private data during this process. In order to protect sensitive data, in edge computing, a privacy-preserving algorithm may be run between the cloud server and the edge server or the end device and the edge server.
- **Location Privacy** In edge computing, the location privacy mainly refers to the location privacy of the edge device users. Since edge devices are usually connected to the geographically closest edge server to offload tasks for a better experience, a curious edge server, can easily infer that the location of the end device user is not far from the edge server. In addition to this concern, once the

user with the end device moves and the end device shifts service from one edge server to another edge server, by communicating with each other, the active path information of this user may even be disclosed to the curious edge server. This will pose a big challenge to protect our location information.

- **Identity Privacy** Identity privacy is also a very serious privacy issue. It occurs in some practical applications in real life. For instance, when connect with the edge server, it is very common to fill some online forms or provide sensitive personal information. This information may be stored in the edge server and later be requested for authorization. Since this information is related to end user's identity and may connect with user's payment or other sensitive information, for most people, this identity privacy is more sensitive.

Apart from these three concerns, recent research has focuses more on the combination of edge computing and emerging technologies, such as big data and machine learning. Big data and machine learning are the hottest areas recently. Edge computing is an ideal platform for big data and machine learning because of its high bandwidth and low latency. It can also arrange resource allocation to maximize the use of the computational resources from end devices, edge servers and cloud servers. However, some wireless big data and machine learning applications such as smart city, online business, smart hospital, etc., contain a lot of sensitive information, which brings new challenge to user privacy. In fact, most of the privacy issues with big data and machine learning are still about data privacy, location privacy and identity privacy. However, since the scale of the data set in these areas is much larger than the conventional privacy problem, we will introduce this problem in a separate section as well as differential privacy, which is a popular technique for big data privacy issues on the edge.

In this chapter, we will first introduce three conventional privacy issues and some existing algorithms for these issues in edge computing: data privacy, identity privacy, location privacy. Then we will discuss the privacy issues in edge computing and emerging technologies big data and machine learning. The definition, implementations and properties of a widely used technique differential privacy are introduced. Then we talk about several proposed algorithms using differential privacy in big data and machine learning privacy. In the end of this chapter, we discuss about some future work and summarize this chapter.

2 Conventional Privacy Issues in Edge Computing

2.1 Overview

The high-frequency interactions between edge users and service providers (such as data transmission, information query and online transaction) continue to arouse people's extensive attention to various privacy requirements in data, identity and location. Users want to store their private data on a data server or cloud with cheap

maintenance fees and access it anywhere at anytime with any device. Users can also collect useful information such as working hours around facilities, gasoline prices, attractions etc. around them. In addition, users can benefit from fast online payment through their portable devices. However, there exists many honest but curious adversaries such as edge data centers, infrastructure providers, service providers etc. They greatly challenge the privacy of edge computing paradigms. Compared with traditional privacy issues in cloud schemes, privacy in edge computing is more difficult to protect because for users, they do not even know whether a service provider is trustworthy. Therefore, preserving the privacy of users is a huge challenge that must be carefully considered. In this section, similar to the privacy issues in cloud computing, we consider three conventional privacy issues on data privacy, identity privacy and location privacy in edge computing area.

2.2 Data Privacy

Storing data into cloud/edge storage is attractive for most end users due to the facts that:

- users can access their data remotely and share data easily;
- users can avoid capital expenditure on physical hardware costs;
- users do not have to worry about file and storage management issues and leave this burden to the cloud/edge service.

Although cloud/edge services provide users with convenience and value, the data privacy issue remains a big challenge. Sensitive information such as photos, personal health records, and even government data may be leaked to unauthorized users and third party companies or even be hacked. While cloud service providers are usually under strict supervision, edge service providers are not trustworthy at all time, which brings more serious privacy challenges. The edge service providers therefore use firewall or virtualization to prevent data leakage. However, these mechanisms can not protect users' privacy because of untrusted edge storage services.

In cloud computing, in order to preserve the privacy of the data stored in the cloud, the conventional approach is to encrypt users' sensitive data before loading it into the cloud. Then users can retrieve the data back via keyword search or ranked keyword search. Several keyword search based encryption schemes have been proposed to ensure the privacy of data, including [7–9]. However, the main drawback of the encryption methods is their high computational cost and computational overhead. When using resource-constrained mobile devices, it may not be feasible to encrypt large size data. Another type of scheme uses symmetric-key cryptography and public-key cryptography respectively such as [10–12]. Similar to keyword based search scheme, high CPU usage and memory requirement during the encryption and decryption process become a bottleneck.

In order to mitigate the problem that the traditional encryption schemes in cloud computing cannot work well in the mobile edge cloud environment, there are several solutions.

2.2.1 Hybrid Architecture for Privacy-Preserving Data Utilization

Li et al. [13] proposed a practical hybrid architecture, in which a private cloud is introduced as an access interface between the data owner and the public cloud. This private cloud can be used as an extension for the resource-constrained mobile devices. Under this architecture, a data utilization system is provided to achieve both exact keyword search and fine-grained access control over encrypted data.

Although this architecture is introduced in cloud computing, it is suitable for resource-constrained mobile devices. This hybrid architecture has four entities in the system: data owners/users, attribute authority, private cloud and public cloud. The attribute authority is a key authority to generate public and private parameters and public cloud is used to store the data. Private cloud is a new entity to solve the problems that the computing resource of end devices is restricted. This system supports key authorization and revocation such that the access control of users is easy to manage. In addition, keyword-based query is supported so that an authorized user is able to use individual private key to generate a query for certain keywords. With the symbol-based trie, it can improve keyword search efficiency.

2.2.2 Pseudo-Random Permutation Based Method for Mobile Edge Computing

Later Bahrami and Singhal [14] proposed a new light-weight method for mobile clients to store data on one or more clouds by using pseudo-random permutation (PRP) based on chaos systems. The biggest advantage of using PRP is that this method does not require too much computational power and therefore can be run in mobile devices with low overhead.

There are two phases in this method: disassembly phase and assembly phase, which are similar to the encryption and decryption. In the disassembly phase, we split the files into several parts, which includes one file that contains the header of the original file and multiple files that contain the content of the original file. This process is based on a pseudo-random based pattern and the chunks in each file are also pseudo-randomly scramble with a chaos system. Then in the assembly phase, the end user can use the stored chaos system to reorder the chunks, and then use the stored pattern to decrypt the original file.

2.2.3 ESPPA

In 2016, Pasupuleti et al. [15] proposed a method named efficient and secure privacy-preserving approach (ESPPA). This approach utilizes the probabilistic public key encryption technique and ranked keyword search, which reduces the processing overhead of data owners while encrypting files. This also provides an efficient solution for resource-constrained mobile devices.

Unlike the solution we introduced in Sect. 2.2.1, this algorithm does not require an additional private cloud. However, it can still achieve the ranked keyword search for effective data utilization reasons. The ranked keyword search means that when the authorized user queries for a keyword, the edge cloud server finds matching files, ranks the matching files by the relevance scores and send back the top-K relevant files. In order to preserve the privacy on resource-constrained mobile devices, instead of using homomorphic encryption [7], they choose to use probabilistic public key encryption technique [16]. Therefore, the proposed method can achieve the integrity of encrypted files and index stored in the edge cloud.

2.3 Location Privacy

In recent years, more and more applications have been adopted for location-based services (LBSs) and have achieved success in many aspects such as improving traffic, road planning, finding the nearest points-of-interest (POIs) etc. Edge computing is a natural and perfect system for LBS, since the end devices usually connect to the geographically nearest edge server for a better quality of experience. To enjoy such conveniences provided by LBS, users have to send queries to the LBS server. However, these queries contain massive information such as users' locations, interests, hobbies etc. Untrusted LBS servers can easily access these sensitive personal data and release these data to third parties such as advertisers. There are two types of location based privacy issue.

- **Restricted Space Identification** For example, the disclosure of a user's location may reveal the user's real-world identity and it has the potential problems to allow an adversary to locate the subject and cause physical harm.
- **Observation Identification** For example, if a LBS provider frequently observes the user's queries for bar and liquor, the adversary may infer the user is alcoholic.

Although distinct, the above two types of privacy issues are closely related.

In order to address privacy issues in LBS and avoid personal information abuse, many approaches have been proposed over recent years. In general, they all share a simple principle of *k-anonymous*. K-anonymity was firstly introduced by Gruteser and Grunwald [17]. The location information is represented by a tuple with three intervals $([x_1, x_2], [y_1, y_2], [t_1, t_2])$. The first two intervals describe an area $\mathcal{A}$ where the specific user is located. And $[t_1, t_2]$ indicates a time period of the user being present in such an area. When the user submits the query with a location tuple to the

LBS server, the k-anonymous protocol requires an area $\mathcal{A}$ containing at least $k-1$ neighbors. Therefore k-anonymity prevents disclosure of user location by ensuring that user location information can only be accessed by LBS servers if there are at least $k-1$ distinct associated locations which are indistinguishable. Generally speaking, the larger the anonymity set k is, the higher is the degree of anonymity. To achieve k-anonymity, there are three types of solution: trusted anonymization server-based schemes, mobile device-based schemes and caching schemes.

2.3.1 Trusted Anonymization Server-Based Schemes

This method is based on a trusted central server called centralized location anonymizer [18, 19]. The goal of this anonymizer is to randomly arrange queries from end devices to one of several edge servers for protecting location privacy.

In detail, to achieve k-anonymity, a query is submitted to the LBS server via a centralized location anonymizer. The centralized location anonymizer enlarges the queried location into a Cloaking Region (CR) where the other $k-1$ neighbors are also covered. As a result, it is difficult for the untrusted LBS server to distinguish the user's real location from other users. These are simple, straightforward and effective methods. However, they suffer from a single point of failure. Since these methods heavily rely on the location anonymizers, once adversary gains control of them, the privacy of all users will be compromised. At the same time, they have a performance bottleneck because all the submitted queries have to go through a single location anonymizer.

2.3.2 Mobile Device-Based Schemes

In order to avoid the problems in Sect. 2.3.1, instead of using a centralized anonymizer, [20, 21] suggest using dummy locations, which are randomly selected from the user's mobile device to achieve k-anonymity. This can apparently solve the privacy leakage risk in centralized location anonymizer.

However, the side information (e.g., query probability) can be utilized by adversaries, and hence reducing the anonymity degree of k-anonymity. By carefully selecting dummy locations, one can potentially eliminate side information leakage [22, 23]. The drawback of these kind of solutions is quite obvious, that is, the communication and storage cost is pretty high. Some compromising solutions focus on decreasing the computational and storage overhead by using VHC mapping [24], encountered-based solution [25], and k-anonymous cloaking box [26]. In addition, it is also a heavy computational cost for resource-constrained mobile devices.

2.3.3 Caching Schemes

There are more and more research on caching schemes recently. All these schemes are based on pre-fetching the useful location-based information in cache of end devices. We list some recent works below.

Shahriyar et al. proposed Caché to improve user location privacy [27]. The core idea of this method is to periodically pre-fetch potentially useful location-enhanced content well in advance. Thus the end devices can retrieve the location-enhanced content when they need it. This protects the precise location of end users.

Xiaoyan et al. proposed an alternative method called MobiCache [28]. Their method combines k-anonymity and Dummy Selection Algorithm (DSA). In order to increase the cache hit ratio, they generate some dummy locations that have not been queried before and choose them to query. In addition, they proposed an enhanced-DSA to choose dummy locations from cells which can make more contributions to both the cache hit ratio and user's location privacy even if they are not cached before.

In 2015, Ben et al. [53] proposed another caching-based solution to protect location privacy in LBSs. In their method, they propose an entropy-based privacy metric to measure the relation between cache hit ratio and the achieved privacy. Then based on this metric, they propose Caching-aware Dummy Selection Algorithm to achieve location privacy.

Although these methods can somehow protect the location privacy of end users, end users still need to store a huge amount of service data for a large area. Besides, to cache data, the end devices need more communications and computations.

2.4 Identity Privacy

Personal Identifiable Information (PII) or user identity is information about a person which has been collected, assessed or used by edge cloud services on demand. For example, when users establish their new edge service, they usually fill out an online form and provide sensitive personal information (e.g., name, gender, address, phone number, credit card number, etc.). This information may be stored in a central Identity Provider (IdP) and may be disseminated to service providers (SPs) later for the use of authorizing requests, completing payment, customizing services and so on. In early 2018, the Facebook data scandal caused 50 million users' PII to be disclosed to third party company, Cambridge Analytica, for "analysis" purposes via SP. This practical example tells us to stay alert to protect our personal information properly. Identity privacy issues are highly related to the problem of Identity Management (IDM) in the past decade. There are several solutions recently about identity privacy in edge computing.

In 2013, Khan et al. [29] proposed a light-weight identity protection scheme for Cloud-based mobile users for dynamic credential generation instead of the digital credential method. It uses a trusted entity to offload frequently occurring dynamic

credential generation operations to reduce the computational cost on resource-limited mobile devices.

Then Park et al. [30] proposed an Improved Identity Management Protocol (I2DM) by using Pretty Good Privacy (PGP) that is based on Public Key Infrastructure (PKI) for secure mobile cloud computing. I2DM aims to find the weakest point in the network, maximize the load balance at this point to reduce the communication cost. This helps end device users manage their identity information easier.

Consolidated Identity Management (CIDM) system [31] aimed at mitigating three possible vulnerabilities: IDM server compromise, mobile device compromise, and network traffic interception. In practice, privacy is a challenge in IDM. According to [32, 33], identity management must meet the following challenges: *undetectability* aims to hide users' transactions and any other actions in a system; *unlinkability* aims to disconnect user identities and their history of transactions; *confidentiality* aims to enable users' controls. CIDM also uses a trusted third-party authorization server of IDM to manage the sensitive identity information of users. This method can distribute authorization credentials in the token into two related but different parts to countermeasure illegal access vulnerabilities. They also add a human interaction layer before each access is granted which can help defeat mobile devices that are compromised by adversaries.

3 Privacy in Edge Computing with Emerging Technologies

3.1 Overview

The emerging technology big data [34, 35] and machine learning [36, 37] have brought convenience to people's daily work and life. For example, people can talk to smart home assistant such as Siri or Alexa to get daily temperature, traffic information, control of lights and TV in their home, etc., instead of doing so by themselves; smart city can collect people's public safety, health, utility, and transportation data to help organize the city and make decisions; video analytics use machine learning and especially deep learning to analyze, classify, and process the video in real time; hospital can collect patients' symptoms and disease data and use machine learning to help them understand the disease and diagnosis more effectively.

In order for all these applications to be implemented in practice, a powerful and efficient infrastructure must be provided. Edge computing is naturally a good solution. With the benefit of high bandwidth, edge computing is capable to transmit large amounts of data to edge server for processing to help those end devices with limited computational resources. On the other hand, in some areas that require real time response such as deep neural network training and smart hospital diagnosis, low latency of edge computing is a big advantage.

However, there are several privacy concerns when we talk about big data and machine learning with edge computing. Since the high bandwidth brings more data exchange between the end device and the edge server, the data privacy concern is more serious than ever. As we cannot guarantee if the edge server is trustworthy because of the various edge server providers, the edge server may be curious about end user's private data, causing the leakage of the sensitive information. In summary, the privacy issues in machine learning usually occurs in the training phase and inference phase and the privacy issues in big data usually occurs in the data collection and data mining process.

To protect the users from these kinds of privacy concerns. A lot of privacy-preserving algorithms have been proposed recently. In summary, a good privacy-preserving algorithm must have the following requests.

- **Private** Privacy is obviously the core requirement of the algorithm. We must have privacy guarantee to prevent malicious or curious edge servers.
- **Effective** Here the effectiveness does not mean how effective this algorithm protects the privacy. It means that after we make privacy-preserving modifications to the algorithm, it should not lose the effectiveness performance a lot than state-of-the-art algorithm without preserving the privacy. For example, assuming that a face recognition model with normal training can have 95% accuracy, when we use privacy-preserving training, the accuracy should not have a big drop. If the performance drops a lot, it loses the meaning of training.
- **Scalability** The scale of the dataset in big data and machine learning is usually considerably large. Regardless of the smart city, smart hospital, or any other applications in big data, a big amount of data is the necessary foundation of big data for future analysis and implementation. Thus, the privacy-preserving algorithm must be able to deal with a large scale of dataset while the time complexity should not increase significantly with the increasing scale of the dataset.
- **Lightweight** Since one advantage of edge computing is the low latency, preserving the privacy should not bring more computational or communication overhead to the edge computing system. When we need to make considerable computations in edge server and end devices or make several data communications between edge server and end device to ensure privacy, the quality of experience will reduce significantly. So lightweight is a reasonable condition for privacy-preserving algorithms.

There are several existing techniques to preserve the privacy of big data and machine learning in edge computing. We will first introduce preliminary knowledge of differential privacy, which is a common technique in practice, then talk about some recent algorithms to protect the privacy.

3.2 Differential Privacy Preliminary

In this subsection, we will introduce some preliminary knowledges of differential privacy to help understand the recent privacy-preserving algorithms in big data and machine learning along with edge computing.

3.2.1 Definition

Differential privacy is one of the most important techniques to protect privacy recently. It is a practical technique to protect the privacy leakage problem in small perturbation of dataset. In another word, it provides a constraint to conventional algorithms that are used to analyze the dataset such that in statistics, it can limit the private information disclosure for whose information is in the dataset.

For example, there is a database that records the salary of each employer. We have a query to get the total compensation of a group of employers. Then we can query the database twice: one is for all data and the other is for all data but a specific employer. Then the difference of the two queries is the exact salary of this specific employer. We can find out that even if the query does not leak any information of a single user, the difference of subdataset can leak the private information. Theoretically we have following definitions.

Definition 1 (Query) A Query f_i is a mapping function defined on a database D. Denote $F = \{f_1, f_2, \cdots, f_n\}$ as a group of queries.

Definition 2 (Adjacent Databases) Assuming database D and D' have the same attribute structure and they differ at most one element, in another word, one database is the proper subset of the other database and the larger database contains one more additional data, we call D and D' adjacent databases to each other.

For example, if $D = \{1, 2, 3, 4, 5, 6, 7\}$ and $D' = \{1, 2, 3, 5, 6, 7\}$, D and D' differ only one element 4, then D and D' are adjacent databases.

The first theoretical definition ϵ-differential privacy is from [38] as following.

Definition 3 (ϵ-Differential Privacy) A randomized function query f gives ϵ-differential privacy if for all datasets D_1 and D_2 who are adjacent databases, and all $S \subseteq Range(f)$,

$$Pr(f(D_1) \in S) \leq e^{\epsilon} \cdot Pr(f(D_2) \in S) \tag{1}$$

The probability is taken over the coin tosses of f.

Definition 3 is a little bit abstract. Let's talk about it in details. In fact, this definition uses ϵ to control the difference of the query output distribution in those two adjacent databases. In some papers we also call this query a mechanism or algorithm. They output of the query has randomness. This definition guarantees that even if some data are added or removed from the database, the output of the query

should not be significantly changed. For example, when we want to query the total compensation of a group of people, just returning the exact total compensation is at privacy leakage risk. An ϵ-differential private query should return a range or a random number around the total compensation, so even if we run this query a lot of times to its adjacent databases, the exact salary for an individual is still protected.

ϵ-differential privacy is a powerful way to protect privacy, but sometimes Definition 3 is too strong to achieve, there is a general (ϵ, δ)-differential privacy defined in [39] as following.

Definition 4 (ϵ, δ-Differential Privacy) A randomized function query f gives (ϵ, δ)-differential privacy if for all datasets D_1 and D_2 who are adjacent databases, and all $S \subseteq Range(f)$,

$$Pr(f(D_1) \in S) \leq e^{\epsilon} \cdot Pr(f(D_2) \in S) + \delta \tag{2}$$

The probability is taken over the coin tosses of f.

Definition 4 is more general than Definition 3 since there is an added δ on the right of (2). In practice, we always choose δ as a very small constant. ϵ-differential is a special case of (ϵ, δ)-differential privacy when we choose $\delta = 0$.

3.2.2 Implementation

After we know definitions of the classic ϵ-differential privacy and the generalized (ϵ, δ)-differential privacy, we know that the differential privacy is to make modifications to query results such that query results are accurate in general but ambiguous enough to protect the privacy of the individual data. However, how to implement the differential privacy is a problem.

A natural idea to implement differential privacy is to add the noise to the query result so that the result can be more ambiguity. For example, in the previous example about the total compensation, we can add some noise to the total compensation result such that the query result is not exact number of total compensation but an approximate result around the exact number. Then we can use this result as the query output and it can protect the privacy of a single data.

Practical ways to implement differential privacy are based on this idea of adding noise. Basically there are three ways to add the noise: Laplace mechanism, Gaussian mechanism and exponential mechanism. We will talk about them one by one. Before we talk about these three mechanisms, we first define the global and local sensitivity of a query, or function.

Definition 5 (Global Sensitivity) For a query function $f : D \rightarrow \mathbb{R}^d$ where D is a database and $\mathbb{R}^d$ is a d-dimensional real number vector, its global sensitivity in any adjacent databases D and D' is

$$GS_f = \max_{D, D'} ||f(D) - F(D')|| \tag{3}$$

Table 1 Laplace, Gaussian and exponential mechanisms

Mechanism	Modified query result	Noise PDF
Laplace	$f(D) + N(0, GS_f^2\delta^2)$	$\frac{1}{2b}e^{-\frac{\lvert x-\mu\rvert}{b}}$
Gaussian	$f(D) + N(0, GS_f^2\delta^2)$	$\frac{1}{b\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2b^2}}$
Exponential	$Pr(M_{ex}(D) = r) = \frac{e^{\frac{\epsilon q(D,r)}{2\cdot GS_{q(\cdot,r)}}}}{\sum_r e^{\frac{\epsilon q(D,r)}{2\cdot GS_{q(\cdot,r)}}}}$	N/A

Definition 6 (Local Sensitivity) For a query function $f : D \to \mathbb{R}^d$ where D is a database and $\mathbb{R}^d$ is a d-dimensional real number vector, its local sensitivity in a given database D and all its adjacent database D' is

$$LS_{f(D)} = \max_{D'} ||f(D) - F(D')|| \tag{4}$$

The Laplace, Gaussian and exponential mechanisms are listed in Table 1.

According to [39–41], we have the following Theorem 1 to introduce the privacy guarantee of these three mechanisms.

Theorem 1 *If we use the Laplace mechanism and exponential mechanism as the query result, then it satisfies ϵ-differential privacy. If we use Gaussian mechanism as the query result, then it satisfies (ϵ, δ)-differential privacy.*

3.2.3 Properties

When we have several different differential private algorithms, one question is how the privacy will change after we combine them. Here we introduce two major combination methods: sequential and parallel. Theoretically, we have following theorems.

Theorem 2 (Sequential Composition) *If $M_1, M_2, \cdots, M_n$ are algorithms or queries that access a private databse D such that M_i satisfies ϵ_i-differential privacy, then the combination of their outputs satisfies ϵ-differential privacy where $\epsilon = \epsilon_1 + \epsilon_2 + \cdots + \epsilon_n$.*

Theorem 3 (Parallel Composition) *If $M_1, M_2, \cdots, M_n$ are algorithms or queries that respectively access disjoint database $D_1, D_2, \cdots, D_n$ such that M_i satisfies ϵ_i-differential privacy, then the combination of their outputs satisfies ϵ-differential privacy where $\epsilon = \max(\epsilon_1, \epsilon_2, \cdots, \epsilon_n)$.*

These properties provides the privacy guarantee for multiple combinations of algorithms. Therefore, once we have more than one differential private algorithms, we can also achieve differential privacy with sequential or parallel combinations of them.

3.3 Privacy-Preserving Algorithms

In this section, we will introduce some privacy preserving algorithms in big data and machine learning along with edge computing in details. It should be noted that the privacy problems in this area are at a very new stage and still have a lot of work to do, some pioneers have already found some important topics along with a series of techniques to solve these problems. The two major problems in this area are:

- Machine learning model training and prediction privacy.
- Big data privacy.

There are also some other privacy topics in this area. We will talk about those problems in the following sections.

3.3.1 Preserving Privacy in Machine Learning

The model training and prediction are the most important part of machine learning. They are also two of the major procedures in big data analysis. A good machine learning model can provide very accurate fitting and prediction results, which brings a lot of useful information to help us study the dataset and make a convenience to human beings. However, the model training of deep learning or big data analysis always requires a large demand of computational resources. To distribute the computational resources, edge computing provides a natural platform to offload the heavy computation from the end devices to edge servers or cloud servers.

The data privacy is always a big concern in this process. From the trained model or the training process, a lot of information can be leaked. Recent research can even rebuild the dataset from the training model by generative adversarial networks [42]. Therefore it is very important to protect the privacy while training the model. In summary, it has two phases of privacy leakage concerns:

- **Training Phase** Training a machine learning model encourages the machine to learn the explicit knowledge of the training dataset. With edge computing, there are two ways of training. The first one is the distributed machine learning. Multiple end devices or edge servers work together to train a model. In this scenario, privacy may leak when the centralized parameter server is malicious or some peer workers are curious. The second case is leveraging the training phase to the edge server by end device. Because some machine learning models are complicated such as deep neural networks, it is a common way to leverage part of the training to the edge server. There are lots of interactions between end devices and edge server such as parameter transmission and weight update, which can also leak private information.
- **Inference Phase** After the machine learning model is trained, the application is to use the trained model to do inference. While the most of the models are deployed in the edge server, end user can communicate with the edge server to implement the inference phase. The input data are uploaded to the edge server

and the inference results will be returned to the end device. This can cause data or identity leakage.

Differential privacy is a very common technique to use in this area. One of the problems in differential privacy is that it will lose the accuracy of the query if we want to protect more privacy since we add noise to query results. However in machine learning training, this is not a very big concern because in machine learning and big data, the model training is usually based on numerous data and to learning the statistics regulation among them. It also usually needs a lot of epochs of training to get a good performance, so an individual noise is not a big deal and those noise is assumed in the statistical model itself. Since in differential privacy, the noise we added usually satisfies a distribution of mean value 0, after several epochs of training, the interference of the noise will be offset. Therefore, differential privacy is a good way to protect privacy in machine learning model training. There are several existing works in this area.

Model Partition

In 2018, Yunlong et al. [54] proposed a model partition based privacy preserving model training algorithm in edge computing. Their algorithm mainly focuses on deep neural network model training.

In deep neural network, there are multiple layers in the neural network. Training a whole network on the end device is a cumbersome work and usually consumes a lot of time. A natural idea is to offload this work to edge server. To protect the privacy, they split the whole neural network into two parts. The first part contains only the first layer and the activation function. The other part contains all rest layers. To reasonably distribute computational resources and protect the privacy, they let the first part run on end devices and the second part run on the edge server. To protect the privacy, in the forward propagation process, Gaussian noise is added to activation results of the first part so that the adversary cannot refer the original input data by activation results. In the backward propagation, the backward loss change information is transmitted from the edge server to end devices to finish one iteration.

It is apparent that activation results that is transmitted from end devices to the edge server is (ϵ, δ)-differential private because of the Gaussian noise added to results. However, to ensure the privacy, it is also important to ensure that when all activation results pass through the second part of neural networks on the edge server, the output is still private. We can take the output as a combination of input activation results. From properties of differential privacy, since the total loss of network prediction can be seen as a composed mechanism of multiple differential private mechanisms, the final output also satisfies differential privacy. This means that this model partition way can preserve the privacy in neural network model training on edge computing.

Output Perturbation and Objective Perturbation

Output perturbation and objective perturbation are two algorithms that was proposed by Miao et al. in 2018 [43]. Unlike the model partition, output perturbation and objective perturbation can not only be used in deep neural networks, but also extend to other machine learning techniques.

The difference between those two algorithms is that output perturbation is adding Laplace noise to the output prediction to preserve the privacy of the model prediction and objective perturbation is adding Laplace noise to the objective function or loss function to preserve the privacy of the model training.

Assuming the modified loss function is $K(u, D)$, we have:

$$K(u, D) = \frac{1}{n} \sum_{i} Loss(u(x_i), y_i) + \lambda Z(u) \tag{5}$$

where $u(\cdot)$ is the function of the prediction model, x_i, y_i are the data and target in database D, $Loss(\cdot)$ is a loss function and $Z(\cdot)$ represents the smoothness of the function. Then the prediction model U is to minimize this modified loss function to get a good model.

$$U = \arg\min_{u} K(u, D) \tag{6}$$

Output perturbation algorithm is adding a random Laplace noise q to the result of U:

$$U'(x) = U(x) + q \tag{7}$$

This can protect the privacy of prediction results.

On the other hand, objective perturbation algorithm adds Laplace noise to the modified loss function $K(u, D)$:

$$K'(u, D) = K(u, D) + \frac{1}{n} q^T u \tag{8}$$

By minimizing the L_2 normalization of $K'(u, D)$, we can get the prediction model U by:

$$U = \arg\min_{u} K'(u, D) + \frac{1}{2} ||u||^2 \tag{9}$$

From Du et al. [43], the privacy analysis is given to ensure the ϵ-differential privacy in those two algorithms. They can respectively protect the data privacy during the training and prediction processes.

Separate Training

Separate training is another algorithm proposed by Mengmeng et al. to protect the data privacy [44]. It uses two edge servers and assumes that there are no collusion between both of them.

Within the separate training aggregation framework, two edge servers collect data from sensors in each area. Once a sensor collect some data, it will randomly split the collected data into two parts and randomly send each part with Laplace noise to an edge server. The process will continue until edge servers collect enough data. Both edge servers will aggregate the received data and train a machine learning model based on those data. With the assumption that there is no collusion between edge servers, each edge server cannot get or infer the completed data by itself. When executing queries, both of edge servers compute queries based on their models. To protect the privacy, Laplace noises are added to query results. They by aggregating prediction results from both edge servers, a privacy-preserving prediction result is obtained.

The authors of this paper proved that if each record is independent in this dataset, the aggregated result can also provide ϵ-differential privacy. Meanwhile, from properties of differential privacy, we know that even we have a series of private mechanisms, the composition of them can still provide a ϵ-differential privacy.

3.3.2 Preserving Privacy in Big Data

The emerging development of internet of things, massive data are produced and shared everyday. Big data is a science to study and analyze those large scales of data. Edge computing has accelerated the speed of collection data. In edge computing, end devices including various kinds of sensors are collecting numerous amount of data and transmit to edge servers everyday.

Among big data one important privacy issue is that how to provide privacy guarantee when publish or release those data. For example, when the data publisher releases various statistics of the data, the adversary can query those statistics to recover the original data, which will significantly disclose some important message or private information. So how to preserve the data privacy while publishing the big data statistics is a big problem recently. On the other hand, since the data are usually stored scatteredly on edge servers. Then how to prevent privacy breaches from honest but curious edge servers is a challenging issue. In the implementation of big data techniques, there are two phases that may leak the privacy:

- **Data Collection** Data collection and aggregation are the most intuitive way to leak the sensitive information. The first step of big data is to collect a large scale of data and aggregate them, among which a lot of data are sensitive. Meanwhile, the uncertainty of the credibility in edge servers increases the risk of privacy leakage. Once the sensitive data is taken advantage by malicious edge servers, the privacy issue can be really severe.

- **Data Mining** Data mining and data analysis are the core step of big data. Data mining uses some techniques such as statistics learning, machine learning to study the implicit information of the data. These information usually include the identity, location, preference, habit information of a single person or a group of people. While users usually leverage those data mining procedures to the edge server, privacy-preserving algorithms must be used here to protect against information leakage by untrusted edge servers.

Differential Privacy is still a common technique in the big data privacy area. Since for the large scale of data, one row of data is not that important. People are more willing to care about the approximate result. Therefore, adding noise to the data will not harm the total performance. In this section, we will summarize some recent privacy preserving algorithms in big data.

Partitioned Histogram Data Publishing Algorithm

Partitioned histogram data publishing algorithm is proposed by Yi et al. in [45]. It is an algorithm to protect the data privacy when the edge server release the data. On one hand, they add noise to the data to keep data private between end devices and edge server against privacy leakage. On the other hand, during the data transmission, only the partition histogram of the data statistics will be transmitted to cancel the impact of the noise.

The algorithm contains four steps. First of all, we need to divide the original dataset into several histogram bins and add Laplace noise to each bin. Secondly, cluster partition operation is used to obtain new partition histogram. Thirdly, we are going to use this new partition histogram to build a wavelet tree using wavelet transform and add Laplace noise to the wavelet tree. Lastly, we can restore the histogram partition and publish the private histogram.

The privacy analysis is still based on the differential privacy. Since this algorithm has added the Laplace noise twice, authors proved that the partition histogram publishing algorithm based on wavelet transform satisfies ϵ-differential privacy.

Content-Based Publish–Subscribe Scheme

The content-based publish-subscribe scheme is proposed by Qixu et al. in [46]. It provides privacy protection to brokers in the publish-subscribe system. The publish-subscribe system is widely used in modern applications. It is a scheme to categorize the published messages and send categorized messages to subscribers by brokers. Edge computing provides a effective platform for this system. However, privacy issues are in the implementation of publish-subscribe scheme in edge computing because there may be unethical brokers or brokers are facing risk of hacking, sniffing and corrupting, which may cause private information leakage.

This algorithm includes three major steps. The publish-subscribe system firstly generated notification messages by using top-K U-FIM algorithm based on user's dataset to min top-K most frequent itemsets and adding exponential mechanism to results. Then based on the results, Laplace noise is added to the operated dataset. In the end, the broker uses attributes of top-K most frequent itemsets to match corresponding events to publish.

Exponential and Laplace mechanisms are used here to ensure the ϵ-differential privacy. So it is both suitable for numeric data and non-numeric data.

3.3.3 Other Topics

There are a lot of other topics in this area. In fact, since this is a very new research area, the topics are various.

- Smart home hub privacy [47]. They propose a smart home system called HomePad, which uses elements in a directed graphs to represent applications in this system and use module functions to isolate the usage of the data. It can achieve user defined privacy policy by modeling elements and the flow graph using Prolog rules.
- Differential privacy–based location privacy [48]. A differential privacy based framework is proposed by Qiucheng et al. to protect location privacy. They build a noise quadtree to map two-dimensional spatial data into an interval tree, and nodes in the tree correspond to a certain sub-area of the two-dimensional spatial data. Then they used Hilbert curve to reduce the retrieval computation cost.
- Online social multimedia big data retrieval privacy [49]. To support big data analytics while preserving privacy in edge computing, they can build multimedia content cluster tree from top to the bottom to handle the dynamically varying cached MC datasets and add noise to the cluster tree. In addition, they propose an evaluation method to measure the credibility of edge nodes.
- Smart city privacy [50]. The authors talk about modeling the privacy content among the big data including data graph, information graph and knowledge graph of smart city in DIKW (Data, Information and Knowledge) architecture. They categorize context graphs into target resources and add privacy guarantee to the conversion for IoT devices to process.
- Internet of connected vehicles privacy [51]. They design a V2V (Vehicle-to-Vehicle) communication-based route-obtaining algorithm to offload the computation to edge nodes. Then they propose NSGA-II (non-dominated sorting genetic algorithm II) to reduce the computational cost and preserve privacy of vehicles tracking, identity tampering and virtual vehicle hijacking.
- Federated learning privacy [52]. They adopt a blockchain to replace the centralized server in the classic federated learning system to reduce the privacy leakage risk. In addition, they also add differential privacy noise to extracted features and intermediate computational results in order to protect end user's privacy and enhance test accuracy.

All these topics are new and interesting. In reality, many work of privacy problems in edge computing are still at an early stage and have a lot of work to explore. We will talk about them in the next section.

4 Future Work

In the previous section, we mentioned that the privacy problems in edge computing are still in a very early stage. In fact, with the rapid development of edge computing and the gradual demand of privacy, more and more people have started to pay attention to the privacy issues. In addition to the topics we discussed in previous sections, we believe that there will be more interesting privacy problems in edge computing area. Here are some examples.

- Real time analysis. In edge computing, low latency brings a lot of opportunities on real time analysis. While sensors and end devices collect a lot of dynamic data and need to analyze them in real time, privacy guarantee should also be provided in a timely manner. Therefore, there must be efficient and effective privacy-preserving algorithms in real time.
- Privacy overhead. Nowadays, privacy-preserving algorithms are more focused on how to preserve privacy effectively. However, in order to obtain privacy, there are more computational overheads, which may introduce significant latency to the edge computing environment. Therefore, it is also important to preserve privacy while saving computational costs.
- Privacy accuracy balance. At present, many privacy-preserving algorithms, especially those that use differential privacy, will introduce noise into query results, which may lead to a decrease in accuracy. How to balance the privacy guarantee and the accuracy reduction will be an interesting problem.
- Smart vehicle. Self-driving vehicle is a trend of the future car. A lot of companies and research institutions are paying attention to it. Edge computing provides convenience on sensing data and vehicle communication. However, driving information is important and sensitive. How to provide the privacy for self-driving car will be a very popular privacy problem.

5 Summary

In this chapter, we have talked about some existing privacy-preserving algorithms and useful techniques in conventional data, location and identity privacy and new technologies machine learning and big data, but there are a lot of more interesting open privacy problems in edge computing. It is a growing demand for people to protect their private information in this age of information explosion. The privacy problem in edge computing must be paid more attention in the future.

References

1. Satyanarayanan, M.: The emergence of edge computing. Computer **50**(1), 30–39 (2017)
2. Shi, W., et al.: Edge computing: vision and challenges. IEEE Internet of Things J. **3**(5), 637–646 (2016)
3. Yi, S., Li, C., Li, Q.: A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data (2015)
4. Yi, S., et al.: Fog computing: platform and applications. In: 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb). IEEE, Piscataway (2015)
5. Zhang, J., et al.: Data security and privacy-preserving in edge computing paradigm: survey and open issues. IEEE Access **6**, 18209–18237 (2018)
6. Yi, S., Qin, Z., Li, Q.: Security and privacy issues of fog computing: a survey. In: International Conference on Wireless Algorithms, Systems, and Applications. Springer, Cham (2015)
7. Yu, J., et al.: Toward secure multikeyword top-k retrieval over encrypted cloud data. IEEE Trans. Dependable Secur. Comput. **10**(4), 239–250 (2013)
8. Liu, Q., Wang, G., Wu, J.: Secure and privacy preserving keyword searching for cloud storage services. J. Netw. Comput. Appl. **35**(3), 927–933 (2012)
9. Kuzu, M., Islam, M.S., Kantarcioglu, M.: Efficient similarity search over encrypted data. In: 2012 IEEE 28th International Conference on Data Engineering. IEEE, Piscataway (2012)
10. Cao, N., et al.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. **25**(1), 222–233 (2013)
11. Örencik, C., Savaş, E.: An efficient privacy-preserving multi-keyword search over encrypted cloud data with ranking. Distrib. Parallel Databases **32**(1), 119–160 (2014)
12. Wang, C., et al.: Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Trans. Parallel Distrib. Syst. **23**(8), 1467–1479 (2011)
13. Li, J., et al.: Privacy-preserving data utilization in hybrid clouds. Future Gener. Comput. Syst. **30**, 98–106 (2014)
14. M. Bahrami, M. Singhal, A light-weight permutation based method for data privacy in mobile cloud computing. In: 2015 Third IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, San Francisco, pp. 189–198 (2015)
15. Pasupuleti, S.K., Ramalingam, S., Buyya, R.: An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing. J. Netw. Comput. Appl. **64**, 12–22 (2016)
16. Menezes, A.J., et al.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
17. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the First International Conference on Mobile Systems, Applications and Services (2003)
18. Mokbel, M.F., Chow, C.-Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proceedings of the 32nd International Conference on Very Large Data Bases (2006)
19. Chow, C.-Y., Mokbel, M.F., Aref, W.G.: Casper* Query processing for location services without compromising privacy. ACM Trans. Database Syst. **34**(4), 1–48 (2009)
20. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: ICPS'05. Proceedings. International Conference on Pervasive Services, 2005. IEEE, Piscataway (2005)
21. Lu, H., Jensen, C.S., Yiu, M.L.: PAD: privacy-area aware, dummy-based location privacy in mobile services. In: Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access (2008)
22. Niu, B., et al.: Achieving k-anonymity in privacy-aware location-based services. In: IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, Piscataway (2014)
23. Niu, B., Li, Q., Zhu, X., Li, H.: A fine-grained spatial cloaking scheme for privacy-aware users in location-based services. In: 2014 23rd International Conference on Computer Communication and Networks (ICCCN), Shanghai, pp. 1–8 (2014)

24. Pingley, A.: CAP: a context-aware privacy protection system for location-based services. In: 2009 29th IEEE International Conference on Distributed Computing Systems. IEEE, Piscataway (2008)
25. Manweiler, J., Scudellari, R., Cox, L.P.: SMILE: encounter-based trust for mobile social services. In: Proceedings of the 16th ACM Conference on Computer and Communications Security (2009)
26. Hu, H., Xu, J.: Non-exposure location anonymity. In: 2009 IEEE 25th International Conference on Data Engineering. IEEE, Piscataway (2009)
27. Amini, S., et al. Caché: caching location-enhanced content to improve user privacy. In: Proceedings of the Ninth International Conference on Mobile Systems, Applications, and Services (2011)
28. Zhu, X., Chi, H., Niu, B., Zhang, W., Li, Z., Li, H.: MobiCache: when k-anonymity meets cache. In: 2013 IEEE Global Communications Conference (GLOBECOM), Atlanta, pp. 820–825 (2013)
29. Khan, A.N., Mat Kiah, M.L., Madani, S.A., et al.: Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. J. Supercomput. **66**, 1687–1706 (2013)
30. Park, I., Lee, Y., Jeong, J.: Improved identity management protocol for secure mobile cloud computing. In: 2013 46th Hawaii International Conference on System Sciences, Wailea, pp. 4958–4965 (2013)
31. Khalil, I., Khreishah, A., Azeem, M.: Consolidated Identity Management System for secure mobile cloud computing. Comput. Netw. **65**, 99–110 (2014)
32. Birrell, E., Schneider, F.B.: Federated identity management systems: a privacy-based characterization. IEEE Secur. Privacy **11**(5), 36–48 (2013)
33. Werner, J., Westphall, C.M., Westphall, C.B.: Cloud identity management: a survey on privacy strategies. Comput. Netw. **122**, 29–42 (2017). ISSN 1389-1286
34. Chen, M., Mao, S., Liu, Y.: Big data: a survey. Mobile Netw. Appl. **19**(2), 171–209 (2014)
35. McAfee, A., et al.: Big data: the management revolution. Harv. Bus. Rev. **90**(10), 60–68 (2012)
36. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2020)
37. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge (2012)
38. Dwork, C.: Differential privacy: a survey of results. In: International Conference on Theory and Applications of Models of Computation. Springer, Berlin (2008)
39. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
40. Dwork, C., et al.: Calibrating Noise to Sensitivity in Private Data Analysis. In: Theory of Cryptography Conference. Springer, Berlin (2006)
41. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07). IEEE, Piscataway (2007)
42. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (2014)
43. Du, M., et al.: Differential privacy preserving of training model in wireless big data with edge computing. IEEE Trans. Big Data **6**(2), 283–295 (2018)
44. Yang, M., et al.: Machine learning differential privacy with multifunctional aggregation in a fog computing architecture. IEEE Access **6**, 17119–17129 (2018)
45. Qiao, Y., et al.: An effective data privacy protection algorithm based on differential privacy in edge computing. IEEE Access **7**, 136203–136213 (2019)
46. Wang, Q., et al.: PCP: a privacy-preserving content-based publish–subscribe scheme with differential privacy in fog computing. IEEE Access **5**, 17962–17974 (2017)
47. Zavalyshyn, I., Duarte, N.O., Santos, N.: HomePad: a privacy-aware smart hub for home environments. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, Piscataway (2018)
48. Miao, Q., Jing, W., Song, H.: Differential privacy–based location privacy enhancing in edge computing. Concurrency Comput. Pract. Experience **31**(8), e4735 (2019)

49. Zhou, P., et al.: Differentially-private and trustworthy online social multimedia big data retrieval in edge computing. IEEE Trans. Multimedia **21**(3), 539–554 (2018)
50. Duan, Y., et al.: Data privacy protection for edge computing of smart city in a DIKW architecture. Eng. Appl. Artif. Intell. **81**, 323–335 (2019)
51. Xu, X., et al.: An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Future Gener. Comput. Syst. **96**, 89–100 (2019)
52. Zhao, Y., et al.: Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: a secure, decentralized and privacy-preserving system. arXiv preprint arXiv:1906.10893 (2019)
53. Niu, B., et al.: Enhancing privacy through caching in location-based services. In: 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, Piscataway (2015)
54. Mao, Y., et al.: Learning from differentially private neural activations with edge computing. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, Piscataway (2018)