Privacy-Preserving Subset Aggregation with Local Differential Privacy in Fog-Based IoT

Lele Zheng(✉), Tao Zhang, Ruiyang Qin, Yulong Shen, and Xutong Mu

School of Computer Science and Technology, Xidian University, Xi'an 710071, China
{llzheng,xtmu}@stu.xidian.edu.cn, taozhang@xidian.edu.cn,
ylshen@mail.xidian.edu.cn

Abstract. As a typical novel IoT(Internet of Things) architecture, fog-based IoT is promising to decrease the overhead of processing and movement of large-scale data by deploying storage and computing resources to network edges. However, since the edge-deployed fog nodes cannot be fully trustable, fog-based IoT suffers some security and privacy challenges. This paper proposes a novel privacy-preserving scheme that can implement data aggregation from a subset of devices in fog-based IoT. Firstly, our scheme identifies the subset to be aggregated by computing the Jaccard similarity of attribute vectors of the query users and IoT devices, where the local differential privacy is employed to protect the attribute vectors. In addition, we use local differential privacy truth discovery to protect the data of IoT devices and improve the accuracy of the aggregation result. Finally, experiments show that our scheme is efficient and highly available by comparing it with state-of-the-art works. Theoretical analyses demonstrate that our proposed scheme has excellent performance on both computational costs and communication costs.

Keywords: Fog-based IoT · Local differential privacy · Truth discovery

1 Introduction

With the rapid development of the IoT, more and more physical devices (such as smartphones, vehicles, etc.) can freely connect to the Internet and generate various valuable data for different applications. Many emerging IoT applications rely

Supported by the National Key R&D Program of China (Grant 2018YFB2100400), the Natural Science Basic Research Program of Shaanxi (Program No. 2019ZDLGY13-03-01, 2020CGXNG-002, 2021ZDLGY07-05), the Fundamental Research Funds for the Central Universities (Grant No. JB210306), the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University(Grant No. YJS2103).

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 399–412, 2021.
https://doi.org/10.1007/978-3-030-89814-4_29

on generating large amounts of data to provide users with better services [20], but extensive computing, communication, and storage resources are required. To cope with such IoT applications' proliferation, fog computing has become a promising supplement to cloud computing by extending the network functions from the cloud to network edges.

However, there are still many unsolved problems in fog–based IoT architecture, in which security and privacy issues occupy one of the most critical positions [3,16]. The data submitted by the IoT device may contain sensitive information about the device (such as the location of the vehicle, the brand of the mobile phone, etc.). Once malicious attackers obtain these data, the user's privacy will be leaked. Therefore, the data of IoT devices should be protected before they are uploaded. Besides, compared with traditional IoT, fog-based IoT has a more complex network environment and architecture. The fog node needs to collect data from multiple data sources and provide novel aggregation services (selective data aggregation) for different applications. Fog computing can select some relevant nodes for data collection according to users' demands to offer personalized services. However, malicious attackers can infer the user's preferences from attributes of the IoT devices selected by the user. Thus, IoT devices' attributes should be protected while safeguarding the data because they also implicitly leak privacy, especially when the data source comes from a smartphone [10]. Therefore, it is urgent for us to protect IoT devices' attributes and data in fog-based IoT.

Our Contributions. Motivated by the above statements, in this paper, we investigate a novel privacy-preserving subset aggregation scheme. Using Jaccard similarity estimation and truth discovery subject to local differential privacy, our scheme can protect user's query vectors, IoT devices' attribute vectors, and values of IoT devices' data. The contributions of this paper can be summarized as follows:

1. We propose a novel secure subset aggregation scheme in fog-based IoT, where a Jaccard similarity estimation method that satisfies local differential privacy is utilized to select a desirable subset. Our scheme can protect the attribute of IoT devices and efficiently select nodes that meet the conditions.
2. We use local differential privacy technology to ensure IoT device data security and use the truth discovery method that satisfies local differential privacy to improve the aggregated results' accuracy.
3. Experiments claim our proposed scheme can effectively estimate the similarity and aggregate data. The aggregation result has high utility while we protect the privacy of both users and IoT devices. With the growth of the number of attributes and IoT devices, our scheme's time overhead increases very slowly.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 introduces the system model and design goals. The details of our scheme are presented in Sect. 4. The analysis and experimental evaluations are shown in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related Work

The security of data aggregation has been studied for a long time. There are many privacy-preserving data aggregation schemes [2], such as multi-dimensional aggregation, fault-tolerant aggregation, and differential privacy aggregation. Moreover, Lu et al. [11] proposed a lightweight privacy protection data aggregation scheme for fog computing to enhance the Internet of Things, which focuses on saving communication bandwidth and many of the security as mentioned earlier features. Cheng et al. [5] proposed reliable and privacy-protected selective data aggregation based on fog-based IoT. HASSAN et al. [12] proposed a privacy protection subset aggregation scheme called PPSA in fog-enhanced IoT scenarios, which enables query users to obtain the sum of data from a subset of IoT devices. However, the enormous computational overhead in these schemes reduced the availability of them.

Similarity estimation is used in many fields, such as social networks, recommendation systems. The most commonly used methods are inner product, euclidean distance, cosine similarity, Jaccard similarity. However, conventional methods can not protect the privacy of two vectors. To solve this problem, Lu et al. [11] proposed a scheme based on homomorphic encryption, which used homomorphic encryption to calculate two vectors' inner product. The Euclidean distance based on homomorphic encryption has been presented by Zhang et al. [21]. Homomorphic encryption and zero-knowledge proof have been used to compute the cosine similarity in [19]. [6] computed Jaccard similarity between encrypted data by homomorphic encryption. These schemes have massive computation and communication costs, so the Jaccard similarity estimation schemes satisfied local differential privacy has been proposed. [1] proposed an LDP Jaccard similarity estimation scheme based on the Laplace mechanism, and [18] proposed a scheme based on the exponential mechanism.

Truth discovery is a technique that can improve the accuracy of the aggregation result, usually used in crowdsensing [8,13]. Truth discovery updates the weight based on the distance and result of the participant's data so that the impact of low-quality data is reduced. However, these truth discovery schemes can not protect the privacy of participants. Some works use encryption or secure multi-party computation techniques to protect privacy [14,15]. Li et al. [9] proposed a local differential privacy truth discovery scheme, which can effectively compute the truth value and protect users' privacy.

3 System Model and Design Goal

3.1 System Model

Figure 1 shows our system model. We divide it into four layers: IoT-device layer, Fog layer, Cloud layer, and Query User layer.

IoT-Device Layer: A set of IoT devices $I = \{I_1, I_2, \cdots, I_N\}$ are deployed at the IoT-device layer. Each device $I_i \in I$ is equipped with sensing, calculation,

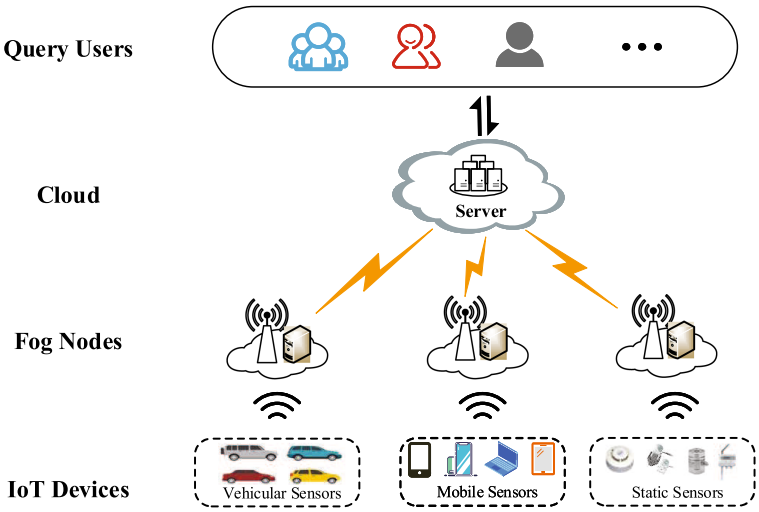


Fig. 1. System model

and communication modules, which allows them to collect, disturb, and upload data to the fog layer. In our model, IoT devices periodically collect data $D_i =< atr_i, x_i >$, where $atr_i \in \{0,1\}^m$ is the IoT devices' attributes and $x_i \in R$ is the value of the data. They respectively use privacy-preserving 1-bit minhash to perturb the attributes of IoT devices and use Laplace noise to perturb the values of the data. Finally, IoT devices upload their perturbed data $\widehat{D_i} =< \widehat{art_i}, \widehat{x_i} >$ to fog servers without disclosing any sensitive information.

Fog Layer: In the fog layer, fog servers with computing capabilities are deployed in each area. After receiving the user's query $\widehat{U}$ sent via the cloud layer, fog servers request IoT devices to upload their perturbed attributes $\widehat{art_i}$. In order to achieve efficiently privacy-preserving subset aggregation, fog servers use local differential privacy Jaccard similarity estimation to select IoT devices. Finally, the fog server will collect the value $\widehat{x_i}$, where $i \in I'$ from IoT devices, and upload them to the cloud layer.

Cloud Layer: In our model, the cloud layer is composed of servers, which are responsible for sending user queries and data aggregation from the fog layer. Here we use the local differential privacy truth discovery algorithm to improve the data's accuracy and send the obtained data to the user. All data and queries are disturbed, so there will be no privacy leakage.

Query User Layer: The user layer is mainly responsible for disturbing user queries and sending queries to the cloud layer. In our model, the query user selects a subset of IoT devices $I' \subseteq I$ to obtain the aggregate sum value of their prepared data D. User has an attribute vector $U \in \{0,1\}^m$ means the user wants to aggregate the data of the IoT devices which satisfy the demand U. That is to say, the user wants to query the data of the devices in a subset $I' \subseteq I$, and

each I_i in the I' satisfies the similarity $S(U, atr_i) > \tau$, and τ is a threshold. The user also uses the private 1-bit minhash to perturb the attribute to protect the private information.

3.2 Security Model

In our security model, all entities, deployed at both device and fog layers, are assumed to be honest-but-curious participants, i.e., they are obliged to follow the protocols faithfully, but may be curious about the query information during data preparation and throughout the query processing steps. For example: 1) the fog node may try to identify the IoT devices Ii with exact data xi; 2) each IoT device may be curious about other IoT devices' data; 3) the user may be curious about each IoT device Ii's prepared data or, at least, prepared data in each IoT device in I. Note that the honest-but-curious assumption would be guaranteed in practice since the service providers should protect their own reputation and financial interests. Finally, there would be no collusion between entities such as sharing their results.

3.3 Design Goal

Considering the above system model, our goal is to design a privacy-preserving subset aggregation scheme in fog-based IoT. Precisely, the following three objectives should be achieved:

Privacy-Preserving. In our scheme, the user uses the private 1-bit minhash to perturb the attribute to protect the private information. IoT devices use private 1-bit minhash to perturb the attribute of the data and use Laplace noise to perturb the data's value.

Efficient. With fog servers deployed at the fog layer, the cloud server can communicate with IoT devices efficiently through fog servers. Our local differential Jaccard similarity estimation has low computational overhead compared with other schemes using homomorphic encryption. And IoT devices can offload the calculation task of Jaccard similartiy estimation to fog servers to get higher efficiency.

High Utility. In our scheme, the cloud server uses a truth discovery algorithm that satisfies local differential privacy to aggregate the data of IoT devices. Truth discovery improves the accuracy of the aggregation result by decreasing the weight of low-quality data so that users can get high utility results.

4 Our Proposed Scheme

4.1 Workflow

(1) System Initialization. In the system initialization, we assume the cloud server will bootstrap the whole system. The cloud server will set the number of private minhash functions to K, chose K different hash functions, and set the privacy budget to ϵ.

(2) Query Generation at The User's Side. The user has an attribute vector $U \in \{0,1\}^m$ meaning that the user wants to satisfy the demand U. That is to say, the user wants to query the aggregation result of the data of IoT devices which are in a subset $I' \subseteq I$, and each I_i in the I' satisfies the similarity $S(U, atr_i) > \tau$, where τ is a threshold.

The user also uses the noisy 1-bit minhash to perturb the query vector to protect the private information as $\hat{U} = h(U)$, where $h(\cdot)$ is a noisy minhash function. Then the user sends the query $\hat{U}$ to fog servers from different areas through the cloud server.

(3) Similarity Estimation at Fog Server. In this step, each fog server receives the perturbed query $\hat{U}$ from the user and sends the request of uploading attribute vectors to IoT devices. After receiving the request, IoT devices will use noisy 1-bit minhash to perturb their attributes atr_i, get the perturbed attributes $\widehat{atr_i} = h(atr_i)$, then upload them to the fog server. Fog server executes the similarity estimation to compute the similarity between each $\widehat{art_i}$ and $\hat{U}$, and selects the data whose similarity is greater than τ using the local differential privacy Jaccard silmilarity.

Finally, the fog server will collect the value of IoT devices' data whose attribute satisfies the user's query and upload the values $\{\hat{x}_{i \in I'}\}$ to the cloud server.

(4) Data Aggregation at Cloud Server. After receiving the data from fog servers, the cloud server uses local differential privacy truth discovery to increase the accuracy of the result of the aggregation. Finally, the cloud server sends the result to the user.

4.2 Local Differential Privacy Jaccard Similarity Estimation

In our model, users need the data that satisfies themselves, so we need to estimate the similarity of the user's query vector and IoT devices' attribute vectors. There are many ways to compute the similarity of two vectors, such as inner product, cosine similarity, Jaccard similarity. In fog-based IoT, we should protect users and IoT devices' privacy, so those privacy protection schemes of inner product and cosine similarity always use homomorphic encryption, which gives a heavy computation burden to users and IoT devices. Minhash is a standard method to estimate the similarity of two sets, and it can reduce the costs of communication and protect privacy to some extent. Local differential privacy Jaccard similarity protects privacy effectively by adding Laplace noise at the vector after minhash. Meanwhile, it can estimate the similarity effectively. We estimate the similarity as shown in Algorithm 1.

Minhash. Our scheme uses minhash algorithm to estimate the similarity of two vectors. A minhash function is defined as that, $h : \{0,1\}^m \to [m] := \{1, ..., m\}$ which is a random shuffle of a vector. The hash value of $x \in \{0,1\}^m$ is the position of the first 1 in x after the shuffle. Obviously, we can know the property of minhash that for any pair $x, y \in \{0,1\}^m$, we have $Pr[h(x) = h(y)] = J(x, y)$ where

Algorithm 1. Local Differential Privacy Jaccard Similarity Estimation

Require: user's query $vector < U >$, IoT device attributes $vector < I >$, number of hash functions k.
Ensure: similarity est.
1: $dist \leftarrow 0$
2: **for** $\{i = 0; i < k; i++\}$ **do**
3: $\quad dist \leftarrow dist + (U_i - I_i)^2$
4: **end for**
5: $est \leftarrow 1 - 2/k * dist + 8 * (\Delta f/\varepsilon)$
6: return est

the probability depends on the random choice of h. Then randomly choosing K different hash functions gets result $(h_1(x), \cdots, h_K(x)) \in [m]^K$. By linearity of expectation, the value $\frac{1}{K}\sum_{i=1}^{K}[h_i(x) = h_i(y)]$ is an unbiased estimator of $J(x, y)$.

1-bit Minhash. Riazi et al. [17] described a secure hash function construction based on mapping the produced minhash value to a random bit, building on the idea of b-bit minhash from [7]. Formally, given a random minhash function $h_{min} : \{0,1\}^m \rightarrow [m]$ and a random hash function $h_r : [m] \rightarrow \{0,1\}$, let $h(x) = h_r(h_{min}(x))$. For two vectors x, y, it is well known that $Pr(h(x) = h(y)) = (1 + J(x, y))/2$. Repeating the construction K times, we estimate $J(x, y)$ as $\frac{2}{K}\sum_{i=1}^{K}[h_i(x) = h_i(y)] - 1$.

Lemma 1. Let $x, y \in \{0,1\}^m$, and every vector has at least $\tau \geq 1$ attribute(s), we say that two vectors are neighboring if they differ in at most α positions, so that $J(x, y) \geq 1-\alpha/\tau$. Let $h_1, \cdots, h_K$ be K random 1-bit minhash functions. Let $x^* = (h_1(x), \cdots, h_K(x)), y^* = (h_1(y), \cdots, h_K(y))$. Let $\delta > 0$. With probability at least $1 - \delta$, the number of different positions between x^* and y^* is at most $(K\alpha)/(2\tau) + \sqrt{3ln(1/\delta)(K\alpha)/(2\tau)}$.

Proof. Let the $X = \sum_{i=1}^{K} X_i$, where $i \in [K]$ and $X_i = [h_i(x) \neq h_i(y)]$ is a random variable. Since all X_i are independent and $Pr(X_i = 1) = \frac{1-J(x,y)}{2} \leq \alpha/(2\tau)$, we can get $E[x] = K\alpha/(2\tau)$. Then using Chernoff bound $Pr(X > (1 + \beta)E[x]) \leq exp(\frac{-\beta^2 E[x]}{3})$, and let $\beta = \sqrt{3ln(1/\delta)/E[x]}$.

We can get $Pr\left(X > (K\alpha)/(2\tau) + \sqrt{3ln(1/\delta)(K\alpha)/(2\tau)}\right) \leq 1/\delta$, is that $Pr\left(X \leq (K\alpha)/(2\tau) + \sqrt{3ln(1/\delta)(K\alpha)/(2\tau)}\right) \geq 1 - 1/\delta$, so Lemma 1 is proved.

Noisy MinHash. Let K, α and τ be integers, and let $\varepsilon > 0$ and $\delta > 0$ be the privacy budget. By Lemma 1, we can get the sensitivity $\Delta f = (K\alpha)/(2\tau) + \sqrt{3ln(1/\delta)(K\alpha)/(2\tau)}$. According to [4], the user and each IoT node with $x \in \{0,1\}^m$ can add Laplace noise $N_{x,i} \sim Lap(\Delta f/\varepsilon)$ to get vector $\hat{x} = (h_1(x) + N_{x,1}, \cdots, h_K(x) + N_{x,K})$ with K 1-bit minhash functions $h_1, \cdots, h_K$, so getting (ε, δ)-differential privacy.

Algorithm 2. Local differential privacy Truth discovery

Require: noisy data from IoT devices $Vector < X >$.
Ensure: final truth *Result*.
1: Randomly initialize the truth *Result*
2: **repeat**
3: **for** $\{i = 0; i < n; i++\}$ **do**
4: Update the weight ws based on current estimated ground truth using Equation $w_i = \omega(\frac{d(x_i, Result)}{\sum_{i=1}^{n} d(x_i, Result)})$
5: **end for**
6: Update the truth Result based on current weights using Equation $Result = \frac{\sum_{i=1}^{n} w_i * x_i}{\sum_{i=1}^{n} w_i}$
7: **until** Convergence criterion is satisfied
8: return *Result*

Similarity Estimation. Given $\hat{x}$ and $\hat{y}$ from R^K, their similarity can be estimated as:

$$\hat{J}(\hat{x}, \hat{y}) = 1 - \frac{2}{K} \sum_{i=1}^{K} (\hat{x_i} - \hat{y_i})^2 + 8(\Delta f/\varepsilon)$$

And $\hat{J}(\hat{x}, \hat{y})$ is an unbiased estimator for $J(x, y)$.

Using linearity of expectation, we prove as follows:

We use E_d to express the expectation of the distance of two vectors x and y. We can easily get $E_d = E[\sum_{i=1}^{K} (\hat{x_i} - \hat{y_i})^2]$, because of the linearity of expectation, $E_d = \sum_{i=1}^{K} E[(\hat{x_i} - \hat{y_i})^2]$. Every hash function has the same expectation, we can use h_1 to be the representative, then we expand E_d, get $E_d = KE[((h_1(x) - h_1(y)) + (N_{x,1} - N_{y,1}))^2] = KE[(h_1(x) - h_1(y))^2 + 2(h_1(x) - h_1(y)(N_{x,1} - N_{y,1}) + (N_{x,1} - N_{y,1})^2]$. Both $N_{x,i}, N_{y,i}$ are independently chosen, so $E[N_{x,i}] = E[N_{y,i}] = 0$, get $E_d = K(E[(h_1(x) - h_1(y))^2] + E[(N_{x,1} - N_{y,1})^2])$. According to the property of expectation, we know $E[(N_{x,1} - N_{y,1})^2] = Var[N_{x,1} - N_{y,1}] + (E[N_{x,1} - N_{y,1}])^2$, and we know $Var[N_{x,1}] = 2(\Delta/\varepsilon)^2$, then we can get $E_d = KE[(h_1(x) - h_1(y))^2] + 2KVar[N_{x,1}] = KE[(h_1(x) - h_1(y))^2] + 4K(\Delta f/\varepsilon)^2$. Because $(h_1(x) - h_1(y))^2$ just has two possible values, so $E_d = K(0^2 Pr[h_1(x) = h_1(y)] + 1^2 Pr[h_1(x) \neq h_1(y)]) + 4K(\Delta f/\varepsilon)^2$. $Pr[h_1(x) = h_1(y)] = J(x, y), Pr[h_1(x) \neq h_1(y)] = 1 - J(x, y)$, that is $E_d = K/2(1 - J(x, y)) + 4K(\Delta f/\varepsilon)^2$.

From what has been discussed above, we can get:

$$J(x, y) = 1 - \frac{2}{K} E[\sum_{i=1}^{K} (\hat{x_i} - \hat{y_i})^2] + 8(\Delta f/\varepsilon)^2$$

4.3 Local Differential Privacy Truth Discovery

In the local differential privacy data aggregation model, all data are perturbed by IoT devices. The privacy of IoT devices has been protected, but the noise of the data may be significant. The traditional aggregation scheme treats all data equally, making some low-quality data affect aggregation results' accuracy.

By using truth discovery, the data with high weight will contribute more to aggregation results.

Our truth discovery is used on the perturbed data. The server will receive the data which have been perturbed, then aggregate the perturbed data $\{\hat{x}_{i\in I}\}$ from all IoT devices by conducting local differential privacy truth discovery to obtain the final output. The local differential privacy truth discovery can improve the utility of data and protect IoT devices' privacy. The method is shown in Algorithm 2.

Data Perturbation. Firstly, each IoT device perturbs its data by adding Laplace noise $\hat{x_i} = x_i + N_x \sim Lap(\Delta f/\epsilon)$, then uploads it to the server. We denote the perturbation mechanism as M, the original data as x_i. So the M satisfies ϵ-Local Differential Privacy. For any subset $S \in R$, and two different records x_1 and x_2, the following inequality holds:

$$Pr\{M(x_1) \in S\} \leq e^{\epsilon} Pr\{M(x_2) \in S\}$$

This definition compares the probability of observing the perturbed value of two different records x_1 and x_2 in the same range. In this step, the weights of IoT devices are inferred based on the current aggregated results. An IoT device will have a high weight if it provides information close to the aggregated results. Typically, the weights of IoT devices are calculated as follow:

$$w_i = \omega(\frac{d(x_i, result)}{\sum_{i=1}^{n} d(x_i, result)})$$

Where distance function $d(x, y)$ is a function that measures the distance between the value provided by the IoT device and the aggregated result, and weight computation function $\omega(\cdot)$ is a monotonically decreasing function. If the distance is small, the IoT device's data will get a high weight. In our scheme, weight computation function $\omega(\cdot)$ is $-log(\cdot)$.

Aggregation. In the aggregation step, the weights of IoT devices are fixed. We compute aggregated results as follow:

$$result = \frac{\sum_{i=1}^{n} w_i * x_i}{\sum_{i=1}^{n} w_i}$$

where w_i is the weight of i-th IoT device, and the x_i is the perturbed value of the i-th IoT device data. In this weighted aggregation scheme, the final result relies on those IoT devices which have high weights.

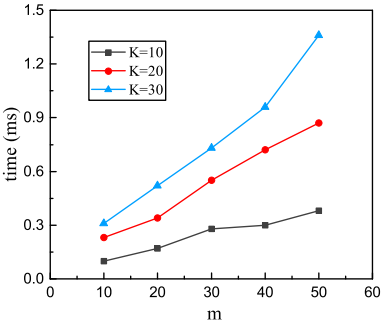
5 Evaluation

This section evaluates our scheme's computational costs and compares them with [12], which compute the similarity of two vectors with homomorphic encryption. The environment of the experiment is Intel core i7-4790 CPU @ 3.60GHz with windows 10 operating system.

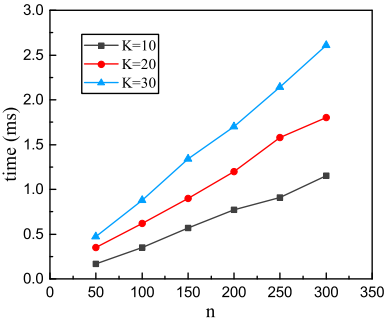
We run our experiments 1000 times for different parameter values. We show the average execution result for each function in Fig. 2 with variables of the number of attributes m, the number of hash functions K, the number of IoT devices n, and the round of truth discovery t.

5.1 Experiment

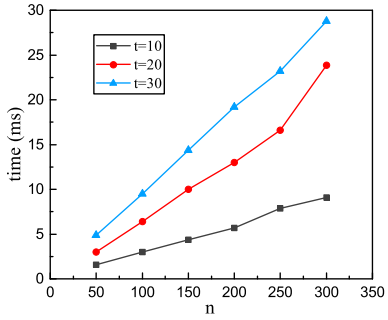
1) In Fig. 2(a), we show the time cost of the generation of user's query and IoT devices' attribute vectors for different m and K. Because the times of computation are just related to the number of the attributes and hash functions, not related to what the attributes are, we randomly select some attributes to compute the time cost, and the computational cost will increase when K or m becomes larger.



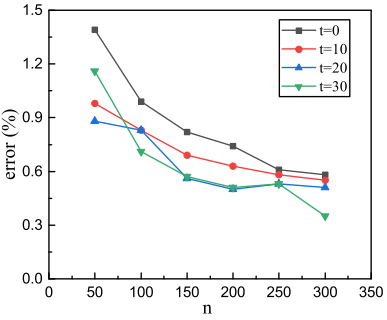
(a) user's query generation time.



(b) similarity estimation time.



(c) truth discovery time.



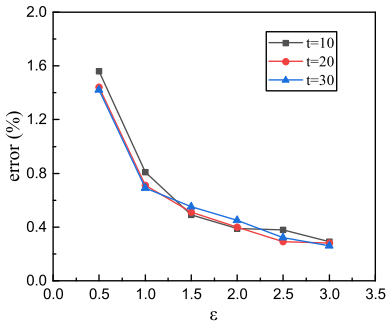
(d) truth discovery error.

Fig. 2. Experimental result

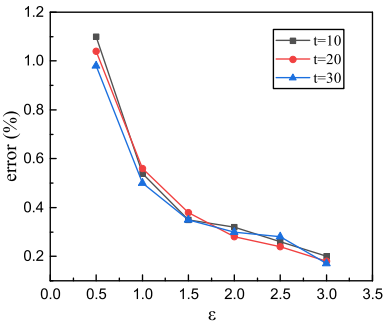
2) In Fig. 2(b), we show the time cost of similarity estimation of two attribute vectors for different n and K. As same as 1), there is no effect on computation for which attributes two vectors have. The computational cost will increase rapidly as k increases.

3) In Fig. 2(c), we show the time cost of truth discovery of the noisy values for different n and t. We randomly generate some temperature data and add Laplace noise to them. When t becomes larger, the calculation cost will also increase.

4) In Fig. 2(d), we show the mean absolute percentage error (MAPE) of truth discover for different n and t. We compare the aggregation results of noisy data and raw data and compute the percentage of the raw data aggregation result, which difference between noisy results and raw result accounts. Through comparing with the method which does not use truth discover(when $t = 0$), we show that the error can be reduced by using truth discover.



(a) MAPE on truth discovery, varying ϵ from 0.5 to 3.0, fixing $n = 100$.



(b) MAPE on truth discovery, varying ϵ from 0.5 to 3.0, fixing $n = 200$.

Fig. 3. Experimental result

5) In Figs. 3(a) and 3(b), we show the mean absolute percentage error of truth discover for different ϵ and t when the $n = 100$ and $n = 200$. Our experimental results demonstrate that the error will reduce with increasing privacy budget ϵ.

6) In Table 1, we compare our user's query generation's time cost with user's query generation in [12] called PPSA. We let our $K = 20$ and 30 because their scheme's time cost is too high, so we let their key generation parameter $\kappa = 512$ and 1024. The number of attributes of our scheme and PPSA is m.

7) In Table 2, we compare our similarity estimation step's time cost and the matching step with PPSA. We set the same parameters, let our $K = 20$ and 30, and let their key generation parameter $\kappa = 512$ and 1024. Let $m = 20$, and n is the number of IoT devices. From the table, we can find out that our scheme's costs increase very slowly, but homomorphic encryption increases rapidly.

Table 1. User's query generation time (ms)

m	Algorithm			
	Our scheme		PPSA	
	K = 20	K = 30	$\kappa = 512$	$\kappa = 1024$
5	0.13	0.20	46	230
10	0.26	0.30	58	402
15	0.28	0.40	79	569
20	0.32	0.52	102	710
25	0.43	0.65	123	873
30	0.46	0.71	145	1003

Table 2. Subset selection time (ms)

n	Algorithm			
	Our scheme		PPSA	
	K = 20	K = 30	$\kappa = 512$	$\kappa = 1024$
50	0.27	0.42	714	4807
100	0.57	0.81	1405	9428
150	0.85	1.17	2057	14184
200	1.10	1.57	2746	18893
250	1.40	1.95	3403	23828
300	1.67	2.36	4095	28672

5.2 Analysis

In this section, we will analyze the computational costs and the communication costs of our scheme.

Computational Costs. In the user's query generation step, the user just needs to compute the noisy attribute vectors. A minhash function's time complexity is $O(m)$, where m is the number of attributes. The user needs to compute K hash values, so the time complexity of generating the step is $O(mK)$. In the matching step, the fog servers need to compute the similarity between the user's query vectors and the IoT devices' vectors. If there are n IoT devices, the time complexity is $O(nm)$. In the truth discovery step, the cloud server needs to aggregate all results which satisfy the user's query, if we run t rounds truth discovery, the time complexity is $O(tn)$.

Communication Costs. IoT devices need to send their attribute vectors and noisy data, so each IoT device's communication cost is $O(K)$, where K is the number of hash functions. Every fog server will receive the attribute vectors and noisy data from IoT devices, so the communication cost is $O(n_i K)$, where n_i is the number of IoT devices for i-th fog server. After similarity estimating, each

fog server will upload the data to the cloud server. For each fog server, the most communication cost of this step is $O(n_i)$, so the cloud server's communication cost is $O(\sum n_i)$.

6 Conclusion

This paper proposed a scheme to select the particular subset privately and effectively by using Jaccard similarity estimation that satisfies local differential privacy. In addition, our scheme used local differential privacy truth discovery to increase the accuracy of the aggregation result. Using local differential privacy jaccard similarity estimation and truth discovery, our scheme can protect user's query vectors, IoT devices' attribute vectors, and IoT devices' data values. Thus there is no private information being disclosed in our model. Moreover, experimental and analysis results show that our scheme has excellent performance in efficiency and accuracy.

References

1. Aumüller, M., Bourgeat, A., Schmurr, J.: Differentially private sketches for Jaccard similarity estimation. In: Satoh, S., et al. (eds.) SISAP 2020. LNCS, vol. 12440, pp. 18–32. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60936-8_2
2. Bao, H., Lu, R.: A new differentially private data aggregation with fault tolerance for smart grid communications. IEEE Internet Things J. **2**(3), 248–258 (2017)
3. Chiang, M., Ha, S., Chih-Lin, I., Risso, F., Zhang, T.: Clarifying fog computing and networking: 10 questions and answers. IEEE Commun. Mag. **55**(4), 18–20 (2017)
4. Dwork, C., Roth, A.: The Algorithmic Foundations of Differential Privacy (2013)
5. Huang, C., Liu, D., Ni, J., Lu, R., Shen, X.: Reliable and privacy-preserving selective data aggregation for fog-based IOT. In: 2018 IEEE International Conference on Communications (ICC) (2018)
6. Le, T.T.N., Phuong, T.V.X.: Privacy preserving Jaccard similarity by cloud-assisted for classification. Wirel. Personal Commun. (5) (2020)
7. Li, P., König, A.: B-bit minwise hashing, October 2009
8. Li, Y., Gao, J., Lee, P.P.C., Su, L., He, C., He, C., Yang, F., Fan, W.: A weighted crowdsourcing approach for network quality measurement in cellular data networks. IEEE Trans. Mob. Comput. **16**(2), 300–313 (2017)
9. Li, Y., et al.: Towards differentially private truth discovery for crowd sensing systems, October 2018
10. Lu, K.: Checking more and alerting less: detecting privacy leakages via enhanced data-flow analysis and peer voting (2015)
11. Lu, R., Heung, K., Lashkari, A.H., Ghorbani, A.A.: A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT. IEEE Access **PP**, 1 (2017)
12. Mahdikhani, H., Mahdavifar, S., Lu, R., Zhu, H., Ghorbani, A.A.: Achieving privacy-preserving subset aggregation in fog-enhanced IoT. IEEE Access **7**, 184438–184447 (2019)
13. Meng, C., et al.: Truth discovery on crowd sensing of correlated entities, pp. 169–182 (2015)

14. Miao, C., et al.: Cloud-enabled privacy-preserving truth discovery in crowd sensing systems. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, pp. 183–196, SenSys 2015, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2809695.2809719
15. Miao, C., Su, L., Jiang, W., Li, Y., Tian, M.: A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems. In: IEEE INFOCOM 2017 - IEEE Conference on Computer Communications (2017)
16. Ni, J., Zhang, K., Lin, X., Shen, X.S.: Securing fog computing for internet of things applications: challenges and solutions. IEEE Commun. Surv. Tutorials **20**(99), 601–628 (2018)
17. Riazi, M.S., Chen, B., Shrivastava, A., Wallach, D., Koushanfar, F.: Sub-linear privacy-preserving near-neighbor search (2016)
18. Yan, Z., Wu, Q., Ren, M., Liu, J., Liu, S., Qiu, S.: Locally private Jaccard similarity estimation. Concurr. Comput. Pract. Exper. **31**(24), e4889 (2019)
19. Yang, D., Xu, B., Yang, B., Wang, J.: Secure cosine similarity computation with malicious adversaries. In: Chaki, N., Meghanathan, N., Nagamalai, D. (eds.) Computer Networks and Communications (NetCom). LNEE, vol. 131, pp. 529–536. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-6154-8_52
20. Yu, S., Wang, G., Liu, X., Niu, J.: Security and privacy in the age of the smart internet of things: an overview from a networking perspective. IEEE Commun. Mag. **56**, 14–18 (2018). https://doi.org/10.1109/MCOM.2018.1701204
21. Zhang, J., Hu, S., Jiang, Z.L.: Privacy-preserving similarity computation in cloud-based mobile social networks. IEEE Access **PP**(99), 1 (2020)