Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A new data clustering strategy for enhancing mutual privacy in healthcare IoT systems

Xuancheng Guo [a], Hui Lin [a,*], Yulei Wu [b,*], Min Peng [a]

[a] *Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, China*
[b] *College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK*

A R T I C L E I N F O

Article history:
Received 10 January 2020
Received in revised form 23 May 2020
Accepted 8 July 2020
Available online 16 July 2020

Keywords:
Healthcare IoT systems
Machine learning
Privacy preservation
Homomorphic encryption

A B S T R A C T

In the rapidly growing era of Internet-of-Things (IoT), healthcare systems have enabled a sea of connections of physical sensors. Data analysis methods (e.g., k-means) are often used to process data collected from wireless sensor networks to provide treatment advices for physicians and patients. However, many methods pose a threat of privacy leakage during the process of data handling. To address privacy issues, we propose a mutual privacy-preserving k-means strategy (M-PPKS) based on homomorphic encryption that neither discloses the participant's privacy nor leaks the cluster center's private data. The proposed M-PPKS divides each iteration of a k-means algorithm into two stages: finding the nearest cluster center for each participant, followed by computing a new center for each cluster. In both phases, the cluster center is confidential to participants, and the private information of each participant is not accessible by an analyst. Besides, M-PPKS introduces a third-party cloud platform to reduce the communication complexity of homomorphic encryption. Extensive privacy analysis and performance evaluation results manifest that the proposed M-PPKS strategy can achieve high performance. In addition, it can obtain approximate clustering results efficiently while preserving mutual private data.

1. Introduction

The advancement of Internet-of-Things (IoT) [1,2] can transform any object into network data through wireless sensor networks. This has derivated a broad spectrum of diversified applications, such as smart cities [3,4], smart homes [5,6], and smart grids [7,8]. Besides, IoT has been widely adopted to design and promote healthcare systems [9,10], due to its effective capability of integrating within infrastructure resources and provisioning crucial information to users. Healthcare systems can collect a multitude of data through wireless sensor networks (WSNs) to help propagate many e-health services, e.g., electronic health cards, remote patient monitoring and health portals. Fig. 1 shows a framework of typical healthcare IoT systems [11,12]. It consists of three layers: perception layer, network layer and application layer.

- Perception layer: The perception layer is the base of the healthcare IoT systems. It flexibly supports the connection of various sensor devices through WSNs. The sensors in healthcare systems include wearable sensors and implant sensors. These sensors can gather sensitive healthcare data from patients (or the person who provides sensitive healthcare data, called participants) and upload them to the network layer.
- Network layer: The network layer is the intermediate bridge in a healthcare IoT system. It is mainly responsible for the coding, authentication and transmission of sensitive healthcare data.
- Application layer: The application layer provides rich IoT applications. It primarily implements intelligent decision making and service provision by processing and analyzing the collected healthcare data.

The k-means algorithm [13] has been widely used to provide useful diagnostic recommendations for physicians and patients, through analyzing the explosively growing sensor data. Given that healthcare systems are often constrained by limited computing capability, data analysis jobs are often outsourced to third-party companies with professional analysts. If an analyst is malicious, sensitive information of patients has the risk of being sold on the black market, which can lead to data privacy leakage and potentially threaten the lives of patients [14]. Accordingly, how to analyze the sensitive data while protecting the participants' privacy data has been a hot research topic [15–18]. An

* Corresponding authors.
E-mail addresses: linhui@fjnu.edu.cn (H. Lin), y.l.wu@exeter.ac.uk (Y. Wu).

https://doi.org/10.1016/j.future.2020.07.023
0167-739X/© 2020 Published by Elsevier B.V.

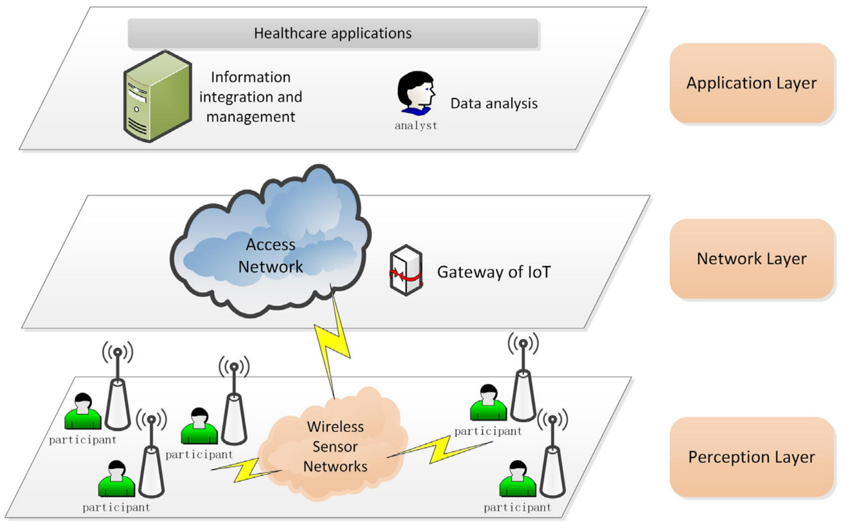


Fig. 1. A typical architecture for healthcare IoT systems.

interactive approach [19] emerged to resist the privacy leakage for the use of *k*-means algorithms. However, there is still a risk of privacy leakage because the cluster center in this method is public, especially when collusion occurs between participants and an analyst who knows the cluster center, i.e., co-conspirators can easily calculate the sensitive data of the other participants in the same cluster. In addition, the time efficiency of an algorithm is an essential factor for healthcare applications [20,21]. Most of existing privacy-preserving *k*-means algorithms cannot protect mutual privacy with low time complexity.

Therefore, it is necessary to develop an efficient strategy for *k*-means algorithms to protect mutual privacy information between participants and analysts. The main contributions of this paper can be summarized as follows.

- We analyze two privacy issues in each iteration of classic *k*-means algorithms and develop a novel privacy-preserving *k*-means strategy (M-PPKS) to carry out mutual privacy-preserving, which not only protects the privacy of participants but also avoids the leakage of cluster center information.
- The proposed M-PPKS introduces a third-party cloud plane to resist collusion attacks. Security analysis demonstrates that M-PPKS can effectively resist collusion attacks and has lower communication complexity.
- Extensive experimental results demonstrate that the proposed M-PPKS can achieve high precision, recall and F1-measure. It can efficiently obtain clustering results close to the classical *k*-means algorithm, without disclosing privacy information.

The rest of this paper is organized as follows. In Section 2, related work is introduced. Preliminaries are provided in Section 3. Section 4 presents the proposed M-PPKS. Section 5 analyzes the privacy and efficiency of M-PPKS. In Section 6, we conduct experiments and evaluate the performance of the proposed strategy. Finally, a conclusion is drawn in Section 7.

2. Related work

The research on the privacy-preserving of *k*-means algorithms has made great progress. Many privacy-preserving *k*-means algorithms have been proposed. Vaidya et al. [22] made the first study on privacy-preserving in *k*-means clustering algorithms. They group vertical data while reducing communication cost and ensuring reasonable data privacy-preserving. Jha et al. [23] considered privacy preservation in the process of calculating cluster centers. They proposed two privacy protection schemes, where one is based on the oblivious polynomial evaluation and the other is based on homomorphic encryption. However, the scheme did not consider the privacy leakage issues when assigning a participant into the nearest cluster center. Bunn et al. [24] proposed a two-party *k*-means clustering protocol, which computes the clustering results without using the cluster center in each iteration. This protocol avoids the leakage of cluster center during the clustering process. Xing et al. [19] proposed a mutual privacy protection scheme to prevent the disclosure of privacy data and effectively resist collusion attacks.

Differential privacy has attracted many interests in the privacy protection of *k*-means research. Blum et al. [25] introduced a differential privacy mechanism into the privacy-preserving algorithm for the first time. The proposed mechanism reduces the risk of privacy leakage by adding noise to the cluster center. However, the selection of initial center points directly affects the performance of *k*-means clustering. Therefore, an improved differential privacy (IDP) *k*-means algorithm was proposed by Yang et al. [26], to enable the random choice of the initial cluster center. As well as privacy-preserving based on homomorphic encryption and differential privacy, a novel blockchain-based data privacy protection scheme [27] was proposed, taking advantage of the blockchain infrastructure to remove the single-point-failure issue.

All the above studies have drawbacks so that they cannot be directly deployed in a time-sensitive system. Therefore, recent research has started paying close attention to the privacy-preserving clustering strategy with low time complexity. Yu et al. [28] proposed a novel privacy-preserving multi-party *k*-means clustering scheme, which is the first to use parallel computing techniques to improve the clustering process. The time complexity of their scheme is much lower than previous studies. Miao et al. [29] proposed a lightweight privacy-preserving framework based on truth discovery, L-PPTD and L2-PPTD, which were implemented by two cloud platforms to achieve less communication cost.

In summary, existing privacy-preserving *k*-means clustering schemes cannot protect the privacy of participants while simultaneously preserving the privacy of the cluster center. Moreover, most of the strategies cannot resist collusion attacks or have high time complexity. To solve these issues, in this paper, we propose a low time complexity strategy M-PPKS. The M-PPKS strategy is

Table 1
The notations used in this paper.

Notation	Description
a_i	The ith participant $(1 \le i \le n)$
$\mathbf{a}_i$	The feature vector of the participant a_i
C_j	The jth cluster $(1 \le j \le k)$
$\mathbf{c}_j$	The cluster center of the jth cluster
$E(\cdot)$	Encryption operation
$D(\cdot)$	Decryption operation
pk	Public key
pr	Private key
m	Plaintext

similar to the work of the literature [19], in preserving the private data of participants and cluster centers. The difference is that M-PPKS can achieve lower time complexity and better resistance to collusion attacks.

3. Preliminaries

In this section, we will introduce the classic k-means algorithm, followed by the homomorphic encryption algorithm. The notations used in this paper are summarized in Table 1.

3.1. K-means clustering algorithm

K-means [13] is a process of continuous iteration, which can cluster participants with similar features. Suppose there are n participants $\{a_1, a_2, \dots, a_n\}$ that should be grouped into k clusters $C = \{C_1, C_2, \dots, C_k\}$. Each participant is represented by a q-dimensional feature vector. The detailed process of k-means is summarized as follows:

(1) Randomly initialize $c = \{c_1, c_2, \dots, c_k\}$, where c_j represents the center of cluster C_j.
(2) Assign each participant a_i to its nearest cluster center c' according to Eq. (1). In this paper, we adopt the Euclidean distance as our default distance measure.

$$c' = \arg\min_{c_j \in c} \|a_i - c_j\| \ . \tag{1}$$

(3) Recalculate the new cluster center c_j based on Eq. (2), where c_j is the mean of participants in C_j.

$$c_j = \frac{1}{|C_j|} \sum_{a_i \in C_j} a_i \ . \tag{2}$$

(4) Repeat Steps (2) and (3) until no change happens or the stop criteria is satisfied.

3.2. Homomorphic encryption

Homomorphic encryption [30] is a method to delegate the processing of data, without disclosing the data. The encryption methods include Paillier cryptosystem and RSA cryptosystem [31]. For instance, Paillier cryptosystem allows a third-party cloud platform to perform addition operation over encrypted data (aka additively homomorphic). Eqs. (3) and (4) define the properties of Paillier cryptosystem.

$$E(m_1 + m_2) = E(m_1) * E(m_2). \tag{3}$$

$$E(s * m) = E(m)^s. \tag{4}$$

where $E(\cdot)$ is the encryption function and s is a constant.

In this paper, we adopt the Paillier cryptosystem to encrypt the private data. The encryption and decryption of Paillier cryptosystem are illustrated as follows.

Encryption: An entity generates a public key $pk(n, g)$ and a private key $pr = \lambda$. The message $m \in N^*$ can be encrypted as

$$E(m) = g^m r^n \bmod n^2, \tag{5}$$

where r is a random number that satisfies $r \in N^*$.

Decryption: The plaintext can be calculated by ciphertext $E(m)$ and private key λ :

$$D(E(m)) = \frac{L(E(m)^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n, \tag{6}$$

where $L(x) = \frac{x-1}{n}$. According to the properties of Paillier cryptosystem, the following equations can be derived:

$$D(E(m_1 + m_2)) = D(E(m_1) * E(m_2)) = m_1 + m_2, \tag{7}$$

$$D(E(s * m_3)) = D(E(m_3)^s) = s * m_3. \tag{8}$$

4. Mutual privacy-preserving k-means clustering algorithm (M-PPKS)

In this section, we will discuss the privacy issues of k-means algorithms and present the proposed M-PPKS strategy.

4.1. The privacy analysis of K-means Clustering

To begin with we will provide a brief privacy analysis on k-means clustering. In data analysis, an analyst A groups participants' information using the k-means algorithm. It is worth noticing that participants do not want to share their private data with the analyst, meanwhile the analyst is reluctant to let participants know the cluster center. To meet above constraints, we propose a M-PPKS strategy that can protect privacy between participants and analysts. As shown in Fig. 2, each iteration of a k-means algorithm consists of two stages: (1) finding the nearest cluster center for a participant and (2) computing a new center of each cluster. In the first stage, we ensure that the private data of a participant and the information of a cluster center is not obtained by each other. In the second stage, we introduce a third-party cloud platform to reduce the communication complexity. The homomorphic encryption is adopted to prevent participants' privacy from being accessed by the third-party cloud platform and analyst A. For M-PPKS, we assume that:

(1) Analyst A generates a public and private key pair (pk, pr). The public key is sent to all participants and the analyst keeps the private key.
(2) Each participant receives the public key pk from an analyst to encrypt its private data. The ciphertext can only be decrypted by the analyst using its private key pr.

4.2. The first stage of M-PPKS

In the first stage, we consider a strategy that ensure the private data of a participant and the information of a cluster center is not obtained by each other.

The traditional method of assigning each participant to their nearest cluster center by comparing the Euclidean distance between a cluster center and a participant. The implementation of this step is described as follows. The analyst A initializes k cluster centers. We can calculate the Euclidean distance D_{ij} between the participant's data $\mathbf{a}_i$ and cluster center $\mathbf{c}_j$ by Eq. (9). The Euclidean distance is used to measure the spatial distance between a participant and a cluster center. The smaller the distance, the more likely to be assigned to this cluster.

$$D_{ij} = (\mathbf{a}_i - \mathbf{c}_j)^T(\mathbf{a}_i - \mathbf{c}_j) \tag{9}$$

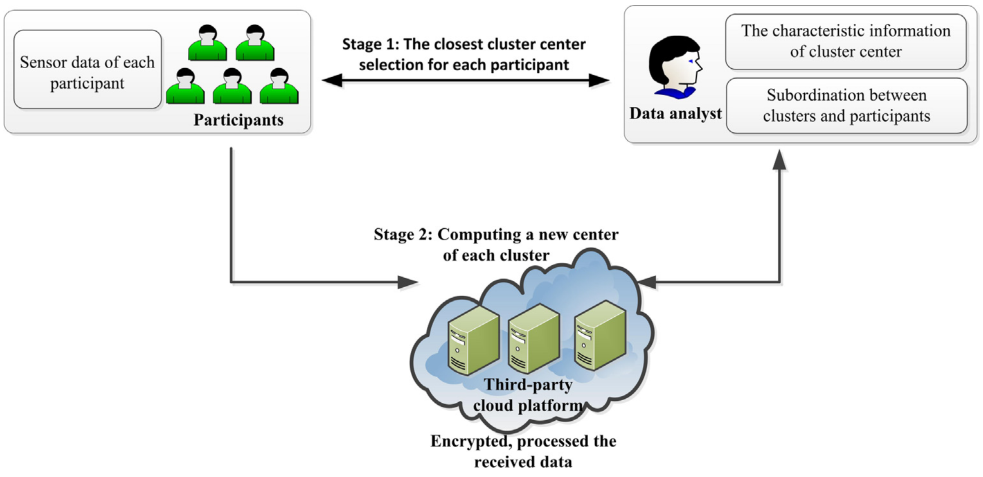

Sensor data of each participant
Participants
Stage 1: The closest cluster center selection for each participant
Data analyst
The characteristic information of cluster center
Subordination between clusters and participants
Stage 2: Computing a new center of each cluster
Third-party cloud platform
Encrypted, processed the received data

Fig. 2. A k-means clustering analysis model.

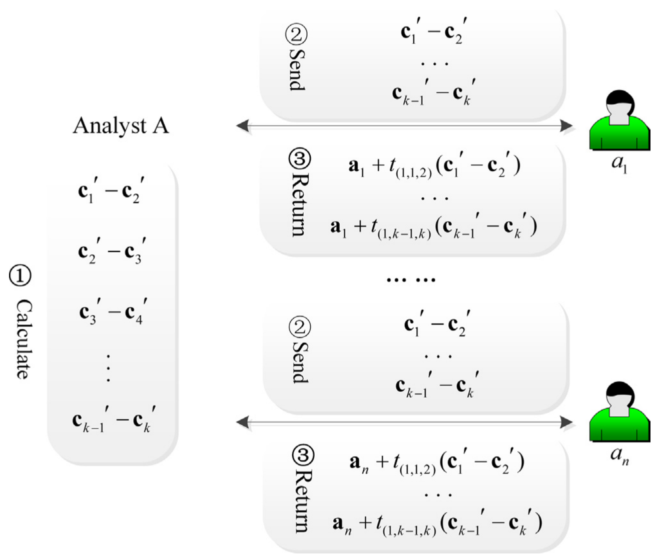


Fig. 3. The process of assigning each participant to their closest cluster center.

Then, we compare the distances of participant a_i to clusters C_j and $C_{j'}$. We have

$$\begin{aligned} &D_{ij}-D_{ij'} \\ &= \mathbf{c}_j{}^T\mathbf{c}_j-\mathbf{c}_{j'}{}^T\mathbf{c}_{j'}-\mathbf{a}_i{}^T(\mathbf{c}_j-\mathbf{c}_{j'})-(\mathbf{c}_j-\mathbf{c}_{j'})^T\mathbf{a}_i \end{aligned} \tag{10}$$

If $D_{ij} - D_{ij'} < 0$, the participant a_i belongs to the cluster C_j. Otherwise, a_i belongs to cluster $C_{j'}$. The closest cluster center to a_i can be identified by repeatedly running this process $(k - 1)$ times. The data analyst A can directly obtain the participants' private information when confirming which cluster center is closest, resulting in the leakage of participants' privacy. To solve this problem, we propose a privacy-preserving strategy to ensure that the privacy of a participant and the private information of a cluster center are not disclosed. This strategy is explained as follows.

The data analyst A calculates the value of $(\mathbf{c}_j{}' - \mathbf{c}_{j'}{}')$ that satisfies Eq. (11), and sends it to participant a_i. Noting that there are many solutions of $(\mathbf{c}_j{}' - \mathbf{c}_{j'}{}')$, we only need to calculate one solution.

$$(\mathbf{c}_j{}' - \mathbf{c}_{j'}{}')^T(\mathbf{c}_j - \mathbf{c}_{j'}) = 0, \left|\mathbf{c}_j{}' - \mathbf{c}_{j'}{}'\right| \neq 0. \tag{11}$$

After receiving the value of $\mathbf{c}_j{}' - \mathbf{c}_{j'}{}'$, the participant a_i can compute $\mathbf{a}_{(i,j,j')}$ as below

$$\mathbf{a}_{(i,j,j')} = \mathbf{a}_i + t_{(i,j,j')}(\mathbf{c}_j{}' - \mathbf{c}_{j'}{}'), \tag{12}$$

where $t_{(i,j,j')}$ is a random number used to disturb the value of $(\mathbf{c}_j{}' - \mathbf{c}_{j'}{}')$ and prevent $\mathbf{a}_i$ from being exposed.

The participants upload $\mathbf{a}_{(i,j,j')}$ to A, and then A calculates the value of $(D_{ij} - D_{ij'})$ according to Eq. (13).

$$\begin{aligned} &D_{ij}-D_{ij'} \\ &=(\mathbf{a}_{(i,j,j')}-\mathbf{c}_j)^T(\mathbf{a}_{(i,j,j')}-\mathbf{c}_j)-(\mathbf{a}_{(i,j,j')}-\mathbf{c}_{j'})^T(\mathbf{a}_{(i,j,j')}-\mathbf{c}_{j'}). \end{aligned} \tag{13}$$

If $D_{ij} - D_{ij'} < 0$, a_i is closer to the center $\mathbf{c}_j$. Then, we compare the distance from a_i to $\mathbf{c}_j$ and a_i to another cluster center. If $D_{ij} - D_{ij'} > 0$, a_i is closer to the center $\mathbf{c}_{j'}$. Then, we compare the distance from a_i to $\mathbf{c}_{j'}$ and a_i to another cluster center. Repeating this process $(k - 1)$ times can figure out the nearest center of a_i. By analyzing this process, we can find that a_i knows nothing about the private information of a cluster center and does not know which cluster it belongs to. The analyst A only knows which cluster a_i belongs, and A does not obtain any private data of a_i. Fig. 3 illustrates the process of assigning a participant to a cluster.

To verify whether the result of $(D_{ij} - D_{ij'})$ calculated by our strategy is the same as that calculated by the original k-means algorithm, we use $\mathbf{a}_{(i,j,j')}$ instead of $\mathbf{a}_i$ to determine the closest cluster center. The same result can be obtained when we use $\mathbf{a}_{(i,j,j')}$ and $\mathbf{a}_i$ to calculate $(D_{ij} - D_{ij'})$, respectively. The detailed verification process can be represented by Eq. (14) according to Eqs. (10)–(13):

$$\begin{aligned} &D_{ij}-D_{ij'} \\ &= \mathbf{c}_j{}^T\mathbf{c}_j-\mathbf{c}_{j'}{}^T\mathbf{c}_{j'}-(\mathbf{a}_i+t_{(i,j,j')}(\mathbf{c}_j{}'-\mathbf{c}_{j'}{}'))^T(\mathbf{c}_j-\mathbf{c}_{j'}) \\ &\quad -(\mathbf{c}_j-\mathbf{c}_{j'})^T(\mathbf{a}_i+t_{(i,j,j')}(\mathbf{c}_j{}'-\mathbf{c}_{j'}{}')) \\ &= (\mathbf{a}_i-\mathbf{c}_j)^T(\mathbf{a}_i-\mathbf{c}_j)-(\mathbf{a}_i-\mathbf{c}_{j'})^T(\mathbf{a}_i-\mathbf{c}_{j'}) \end{aligned} \tag{14}$$

The process of selecting the closest cluster center for each participant is summarized by Algorithm 1.

4.3. The second stage of M-PPKS

In the second stage, we consider a strategy to prevent participants' privacy from being accessed by the third-party cloud platform and analyst A. Similar to the first stage, there are n participants $\{a_1, a_2, \ldots, a_n\}$ and k clusters $\{C_1, C_2, \ldots, C_k\}$. Table 2 shows the relationship between clusters and participants. This relationship can be obtained after participants select the closest cluster center, where $|C_j|$ is the number of participants in cluster

Algorithm 1: Finding the Nearest Cluster Center for Each Participant

```
Input: Participants' information {$\mathbf{a}_1$,...,$\mathbf{a}_n$};The number of clusters $k$.
Output: The closest cluster center of each participant
1  Initialize: cluster centers {$\mathbf{c}_1$,$\mathbf{c}_2$,...,$\mathbf{c}_k$};
2  for j=1 to $k-1$ do
3      $\gamma = j+1$;
4      for $\gamma$ to $k$ do
5          $A$ seeks one random solution of $\mathbf{c}_j{}' - \mathbf{c}_\gamma{}'$ that satisfies Eq. (11);
6      end
7  end
8  for i=1 to $n$ do
9      for j=1 to $k-1$ do
10         $\gamma = j+1$;
11         for $\gamma$ to $k$ do
12             $\mathbf{a}_{(i,j,\gamma)} = \mathbf{a}_i + t_{(j,\gamma)}(\mathbf{c}_j{}' - \mathbf{c}_\gamma{}')$; $a_i$ sends $\mathbf{a}_{(i,j,\gamma)}$ to $A$;
13         end
14     end
15 end
16 for i=1 to $n-1$ do
17     $A$ confirms the nearest center of $a_i$ follows Eq. (13);
18 end
```

Table 2
The relationship between clusters and participants.

Clusters	Participants
$C_1(\lvert C_1\rvert$ participants)	$s_1^1, s_1^2, \ldots, s_1^{\lvert C_1\rvert}$
$C_2(\lvert C_2\rvert$ participants)	$s_2^1, s_2^2, \ldots, s_2^{\lvert C_2\rvert}$
$C_k(\lvert C_k\rvert$ participants)	$s_k^1, s_k^2, \ldots, s_k^{\lvert C_k\rvert}$

C_j, and $\sum_{j=1}^{k} |C_j| = n$. $s_j^l \in \{a_1, a_2, \ldots, a_n\}$ denotes the lth participant in the jth cluster, where $1 \le l \le$ n and $1 \le j \le k$.

In this stage, we need a strategy to ensure that the analyst cannot obtain participants' privacy information. Considering the existing privacy-preserving algorithms, such as [28,32–34] consist of high communication complexity, and the data protection in the interaction process is not being accounted for. Therefore, we propose a homomorphic encryption-based strategy that prevents participants' private information from being leaked to the analyst and malicious attackers. Also, to achieve lower latency, a non-colluding third-party cloud platform P_T is involved in our strategy. The detailed calculation process is mainly composed of two phases as shown below.

4.3.1. The first phase

This phase will be conducted only once during computing a new center for each cluster. Analyst A randomly generates a public key and private key pair (pk, pr). The public key is issued to all participants and the third-party cloud platform, while the private key is only kept by analyst A. Fig. 4 shows the detailed process of this phase. The specific steps are described below.

(1) Each participant generates a random q-dimensional vector $\boldsymbol{\rho}_i$ that is used to interfere with the private data of each participant. The random vector $\boldsymbol{\rho}_i$ satisfies $\| \boldsymbol{\rho}_i \| < \| \mathbf{a}_i \|$. Then, the perturbed data $\mathbf{a}_i{}'$ can be denoted by Eq. (15)

$$\mathbf{a}_i{}' = \mathbf{a}_i - \boldsymbol{\rho}_i. \tag{15}$$

(2) Participants upload all perturbed data to the third-party cloud platform.

(3) The cloud platform encrypts perturbed data with the public key pk, and temporarily save ciphertext $E(\mathbf{a}_i{}')$.

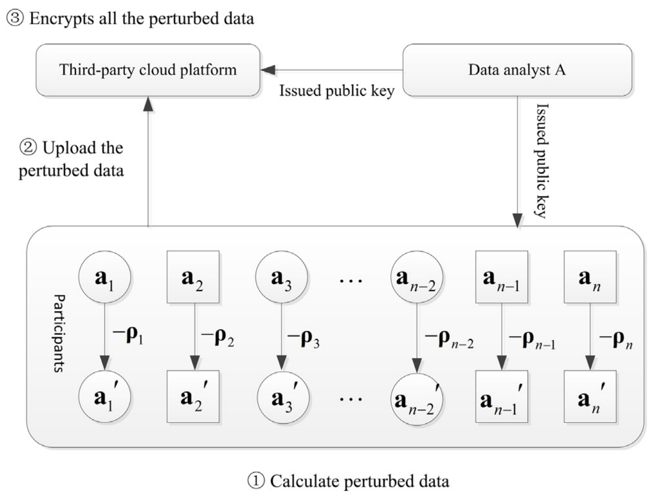

③ Encrypts all the perturbed data
Third-party cloud platform
Issued public key
Data analyst A
Issued public key
② Upload the perturbed data
Participants
$\mathbf{a}_1$ $\mathbf{a}_2$ $\mathbf{a}_3$ … $\mathbf{a}_{n-2}$ $\mathbf{a}_{n-1}$ $\mathbf{a}_n$
$-\boldsymbol{\rho}_1$ $-\boldsymbol{\rho}_2$ $-\boldsymbol{\rho}_3$ $-\boldsymbol{\rho}_{n-2}$ $-\boldsymbol{\rho}_{n-1}$ $-\boldsymbol{\rho}_n$
$\mathbf{a}_1{}'$ $\mathbf{a}_2{}'$ $\mathbf{a}_3{}'$ … $\mathbf{a}_{n-2}{}'$ $\mathbf{a}_{n-1}{}'$ $\mathbf{a}_n{}'$
① Calculate perturbed data

Fig. 4. The first phase of computing a new center for each cluster.

4.3.2. The second phase

In each iteration, the second stage will be repeated multiple times which equal to the number of clusters. Analyst A will share the relationship table of participants and clusters with the third-party cloud platform. For ease of understanding, we take the process of calculating cluster C_1 as an example. The process of this phase is shown in Fig. 5 and the following steps are taken.

(1) In the first place, the third-party cloud platform P_T first sends $R = \{r_1^1, \ldots, r_1^{|c_1|}, \ldots, r_j^1, \ldots, r_j^{|c_j|}\}$, $V = \{\mathbf{v}_1^1, \ldots, v_1^{|c_1|}, \ldots, v_j^1, \ldots, v_j^{|c_j|}\}$ to participants securely. According to the relation table, the value of r can take constant 0 or 1. There have $R_1 = R_1 \cup R_1'$, $R_1 = \{r_1^1, r_1^2, \ldots, r_1^{|c_1|}\}$ meets $\prod_{i=1}^{|c_i|} r_1^i = 1$, and $R_1' = \{r_2^1, \ldots, r_2^{|c_2|}, \ldots, r_j^1, \ldots, r_j^{|c_j|}\}$ satisfies $\sum_{i=2}^{j} \sum_{k=1}^{c_i} r_i^{|k|} = 0$. There have $V = \{\mathbf{v}_1^1, \ldots, \mathbf{v}_1^{|c_1|}, \ldots, \mathbf{v}_j^1, \ldots, \mathbf{v}_j^{|c_j||}\}$ satisfies $\sum_{i=2}^{j} \sum_{k=1}^{c_i} \mathbf{v}_i^{|k|} = 0$.

(2) After all the participants receive r and $\mathbf{v}$, each participant calculates the value of $r \times \boldsymbol{\rho} + \mathbf{v}$, and use public key to encrypt. The encrypted data of each participant as follows:

$$\mathbf{Y} = E\left(pk, r \times \boldsymbol{\rho} + \mathbf{v}\right). \tag{16}$$

Each participant will then be required to share slice of itself private data with other participants. We assume that a participant randomly divides the ciphertext $\mathbf{Y}$ into m $(1 \le m \le n)$ slices, and selects $m-1$ slices to other participants. The m slices satisfy $\prod_{i=1}^{m} \mathbf{Y}^i = \mathbf{Y}$. The key idea of slicing actions is to interrupt the random data $\boldsymbol{\rho}$ generated by each participant, by not changing the sum of them. After slices are distributed, each participant will get a new data $\boldsymbol{\rho}'$ composed of slices from other participants.

(3) After slicing, each participant sends $\boldsymbol{\rho}'$ to the third-party cloud platform. We can calculate the sum of random data generated by participants within the same cluster. The sum of random data in the cluster C_1 can be denoted by Eq. (17).

$$\begin{aligned} &\boldsymbol{\rho}_{1,1}{}' + \ldots + \boldsymbol{\rho}_{1,|c_1|}{}' \ldots + \boldsymbol{\rho}_{j,1}{}' + \ldots + \boldsymbol{\rho}_{j,|c_j|}{}' \\ &= E(pk, \textstyle\sum_{i=1}^{j} \sum_{k=1}^{|c_i|} \sum_{t=1}^{m_i} \mathbf{Y}_{i,k}^t) \\ &= E(pk, \textstyle\sum_{k=1}^{|c_1|} \sum_{t=1}^{m_k} \mathbf{Y}_{1,k}^t) \\ &= E(pk, \textstyle\sum_{k=1}^{|c_k|} \boldsymbol{\rho}_{1,k}), \end{aligned} \tag{17}$$

where $\boldsymbol{\rho}_{j,l}{}'$ denotes a newly composed data in the lth $(1 \le l \le$ n) participant of the jth $(1 \le j \le k)$ cluster. The set of

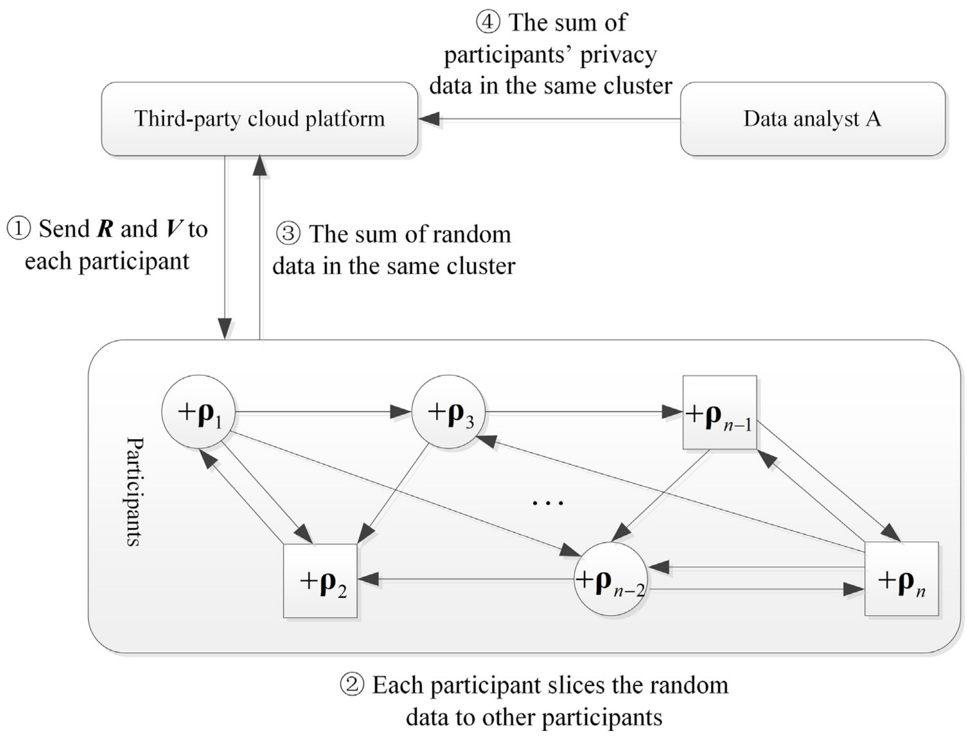

④ The sum of participants' privacy data in the same cluster
Third-party cloud platform
Data analyst A
① Send R and V to each participant
③ The sum of random data in the same cluster
Participants
+ρ1
+ρ3
+ρn−1
+ρ2
+ρn−2
+ρn
...
② Each participant slices the random data to other participants

Fig. 5. The second phase of computing a new center of each cluster.

$\{\mathbf{Y}^1_{l,j}, \mathbf{Y}^2_{l,j}, \ldots, \mathbf{Y}^m_{l,j}\}$ represents the received slices from other participants and the remaining slice saved by itself.

(4) Combining the encrypted perturbed data in the first phase, the third-party cloud platform P_T can compute the sum of participants in the same cluster by Eq. (18). Then, P_T uploads $E\left(\mathbf{a}_{1,1} + \cdots + \mathbf{a}_{1,|c_1|}\right)$ to analyst A who decrypts the ciphertext using the private key pr.

$$\begin{aligned}
&\prod_{i=1}^{|c_1|} E(\mathbf{a}_{1,i}{}') \cdot [E(\textstyle\sum_{i=1}^{|c_1|} \boldsymbol{\rho}_{1,i})] \\
&= E(\textstyle\sum_{i=1}^{|c_1|} \mathbf{a}_{1,i} + \sum_{i=1}^{|c_1|} \boldsymbol{\rho}_{1,i}) \\
&= E[\textstyle\sum_{i=1}^{|c_1|}(\mathbf{a}_{1,i} - \boldsymbol{\rho}_{1,i}) + \sum_{i=1}^{|c_1|} \boldsymbol{\rho}_{1,i}] \\
&= E(\textstyle\sum_{i=1}^{|c_1|} \mathbf{a}_{1,i})
\end{aligned} \tag{18}$$

The analyst A repeats the second phase until all the clusters are traversed. The whole procedure is summarized in Algorithm 2.

5. Performance evaluation

In this section, we analyze the effectiveness of M-PPKS in protecting the mutual privacy between participants and analysts, and the ability to resist collusion attacks. To better understand the performance analysis process, the relevant assumptions are given below.

We assume that: (1) participants cannot access the private information of a cluster center; (2) they know nothing about which cluster they belong to; (3) they do not know who is in the same cluster; (4) they cannot access the private information about other participants.

And then, we assume that: (1) analyst A knows the information of a cluster center, and the affiliation between clusters and participants; (2) it cannot access the private information of participants.

In addition, we also consider the following assumptions: (1) the number of participants satisfy $n > 3$; (2) the third-party cloud platform is semi-honest and will not collude with others.

Algorithm 2: Computing a New Center of Each Cluster

```
   Input: The private information $\mathbf{a}_t^l$ of each participant.
   Output: The cluster center of each cluster
1  Initialize: The number of clusters is $k$, each cluster $C_t$
   has the member participants $s_t^1, s_t^2, ..., s_t^{|c_t|}$, $\mathbf{a}_t^l$ is the
   private data of the member $s_t^l$. Analyst $A$ and
   third-party cloud platform $P_T$ share the relationship
   table;
2  Each participant computes and sends the perturbed
   data $\mathbf{a}_i{}'$ to the third-party cloud platform;
3  The third cloud server encrypts all the perturbed data;
4  for $t$=1 to $k$ do
5      The third-party cloud server platform sends $R_t$ and
       $V_t$ to participants within cluster $C_t$ and sends $R_{t'}$
       and $V_{t'}$ to other participants;
6      Participants compute the encrypted data according
       to Eq. (16);
7      Each participant slices the ciphertext into $m$
       components, and randomly select $m-1$
       components to others.;
8      Each participant calculated $\boldsymbol{\rho}'$ by multiplying all the
       received components and the component itself
       kept;
9      Each participant sends $\boldsymbol{\rho}'$ to the $P_T$;
10     The $P_T$ calculates the difference between $\boldsymbol{\rho}'$ and $\boldsymbol{\rho}'$ ,
       then send it to analyst $A$;
11     The sum of the participant's private information
       $\mathbf{a}_t^1 + \mathbf{a}_t^2 + ... + \mathbf{a}_t^{|c_t|}$ can be obtained via decryption.
       The new center of the cluster $C_t$ can be calculated
       by $\frac{\left(\mathbf{a}_t^1+\mathbf{a}_t^2+...+\mathbf{a}_t^{|c_t|}\right)}{|c_t|}$.
12 end
```

5.1. Privacy analysis on finding the nearest cluster center for a participant

In the stage of finding the nearest cluster center for a participant, a data analyst only knows which cluster is closest to a participant, however, it knows nothing about the participants' information. Participants only know private information about themselves. Let us consider n participants $\{a_1, a_2, \ldots, a_n\}$ and k cluster centers $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$. According to the received data from data analyst A, n participants can build Eq. (19).

For participants, they cannot accurately calculate the value of $\{\mathbf{c}_1 - \mathbf{c}_2, \mathbf{c}_1 - \mathbf{c}_3\ , \ldots, \mathbf{c}_2 - \mathbf{c}_3, \ldots, \mathbf{c}_{k-1} - \mathbf{c}_k\}$. Because there are multi-value of $(\mathbf{c}_j{}' - \mathbf{c}_{j'}{}')$ according to Eq. (11). For the analyst A, all the received information from participants can be constructed in Eq. (20).

$$a_1, a_2, \ldots, a_n \textit{ received} \begin{cases} \mathbf{c}_1{}' - \mathbf{c}_2{}' \\ \mathbf{c}_1{}' - \mathbf{c}_3{}' \\ \ldots, \\ \mathbf{c}_2{}' - \mathbf{c}_3{}' \\ \ldots, \\ \mathbf{c}_{k-1}{}' - \mathbf{c}_k{}' \end{cases} \tag{19}$$

$$A \textit{ received} \begin{cases} \mathbf{a}_1 + t_{(1,1,2)} * (\mathbf{c}_1{}' - \mathbf{c}_2{}') \\ \mathbf{a}_1 + t_{(1,1,3)} * (\mathbf{c}_1{}' - \mathbf{c}_3{}') \\ \ldots, \\ \mathbf{a}_1 + t_{(1,2,3)} * (\mathbf{c}_2{}' - \mathbf{c}_3{}') \\ \mathbf{a}_1 + t_{(1,2,4)} * (\mathbf{c}_2{}' - \mathbf{c}_4{}') \\ \ldots, \\ \mathbf{a}_1 + t_{(1,k-1,k)} * (\mathbf{c}_{k-1}{}' - \mathbf{c}_k{}') \\ \ldots, \\ \mathbf{a}_n + t_{(n,1,2)} * (\mathbf{c}_1{}' - \mathbf{c}_2{}') \\ \mathbf{a}_n + t_{(n,1,3)} * (\mathbf{c}_1{}' - \mathbf{c}_3{}') \\ \ldots, \\ \mathbf{a}_n + t_{(n,k-1,k)} * (\mathbf{c}_{k-1}{}' - \mathbf{c}_k{}') \end{cases} \tag{20}$$

The coefficients $\{t_{(1,1,2)}, t_{(1,1,3)}, \ldots, t_{(1,2,3)}, t_{(1,2,4)}, \ldots, t_{(1,k-1,k)}, \ldots, t_{(n,1,2)}, t_{(n,1,3)}, \ldots, t_{(n,k-1,k)}\}$ and $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ are unknown, and this is generated from participants. There consists of $n + n * \frac{k(k-1)}{2}$ unknown parameters in Eq. (20), but there is only $n * \frac{k(k-1)}{2}$ equations. Therefore, the analyst cannot calculate the private information of $a_1, a_2, \ldots, a_n$ through existing knowledge. Now we consider the worst-case scenario when n participants collude with each other. In other words, participants know the value of $\left\{\mathbf{c}_1{}' - \mathbf{c}_2{}', \mathbf{c}_1{}' - \mathbf{c}_3{}', \ldots, \mathbf{c}_{k-1}{}' - \mathbf{c}_k{}'\right\}$ and the private information of other participants. On one hand, even if n participants collude, the information of $\{\mathbf{c}_1 - \mathbf{c}_2, \mathbf{c}_1 - \mathbf{c}_3, \ldots, \mathbf{c}_2 - \mathbf{c}_3, \ldots, \mathbf{c}_{k-1} - \mathbf{c}_k\}$ will not be leaked. On the other hand, although a participant knows the private information of other participants, participants cannot detect the cluster center because it does not know which cluster it belongs to and which participants are within the same cluster. Thus, we can conclude that our scheme can resist the collusion attacks initiated by the largest n participants without leaking any information about the cluster center, and the analyst A cannot know any private information about participants.

In the following, we analyze the cost of finding the nearest cluster center for each participant. Due to the uncertainty of the size of data sets and the cluster center, the communication cost of our algorithm is also uncertain. Removing the above influencing factors, a simple communication cost analysis for our proposed algorithm is presented as follows. We suppose there are n participants, therefore, each participant should make $O(k^2 + 1)$ comparisons to calculate the closest cluster center. The total communication cost is therefore $O(nk^2 + n)$, which has higher cost than that of the algorithm in [19].

5.2. Privacy analysis on computing the new cluster center

In the stage of computing a new center of each cluster, we adopt the homomorphic encryption and the slicing method to protect participants' private information from being disclosed to the analyst or external attackers. When a collusion attack or external attacks occur among the analyst and participants, our scheme can achieve effective privacy preservation.

First, we perform a privacy analysis on the participant's information stored by the third-party cloud platform. This platform only knows the relationship between participants and clusters, the plaintext $\mathbf{a}_i{}'$, the ciphertext $E\left(\mathbf{a}_i{}'\right)$ and the newly composed data after slicing $\boldsymbol{\rho}_i{}'$. Since the private key is only known by the analyst A, third-party cloud platforms cannot decrypt the ciphertext. $\mathbf{a}_i$ cannot be calculated by the third-party cloud platform thanks to that the plaintext $\mathbf{a}_i{}'$ contains a random number generated by participants. Thus, even if $\mathbf{a}_i{}'$, $E\left(\mathbf{a}_i{}'\right)$ and $\boldsymbol{\rho}_i{}'$ are known by the third-party cloud platform, it cannot learn anything about the participants' private information.

Then, we will conduct the privacy analysis on the side of participants. In addition to their information and the generated random number $\boldsymbol{\rho}$, each participant knows the ciphertext of slices distributed by other participants. Based on this, the sum of the slices received from other participants and the partial slices saved by itself can be calculated. However, each participant cannot learn anything about the private information of other participants and the cluster center to which the other participants belong.

Analyst A knows nothing about the participant's private information except for the aggregated results of participants in the same cluster. In short, the private information of each participant will not be disclosed to the third-party cloud platform and the analyst. Moreover, the private information about the cluster center will not be disclosed to the participants and the third-party cloud platform.

In the following, we analyze the cost of computing a new cluster center. In the second stage, we observe that encryption and decryption operations are done by the third-party cloud platform and analyst respectively. Since the cost of encryption and decryption operations is largely determined by the processing capabilities of the third-party cloud platform and the analyst, we only consider the communication complexity of participants. The participants need to do the following calculations: generating a random q-dimensional vector to calculate perturbed data, slicing and allocating data. In Algorithm 2, line 2 takes $O(n)$ time to calculate perturbed data, line 6 takes $O(n)$ time, line 7 takes $O(q * n)$ time, and line 8 and line 9 need $O(n)$ time. Similar iterations require k times. Therefore, the sum of communication cost for all participants is $O(kqn)$, which has lower communication complexity than that of the algorithm in [19].

5.3. Privacy protection against collusion attacks

In this section, we analyze the ability of M-PPKS to resist collusion attacks in different collusion scenarios, and provide a comparison of collusion resistance with other strategies.

Scenario 1. The collusion between a data analyst and participants. According to Algorithm 2, no further information can be inferred, since data analyst A can only receive the aggregated encrypted results of participants within the same cluster. Consider the scenario that data analyst A colludes with other participants. We can divide the scene into the following two stages: before and after an analyst calculates the new cluster center.

Before analyst A calculates a new cluster center, if analyst A colludes with $t(t < n - 1)$ participants, A can obtain private information of t participants. Since the new cluster center is not calculated, A cannot infer the information of the remaining

Table 3
Comparison of collusion resistance between our strategy, Ref. [19] and the classical k-means algorithm.

The worst collusion case that can infer the private information of all participants	M-PPKS	[19]	Classical k-means
Number of participants colluding with analyst	$n-1$	$n-1$	$n-k$
Number of participants colluding with each other	n	$n-1$	n

participants. Therefore, before analyst A calculates a new cluster center, the collusion of $n-1$ participants with analyst A cannot infer the private information of the remaining participants.

After analyst A calculates the cluster center, the best case is that as many as $n-2$ participants who collude with the analyst cannot infer the remaining two participants, and the remaining two participants must be in the same cluster. The worst-case scenario is when as many as $\frac{n}{2}-1$ participants are colluding with the analyst, the analyst cannot infer the information of the remaining $\frac{n}{2}+1$ participants.

Scenario 2. Collusion among the participants. If there are $n-1$ participants colluding together, it still cannot infer any information about the cluster center, because the $n-1$ participants in the conspiracy do not know the cluster to which the remaining participants belong. Thus, we claim that our scheme can effectively protect the privacy of the data analyst.

At last, we make a comparison of collusion resistance between our strategy, [19] and the classical k-means algorithm, as shown in Table 3. From this table, we can notice the significant advantages of our strategy compared to the other two strategies in terms of collusion attack resistance.

6. Experimental analysis

The simulation experiments are conducted with Python on a computer with the configurations of Intel Core i5-8250U 1.6 GHZ CPU and 8 GB RAM. To show the effectiveness of our strategy, we use three common datasets to conduct the following experiments to verify the accuracy compared between M-PPKS and the classical k-means algorithm.

6.1. Dataset

The first dataset includes the locations of public utilities in a city [35]. It records about 240 utilities' latitude and longitude data, users can figure out the public utility distribution from the results of our proposed clustering algorithm. The second dataset is about the survival of patients who had undergone surgery for breast cancer [36]. This dataset is collected from 306 patient instances. The clustering results can provide advice for the doctor to treat breast cancer. The last dataset is a real smartphone dataset for human activity recognition in ambient assisted living that is an addition to the dataset of [37]. This dataset contains 3-axial linear acceleration and 3-axial angular velocity of six activities (standing, sitting, lying, walking, walking upstairs, walking downstairs) that were collected from 30 participants within the age group of 22-79 years old. Each person carries a smart-phone with sensors to measure linear acceleration and angular velocity. Table 4 shows the detailed information about the three datasets.

6.2. Metrics

In this section, we use the precision, recall rates and F1-Measure to evaluate the performance of the proposed scheme. The Haberman's survival data set classification is taken as an example to describe how to calculate the evaluation indicator.

Table 4
Dataset information.

Dataset	Dims	Type	Tuples	Clusters
Public utilities dataset	2	real	240	4
Haberman's survival dataset	3	real	306	2
Human activity recognition dataset	561	real	5744	6

Table 5
Notations and descriptions of HABERMAN's survival dataset classification.

		Clustering results based on the M-PPKS	
		The patient survived 5 years or longer	The patient died within 5 years
Clustering results based on the classical k-means algorithm	The patient survived 5 years or longer	A	B
	The patient died within 5 years	C	D

As shown in Table 5, the horizontal rows denote the clustering results (the patient survived 5 years or longer, or the patient died within 5 years) based on the M-PPKS strategy, and the vertical columns represent the classification results (the patient survived 5 years or longer, or the patient died within 5 years) calculated by the classical k-means algorithm. The precision and recall of the patients survived 5 years or longer can be represented by:

$$Precision(Survived) = \frac{A}{A+C} \tag{21}$$

$$Recall(Survived) = \frac{A}{A+B} \tag{22}$$

In the process of evaluating the model, it is not comprehensive to evaluate the model by using the precision and recall only. Therefore, we combine the accuracy and the recall rate to obtain the F1-Measure value as another evaluation indicator. It can be denoted by:

$$F1(Survived)=\frac{2*Precision(Survived)*Recall(Survived)}{Precision(Survived)+Recall(Survived)} \tag{23}$$

Our experiments utilize the above three metrics to evaluate the clustering results. The larger F1-Measure becomes, the greater the similarity between our clustering results as well as the clustering results provided by the classical k-means algorithm. In the following section, the values of precision, recall and F1-Measure in different datasets will be given to determine the usability and accuracy of clustering results.

6.3. Experimental results

Take the public utilities data set as an example, the clustering results obtained from our proposed strategy are shown in Fig. 6. Different colors represent different cluster partitions. A data analyst can calculate the popular location of each cluster without disclosing the private location. Fig. 7 shows the cluster center for both our scheme and the original k-means algorithm. From this figure, we can see that the clustering results of our scheme are very close to the results obtained by the original k-means algorithm. The reason for the difference is that the plaintext encryption must be an integer, causing some accuracy loss of data in the encryption process. Table 6 shows the comparison of the

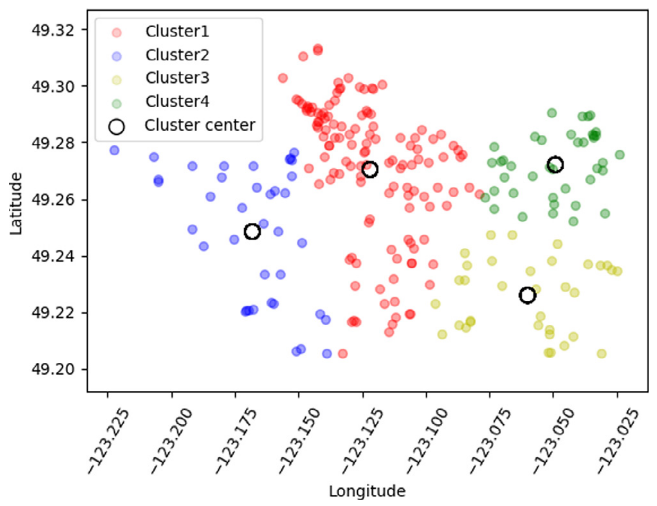


Fig. 6. Clustering result of our algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

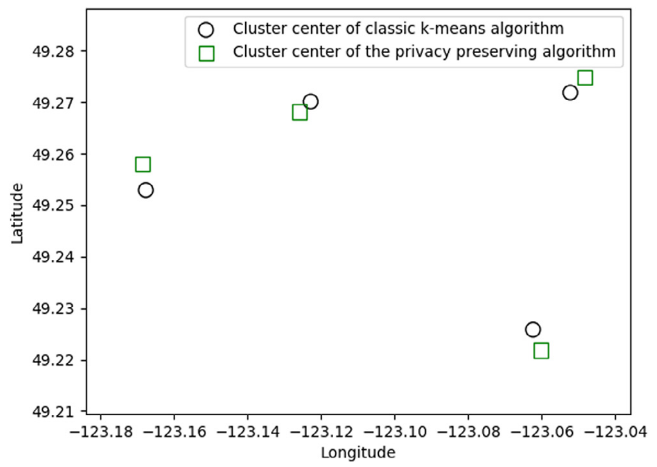


Fig. 7. Comparison of the clustering center of our algorithm and the classical *k*-means algorithm clustering center.

Table 6
The precision, recall and F1-measure of our algorithm compared to the classical *k*-means method.

	Precision	Recall	F1-measure
Cluster 1	96.77%	96.77%	96.77%
Cluster 2	98.51%	97.06%	97.78%
Cluster 3	98.41%	98.41%	98.41%
Cluster 4	97.47%	98.72%	98.09%

precision, recall and F1-measure results obtained by our strategy and the *k*-means clustering results which are computed from the unencrypted private data. The experiments are calculated using Eqs. (21)–(23) which are designed to show the comparison results between M-PPKS and the classical k-means algorithm. We can conclude that our strategy can get accurate clustering results without disclosing the public utilities location information.

In the second experiment, we select the Haberman's survival results as our dataset. We use M-PPKS to cluster the data set.

Table 7
The precision, recall and F1-measure of our algorithm compared to the classical *k*-means method.

	Precision	Recall	F1-measure
The patient survived 5 years or longer	97.52%	96.72%	97.12%
The patient died within 5 years	97.84%	98.37%	98.10%

Table 8
The precision, recall and F-measures of our algorithm compared to the classical *k*-means method.

	Precision	Recall	F-measure
Walking	97.33%	95.17%	96.24%
Standing	92.78%	96.77%	94.73%
Laying	96.91%	93.79%	95.32%
Walking upstairs	97.01%	93.03%	94.98%
Walking downstairs	95.43%	93.64%	94.53%

The clustering results are shown in Fig. 8. The results show the effect of age, patient's year of operation, and the positive axillary nodes' number detected on the patient's recovery. Analysts can cluster datasets without obtaining patient private information, and provide the clustering results to the doctor for effective reference. Fig. 9 shows the comparison of the cluster centers calculated by our proposed algorithm with the cluster centers calculated by the classical *k*-means algorithm. We can conclude the accuracy of the M-PPKS is almost the same as the *k*-means algorithm. The reason for the difference in clustering centers is that the randomness of the initial clustering center selection causes inconsistent clustering results. Table 7 shows the precision, recall and F1-measure of the clustering results of our strategy compared with the classical *k*-means clustering algorithm. The experimental results are calculated following the same way used for Table 6. We can see that our clustering strategy can obtain clustering results that are almost similar to the classic *k*-means algorithm without leaking private information.

The last dataset is a collection of smart-phone sensor data for different human activities. Table 8 shows the comparison clustering results between M-PPKS and classical *k*-means algorithm. The experimental results are calculated following the same way used for Table 6. The results demonstrate that our algorithm does not have a significant impact on the final clustering results while protecting participants' data privacy.

7. Conclusion

In this paper, we have proposed a mutual strategy M-PPKS that can protect the privacy information of participants from being exposed to the analyst and the private information of a cluster center from being disclosed to the participants. Moreover, considering the time-sensitivity nature of the healthcare IoT systems, a third-party cloud platform has been introduced in M-PPKS to reduce the overhead of participants. A comprehensive analysis of the privacy protection capabilities under normal circumstances and collusion attacks has been conducted, demonstrating that the proposed M-PPKS can achieve effective privacy protection. Extensive simulation results under three different datasets have shown that our scheme can achieve better accuracy. In future work, we plan to adopt differential privacy to design a new privacy protection k-means algorithm and further reduce communication complexity.

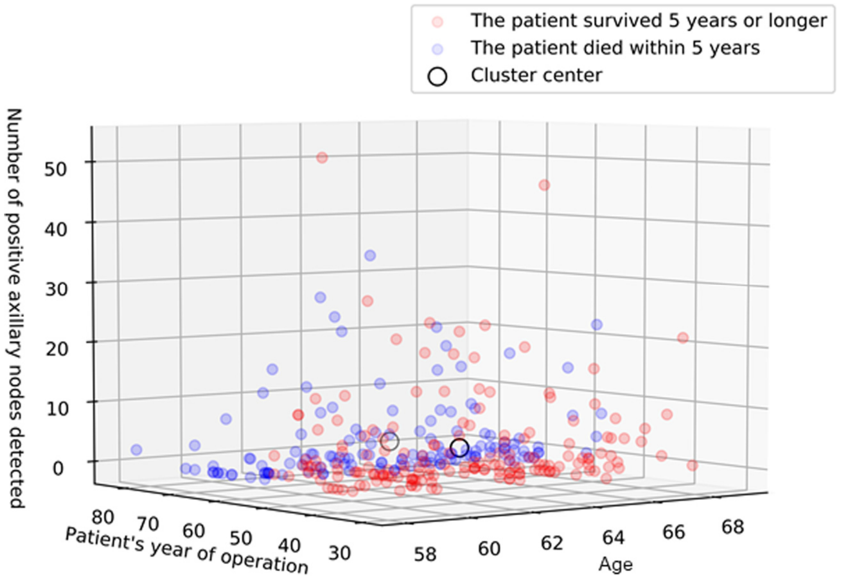


Fig. 8. Clustering result of our algorithm.

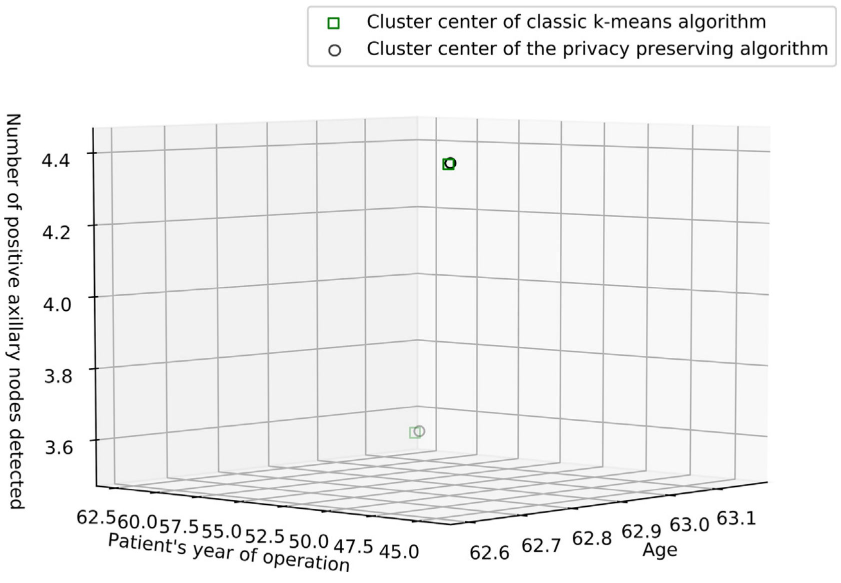


Fig. 9. Comparison of the cluster center of our algorithm and the classical *k*-means algorithm clustering center.

CRediT authorship contribution statement

Xuancheng Guo: Methodology, Software, Visualization, Writing - review & editing. **Hui Lin:** Writing - original draft, Supervision, Methodology, Project administration. **Yulei Wu:** Conceptualization, Investigation, Validation, Formal analysis. **Min Peng:** Writing - review & editing, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61772008) and the Joint Funds of the National Natural Science Foundation of China (U1405255).

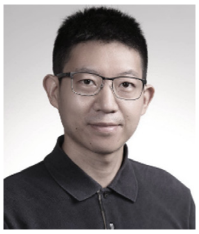
References

[1] L. Jie, Y. Wei, Z. Nan, X. Yang, Z. Wei, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet Things J. 4 (5) (2017) 1125–1142.
[2] X. Guo, H. Lin, Z. Li, M. Peng, Deep Reinforcement Learning Based QoS-Aware Secure Routing for SDN-IoT, IEEE, 2019, p. 1, http://dx.doi.org/10.1109/JIoT.2019.2960033.
[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, IEEE Internet Things J. 1 (1) (2014) 22–32.
[4] S. Talari, M. Shafie-Khah, P. Siano, V. Loia, A. Tommasetti, J.P. Catalão, A review of smart cities based on the internet of things concept, Energies 10 (4) (2017) 421.
[5] B.L.R. Stojkoska, K.V. Trivodaliev, A review of internet of things for smart home: Challenges and solutions, J. Cleaner Prod. 140 (2017) 1454–1464.
[6] S. Feng, P. Setoodeh, S. Haykin, Smart home: Cognitive interactive people-centric internet of things, IEEE Commun. Mag. 55 (2) (2017) 34–39.
[7] V.C. Gungor, D. Sahin, T. Kocak, S. Ergut, G.P. Hancke, Smart grid technologies: Communication technologies and standards, IEEE Trans. Ind. Inf. 7 (4) (2011) 529–539.
[8] S.S. Reka, T. Dragicevic, Future effectual role of energy delivery: A comprehensive review of internet of things and smart grid, Renew. Sustain. Energy Rev. 91 (2018) 90–108.

[9] L. Qi, Research on intelligent transportation system technologies and applications, in: 2008 Workshop on Power Electronics and Intelligent Transportation System, IEEE, 2008, pp. 529–531.
[10] P. Gope, T. Hwang, Bsn-care: A secure iot-based modern healthcare system using body sensor network, IEEE Sens. J. 16 (5) (2016) 1368–1376.
[11] A.A. Mutlag, M.K.A. Ghani, N.A. Arunkumar, M.A. Mohammed, O. Mohd, Enabling technologies for fog computing in healthcare IoT systems, Future Gener. Comput. Syst. 90 (2019) 62–78.
[12] A. Vijayalakshmi, L. Arockiam, A Secured Architecture for IoT Healthcare System, 2020, pp. 904–911, http://dx.doi.org/10.1007/978-3-030-24643-3_106.
[13] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 881–892.
[14] A.M. Ortiz, D. Hussein, S. Park, S.N. Han, N. Crespi, The cluster between internet of things and social networks: Review and research challenges, IEEE Internet Things J. 1 (3) (2014) 206–215.
[15] W. Raghupathi, V. Raghupathi, Big data analytics in healthcare: promise and potential, Health Inf. Sci. Syst. 2 (1) (2014) 3.
[16] M. Chen, Y. Hao, K. Hwang, L. Wang, L. Wang, Disease prediction by machine learning over big data from healthcare communities, IEEE Access 5 (2017) 8869–8879.
[17] S.B. Baker, W. Xiang, I. Atkinson, Internet of things for smart healthcare: Technologies, challenges, and opportunities, IEEE Access 5 (2017) 26521–26544.
[18] Y. Wu, H. Huang, N. Wu, Y. Wang, M.Z.A. Bhuiyan, T. Wang, An incentive-based protection and recovery strategy for secure big data in social networks, Inform. Sci. 508 (2020) 79–91.
[19] K. Xing, C. Hu, J. Yu, X. Cheng, F. Zhang, Mutual privacy preserving k-means clustering in social participatory sensing, IEEE Trans. Ind. Inf. 13 (4) (2017) 2066–2076.
[20] S.B. Baker, W. Xiang, I. Atkinson, Internet of things for smart healthcare: Technologies, challenges, and opportunities, IEEE Access 5 (2017) 26521–26544.
[21] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, G. Marrocco, Rfid technology for iot-based personal healthcare in smart spaces, IEEE Internet Things J. 1 (2) (2014) 144–152.
[22] J. Vaidya, C. Clifton, Privacy-preserving k-means clustering over vertically partitioned data, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 206–215..
[23] S. Jha, L. Kruger, P. McDaniel, Privacy preserving clustering, in: European Symposium on Research in Computer Security, Springer, 2005, pp. 397–417.
[24] P. Bunn, R. Ostrovsky, Secure two-party k-means clustering, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM, 2007, pp. 486–497.
[25] A. Blum, C. Dwork, F. McSherry, K. Nissim, Practical privacy: the sulq framework, in: Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM, 2005, pp. 128–138.
[26] L.I. Yang, Guangzhou, China, research on differential privacy preserving k-means clustering, Comput. Sci. 1 (59) (2013) 1–34.
[27] X. Chen, J. Ji, C. Luo, W. Liao, P. Li, When machine learning meets blockchain: A decentralized, privacy-preserving and secure design, in: 2018 IEEE International Conference on Big Data, IEEE, 2018, pp. 1178–1187.
[28] T.-K. Yu, D. Lee, S.-M. Chang, J. Zhan, Multi-party k-means clustering with privacy consideration, in: International Symposium on Parallel and Distributed Processing with Applications, IEEE, 2010, pp. 200–207.
[29] C. Miao, L. Su, W. Jiang, Y. Li, M. Tian, A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems, in: IEEE INFOCOM 2017-IEEE Conference on Computer Communications, IEEE, 2017, pp. 1–9.
[30] C. Fontaine, F. Galand, A survey of homomorphic encryption for nonspecialists, EURASIP J. Inf. Secur. 2007 (2007) 15.
[31] D. Laurichesse, L. Blain, Optimized implementation of rsa cryptosystem, Comput. Secur. 10 (3) (1991) 263–267.
[32] K. Lin, Privacy-preserving kernel k-means clustering outsourcing with random transformation, Knowl. Inf. Syst. 49 (3) (2016) 885–908.
[33] D. Liu, E. Bertino, X. Yi, Privacy of outsourced k-means clustering, in: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, ACM, 2014, pp. 123–134.
[34] X. Yi, Y. Zhang, Equally contributory privacy-preserving k-means clustering over vertically partitioned data, Inf. Syst. 38 (1) (2013) 97–107.
[35] K.-J. Kim, Drinking fountains data, 2019, http://data.vancouver.ca/datacatalogue/drinkingFountains.htm.
[36] T.-S. Lim, Haberman's survival data set, 1999, http://archive.ics.uci.edu/ml/datasets/Haberman.
[37] A.G.L.O. Jorge, L. Reyes-Ortiz, Davide Anguita, X. Parra, Human activity recognition using smartphones data set, 2012, https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones.

Xuancheng Guo is currently pursuing the master's degree from the School of Mathematics and Information, Fujian Normal University. She received a bachelor's degree in computer science from the Fujian Institute of Engineering in China in 2017. Her research interests include software-defined networking, deep learning and network security.

Hui Lin is a professor in the College of Mathematics and Informatics at the Fujian Normal University, FuZhou, China. He received his Ph.D. degree in Computing System Architecture from College of Computer Science of the Xidian University, China, in 2013. Now he is a M.E. supervisor in College of Mathematics and Informatics at Fujian Normal University, FuZhou, China. His research interests include mobile cloud computing systems, blockchain, and network security. He has published more than 50 papers in international journals and conferences.



Yulei Wu is a Senior Lecturer with the Department of Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, United Kingdom. He received the B.Sc. degree (1st Class Hons.) in Computer Science and the Ph.D. degree in Computing and Mathematics from the University of Bradford, United Kingdom, in 2006 and 2010, respectively. His expertise is on networking and his main research interests include computer networks, networked systems, internet of things, software defined networks and systems, network management, and network security and privacy. Dr. Wu contributes to major conferences on networking and networked systems as various roles, including the Steering Committee Chair, General Chair and Program Chair. His research has been supported by Engineering and Physical Sciences Research Council of United Kingdom, National Natural Science Foundation of China, University's Innovation Platform and industry. He is an Editor of IEEE Transactions on Network and Service Management, Computer Networks (Elsevier) and IEEE Access. He is a Senior Member of the IEEE, and a Fellow of the HEA (Higher Education Academy).

Min Peng is currently pursuing the master's degree from the School of Mathematics and Information, Fujian Normal University. He received a bachelor's degree in computer science from the Lanzhou University of Technology in China in 2018. His research interests include blockchain, deep learning and network security.