



## Article

# PPK-Means: Achieving Privacy-Preserving Clustering Over Encrypted Multi-Dimensional Cloud Data

Hui Yin <sup>1</sup>, Jixin Zhang <sup>2,\*</sup>, Yinqiao Xiong <sup>1,3,\*</sup>, Xiaofeng Huang <sup>4</sup> and Tiantian Deng <sup>1,3</sup>

<sup>1</sup> College of Computer Engineering and Applied Mathematics, Changsha University, Changsha 410022, China; yhui@ccsu.edu.cn (H.Y.); dt@ccsu.edu.cn (T.D.)

<sup>2</sup> College of Information Science and Engineering, Hunan University, Changsha 410082, China

<sup>3</sup> College of Computer, National University of Defense Technology, Changsha 410073, China

<sup>4</sup> Department of Data Center, Hankou Bank, Wuhan 430000, China; 105714@hkbchina.com

\* Correspondence: zhangjixin@hnu.edu.cn (J.Z.); yq.xiong@ccsu.edu.cn (Y.X.);

Tel.: +86-134-2982-7607 (J.Z.); +86+133-9760-8869 (Y.X.)

Received: 21 September 2018 ; Accepted: 6 November 2018; Published: 8 November 2018



**Abstract:** Clustering is a fundamental and critical data mining branch that has been widely used in practical applications such as user purchase model analysis, image color segmentation, outlier detection, and so on. With the increasing popularity of cloud computing, more and more encrypted data are converging to cloud computing platforms for enjoying the revolutionary advantages of the cloud computing paradigm, as well as mitigating the deeply concerned data privacy issues. However, traditional data encryption makes existing clustering schemes no more effective, which greatly obstructs effective data utilization and frustrates the wide adoption of cloud computing. In this paper, we focus on solving the clustering problem over encrypted cloud data. In particular, we propose a privacy-preserving  $k$ -means clustering technology over encrypted multi-dimensional cloud data by leveraging the scalar-product-preserving encryption primitive, called PPK-means. The proposed technique is able to achieve efficient multi-dimensional data clustering as well to preserve the confidentiality of the outsourced cloud data. To the best of our knowledge, our work is the first to explore the privacy-preserving multi-dimensional data clustering in the cloud computing environment. Extensive experiments in simulation data-sets and real-life data-sets demonstrate that our proposed PPK-means is secure, efficient, and practical.

**Keywords:** cloud computing; data encryption;  $k$ -means; privacy-preserving clustering

## 1. Introduction

### 1.1. Motivation

In the big data era, data mining helps people quickly discover new and valuable knowledge from large-scale data-sets, which has been used in various fields such as finance, power, insurance, biology, etc. Nowadays, data mining, as one of the core techniques of artificial intelligence, has been attracting more and more attention from both industry and academia.

Depending on whether the data are labeled or not, data mining algorithms are mainly classified into two categories: Supervised Learning and Unsupervised Learning. No matter what category of algorithm, they commonly output a trained model, which can be used for decision or prediction for future input data.

In recent years, with the increasing popularity of cloud computing, more and more individuals and enterprises have had a burning desire to outsource their applications as well as large-scale data storage to the cloud server for enjoying the abundant benefits brought by cloud computing,

such as cost-efficiency, flexibility, elastic high quality services, etc. We demonstrate a representative cloud-based application by the following example.

Suppose that an e-commerce company outsources a customer analysis program (CAP) to the cloud server. To achieve fast and accurate customer analysis, a trained model is sent to the cloud server. By the model, CAP can find underlying customers by clustering new registered users or classifying customer groups.

However, in the above example, if the trained model and user information are uploaded to the cloud server in the form of plaintext, the privacy problem will arise, as the cloud server is not always fully trusted [1]. A practical case is that the cloud server could analyze the customers' purchase model and satisfaction according to the outsourced model and user information, that may be considered as a trade secret for some companies. Moreover, a user may also be unwilling to public his/her private information to the cloud server, such as position, age, preference, etc. Encrypting trained models and user information is an effective approach to provide a high-strength privacy guarantee for outsourced data [2]. Unfortunately, traditional encryption schemes make the existing data mining algorithms ineffective.

How to achieve privacy-preserving data mining over encrypted data has become a research focus recently. Due to supporting arithmetic operations directly over ciphertexts, homomorphic encryption is currently the most popular encryption technique to implement encrypted data mining schemes, which can provide data semantic security via cryptographic means. By leveraging homomorphic encryption, several data mining approaches have developed the corresponding encryption versions, such as gradient descent [3], ridge regression [4,5], support vector machine [6], naïve bayes [6], decision trees [6],  $k$ -Nearest Neighbor [7]. These techniques focus on the classification problem in the data mining tasks, which belongs to the Supervised Learning scope. A limitation of these schemes is not to efficiently process multi-dimension data due to high computational complexity of homomorphic encryption.  $k$ -Means clustering, as the Unsupervised Learning scope, is a fundamental and critical data mining algorithm that has been widely used in practical applications. Recently, researchers used secure multiparty computation protocols to construct several privacy-preserving  $k$ -means clustering schemes [8–11]. These solutions, however, require participants to cooperatively finish the clustering tasks without revealing any of their individual data items. Moreover, in such solutions, some intermediate computations have to rely on non-encryption data [7], which is not suitable for the cloud-based data outsourcing paradigm, as exposed plaintext data compromise the semantic security.

Another line of research utilizes data perturbation techniques such as differential privacy [12] to achieve privacy-preserving data mining. Since these approaches [13–15] can only work over the data with the form of plaintext, which are perturbed by noise, they cannot be applicable for semantically secure encrypted data.

In this paper, we consider the cloud-based outsourcing environment and propose a privacy-preserving  $k$ -means clustering over encrypted multi-dimensional cloud data, which achieves the following three basic goals: (1) efficiently processes multi-dimension data; (2) the cloud server alone performs clustering tasks without the cooperation parties; (3) achieves data semantic security against the "honest-but-curious" cloud server.

## 1.2. Contributions

In this paper, we mainly make three key contributions, which can be summarized as follows.

- (1) We propose an encrypted cloud-based data mining system model. In our model, the cloud server can perform privacy-preserving clustering and decision over encrypted outsourced data on behalf of users.
- (2) Considering the clustering problem, we propose a privacy-preserving  $k$ -means clustering framework based on our proposed system model. In terms of different attack models, three concrete privacy-preserving  $k$ -means clustering schemes are constructed, PPK-means\_1,

PPK-means\_2, and PPK-means\_3, all of which allow the cloud server alone handle multi-dimension data efficiently in a privacy-preserving manner.

- (3) We provide detailed security analysis for our designs to demonstrate the privacy preservation guarantee against different threat models. Extensive experimental evaluations in simulation data-sets and real-life data-sets demonstrate our scheme is secure, correct, and practical.

The rest of this paper is organized as follows. We review the related work in Section 2. The scenario of our proposed scheme is provided in Section 3, including system model, threat model, and several basic techniques used for our scheme. We also design a work flow and framework for our scheme and give an algorithm overview in this section. Three PPK-Means schemes are constructed in Section 4. Security analyses and evaluations for our proposed schemes are described in Section 5 and Section 6, respectively. Finally, we state the limitations and conclude this paper in Section 7.

## 2. Related Work

Since homomorphic encryption allows for direct arithmetic operations over encrypted data without decrypting data, most of existing works mainly rely on this cryptosystems to implement semantically secure data mining over encrypted data. Graepel et al. [3] used full homomorphic encryption to obtain an encrypted model through gradient descent. Nikolaenko et al. [4] used garbled circuit and additively homomorphic encryption to design a secure ridge regression scheme, which has better computational performance compared with Graepel et al.'s work. However, it still suffers from expensive communication cost. To further improve the time and communication costs, Hu et al. [5] utilized only additively homomorphic to construct a secure and efficient ridge regression scheme with negligible errors. Due to acceptable performance, their scheme makes the encrypted data mining more practical. To solve the problem that only provides privacy-preserving during training phase while not addressing decision [16–18], Bost et al. [6] used additively homomorphic and secure two-party computation protocols to design efficient privacy-preserving protocols for a broad class of classifiers, including hyperplane decision-based, naïve bayes, and decision trees. Their scheme provides the privacy guarantees in both training phase and classification phase, yet, it involves an impractical computation and communication cost. Liu et al. first considered an outsourced data mining scenario and proposed an encrypted gradient descent protocol [19] and encrypted support vector machine protocol [20]. However, the two protocols need the users and the third-party server to perform collaborative operations in multiple rounds. The above schemes focus on the classification problem in the data mining tasks. On the other hand, the secure multiparty computation was also used to construct privacy-preserving classification schemes over vertically or horizontally distributed data [21–23], in which the multiple data holders cooperatively compute trained models without revealing their own data.

In the data mining field, clustering is a fundamental and critical branch that has been widely used in practical applications. Because each clustering technique has its own advantage, this paper concentrates on the most representative  $k$ -means clustering technique. The traditional  $k$ -means was first used by James MacQueen in 1967 [24]. Basu et al. [25] proposed to presents a pairwise constrained clustering framework and an optimal  $k$ -means method for actively selecting informative pairwise constraints to get improved clustering performance. Recently, to improve data cluster performance, researchers propose to use particle swarm optimization (PSO) to enhance  $k$ -means data clustering [26].  $k$ -means clustering is rather easy to implement and apply even on large data-sets. It has been successfully used in various topics, including market segmentation, computer vision, geostatistics, astronomy and agriculture [27]. In this paper, we focus on a traditional  $k$ -means algorithm. Toward the clustering problem in encrypted data environments, researchers utilized the secure multiparty computation techniques to design several privacy-preserving  $k$ -means clustering schemes over partitioned data [8–11]. In such schemes, data are distributed among multiple participants, who cooperatively perform the clustering tasks without revealing any of their own data items, which conflicts with the basic goal of our design stated in Section 1.

### 3. Scenario

#### 3.1. System Model

Our proposed system model is shown in Figure 1, which consists of three entities, i.e., the data owner, the data user, and the cloud server. In our system model, the data owner is actually a model provider, who trains owned data by using a certain data mining algorithm to build a trained model locally and provides a decision service for the data user in the cloud-based service outsourcing paradigm. The data user sends their data to the cloud server and calls the service to make a decision. To guarantee the privacy of the trained model and the data user's data, they are encrypted by the data owner and the data user, respectively, before being outsourced to the cloud server. Once receiving the data user's encrypted data, the cloud server is responsible for performing decision over the encrypted trained model without knowing any sensitive private information. Finally, the cloud server sends the decision results (denoted by labels) to the data user.

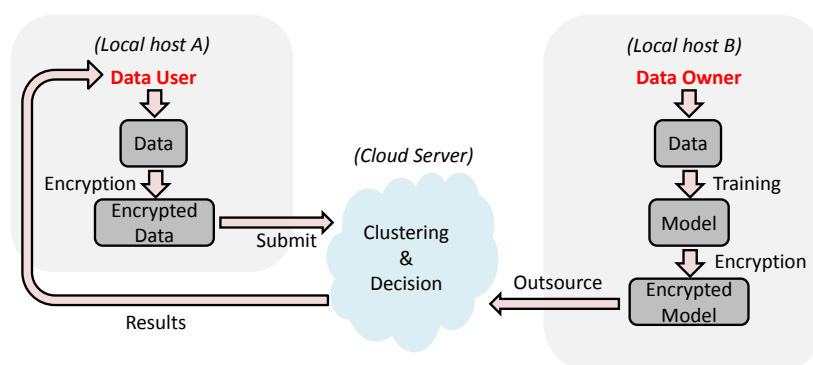


Figure 1. The System Model.

We use a concrete example to further explain our system model as follows. A company (the data owner) provides a decision service for business by outsourcing a data-driven model to the cloud server. A customer of the company may be a hospital (the data user), who uses the provided service to make decisions for patients' data. To hide the trained model and patients' sensitive information from the cloud server, their respective encrypted versions are submitted to the cloud server. The cloud server performs decisions in a privacy-preserving manner.

#### 3.2. Threat Model

In our threat model, the cloud server is considered as an "honest-but-curious" threat entity, which has been widely used in cloud based schemes such as secure cloud search [28–31]. Specifically, the cloud server promises that it honestly obeys the application outsourcing and correctly performs functionality protocols. However, it may be "curious" to access and analyse the trained model and the data user's data to obtain some sensitive information that the user is unwilling to be in public. Depending on what information the cloud server is available to, we consider the following two threat models.

- **Known Ciphertext Model:** In this model, except for the encrypted trained model and the encrypted data user's data, the cloud server can know nothing else.
- **Known Background Model:** In this model, the cloud server bears more knowledge than what can be accessed in the known ciphertext model, by which it can obtain useful information by analysing ciphertext. For example, the cloud server may know a few plaintexts and the corresponding ciphertexts, which can be used to recover the whole plaintexts from the ciphertext space.

Note that, in the above two threat models, the sharing encryption keys between the data owner and the data user are kept secret from the cloud server.

### 3.3. Basic Techniques

#### 3.3.1. $k$ -means

$k$ -means is one of the most popular unsupervised clustering algorithms, which can automatically partition a collection of objects into  $k$  disjoint subsets based on a certain similarity metrics. Normally, we call  $k$  disjoint subsets as *clusters* and use the Euclidean distance to define the similarity among the objects. In other words, the closer Euclidean distance between two objects is, the more similar they are, the higher probability that they are clustered into the same cluster.

#### 3.3.2. Scalar-Product-Preserving Encryption

We adopt the *Scalar-product-preserving Encryption SPE* proposed in [32] as our foundation. The encryption scheme allows one to compute the correct inner product of two encrypted vectors without knowing their actual values. Here, we only provide the main idea about *SPE*; readers can refer to [32] to obtain detailed contents. Given two column vectors  $\vec{p}$  and  $\vec{q}$  with  $D$  dimensions, a  $D \times D$  invertible matrix  $M$  as the symmetric key,  $\vec{p}$  and  $\vec{q}$  are encrypted as  $E(\vec{p}) = M^T \vec{p}$  and  $E(\vec{q}) = M^{-1} \vec{q}$ , respectively. By computing  $(E(\vec{p}))^T \cdot E(\vec{q}) = \vec{p}^T M M^{-1} \vec{q} = \vec{p}^T \vec{q}$ , thus the scalar product of two vectors is preserved.

### 3.4. The Basic Design

In this section, we further state our design, including the work flow and the framework of PPK-means, an algorithm overview is given as well lastly.

#### 3.4.1. Work Flow

A data owner trains his/her possessive data by a traditional  $k$ -means to build a model with  $k$  central points (i.e.,  $k$  clusters). To prevent the cloud server from gaining information about the trained model, all  $k$  clusters are encrypted using an encryption function  $E_{key_1}()$  under a specified key  $key_1$  by the data owner before outsourcing. On the other hand, a data user with a  $D$ -dimension data  $x \in \mathbb{R}^D$  wishes to obtain a decision result for  $x$  according to the outsourced encrypted model. Likewise, for privacy, the data user hides the actual value of  $x$  from the cloud server by encrypting  $E_{key_2}(x)$  under the key  $key_2$ . In our scheme, since  $key_1$  is an invertible matrix and  $key_2$  is the invert matrix of  $key_1$ , we uniformly denote  $key_1$  and  $key_2$  as a symmetric key  $SK$  in the following. After encrypting the data  $x$ , the data user submits the corresponding ciphertext to the cloud server, who clusters the encrypted data into the correct cluster based on a privacy-preserving distance comparison. As a result, we claim that the privacy-preserving distance comparison become the key to solve our encrypted  $k$ -Means problem, which will be achieved in our scheme by skillfully utilizing the scalar-product-preserving encryption *SPE*. We present the work flow of our scheme, as shown in Figure 2.

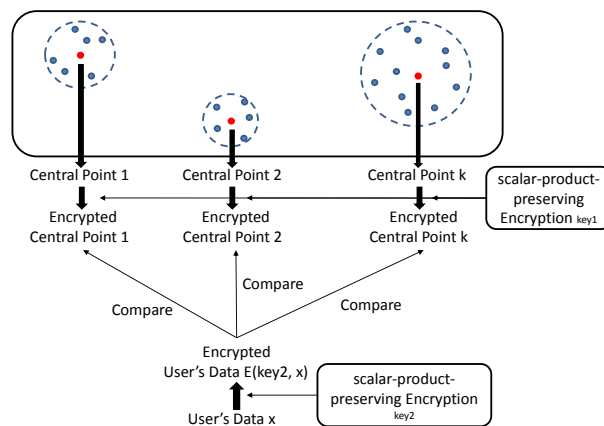


Figure 2. The Work Flow of PPK-means.

### 3.4.2. Framework

The PPK-means is composed of the following four probabilistic polynomial-time algorithms:

- **Setup:** In the initialization, the system generates a symmetric key  $SK$ , which is shared between the data owner and the data user.
- **ModelEnc**( $P, SK$ ): The data owner takes the trained model  $P$  with  $k$  central points and the symmetric key  $SK$  as input, the algorithm outputs the encrypted trained model  $E(SK, P)$ .
- **DataEnc**( $x, SK$ ) The data user takes a  $D$ -dimensional data vector  $x \in \mathbb{R}^D$  and the symmetric key  $SK$  as input, the algorithm outputs the encrypted data vector  $E(SK, x)$ .
- **Decision**( $E(SK, P), E(SK, x)$ ) The cloud server takes the encrypted model  $E(SK, P)$  and the encrypted data vector  $E(SK, x)$ , the algorithm outputs a label for  $x$ , which means which central point  $x$  is clustered into.

### 3.4.3. Algorithm Overview

Finally, we give an algorithm overview of our PPK-means, as shown Algorithm 1. In Section 4, we will design different encryption function  $E$  and  $Dist$  to achieve different security strength under different threat models, i.e., Known Ciphertext Model and Known Background Model.

---

#### Algorithm 1 PPK-Means Algorithm

---

**Input:** A set of data vectors:  $Q = \{x_1, x_2, \dots, x_n | x_i \in \mathbb{R}^D\}$ ; A symmetric key  $SK$ .

**Output:** A set of labels:  $\{C = c_a, \dots, c_k, \dots, c_b\}$ .

- 1: Initial an empty set for the label set  $C$ .
  - 2: Use  $k$ -Means clustering to train data and obtain a set of central points:  $P = \{p_1, p_2, \dots, p_k | p_j \in \mathbb{R}^D\}$ .
  - 3: Encrypt  $P$  by an encryption function  $E(P)$  under  $SK$ , and obtain a set of encrypted central points:  
 $EP = \{E(SK, p_1), \dots, E(SK, p_k) | E(SK, p_i) \in \mathbb{R}^D\}$ .
  - 4: Encrypt  $Q$  by the encryption function  $E(Q)$  under  $SK$ , and obtain a set of encrypted data points:  
 $EQ = \{E(SK, x_1), \dots, E(SK, x_n) | E(SK, x_i) \in \mathbb{R}^D\}$
  - 5: **for**  $i \leq n$  **do**
  - 6:   **for**  $j \leq k$  **do**
  - 7:     Select an encrypted central point  $E(SK, p_j) \in EP$ .
  - 8:     Input an encrypted data point  $E(SK, x_i) \in EQ$ .
  - 9:     Compare with  $E(SK, p_j)$  and  $E(SK, x_i)$  by a scalar product function  $c_j = Dist(E(SK, p_j), E(SK, x_i))$ .
  - 10:   **end for**
  - 11:   Find a cluster  $c_{max}$  which has maximum distance between  $Dist(E(SK, p_j)$  and  $E(SK, x_i)$ .
  - 12:   Insert the label  $c_{max}$  of the cluster into the label set  $C$ .
  - 13: **end for**
  - 14: **Return** the label set  $C$
- 

In addition, to formally evaluate the distance between two data points, given two multi-dimension data vectors  $p, q \in \mathbb{R}^D$ , we define the distance  $d(p, q)$  between  $p$  and  $q$  according to Equation (1). We conduct the distance comparison between  $q, p_1$  and  $p_2$  according to Equation (2).

$$d(p, q) = \|p\|^2 - 2 \cdot p \cdot q + \|q\|^2 \quad (1)$$

$$discom((p_1, p_2), q) = \|p_2\|^2 - \|p_1\|^2 + 2(p_1 - p_2) \cdot q \quad (2)$$

where  $\|p\|$  denotes euclidean norm of  $p$ . Based on Equations (1) and (2), we will construct the privacy-preserving distance comparison between two encrypted data points. Before giving our PPK-means construction, we first prove the following important conclusion.



**Theorem 1.** Given two center points  $p_1, p_2$  and a data point  $x$ , if  $\text{discom}((p_1, p_2), q) > 0$ , then  $q$  is nearer to  $p_1$  than  $p_2$ , where  $\|x\|$  denotes euclidean norm of  $x$ .

**Proof.** According to Equation (2), we have

$$\begin{aligned}\text{discom}((p_1, p_2), q) &= \|p_2\|^2 - \|p_1\|^2 + 2(p_1 - p_2) \cdot q \\ &= (\|p_2\|^2 - 2 \cdot p_2 \cdot q) - (\|p_1\|^2 - 2 \cdot p_1 \cdot q) \\ &= (\|p_2\|^2 - 2 \cdot p_2 \cdot q + \|q\|^2) - (\|p_1\|^2 - 2 \cdot p_1 \cdot q + \|q\|^2) \\ &\stackrel{\text{Equation (1)}}{=} d(p_2, q) - d(p_1, q)\end{aligned}$$

Therefore, if  $\text{discom}(p_1, p_2) > 0$ , then  $d(p_2, q) - d(p_1, q) > 0 \Rightarrow d(p_2, q) > d(p_1, q)$ , this indicates the distance between  $p_2$  and  $q$  is farer than that between  $p_1$  and  $q$ , i.e.,  $q$  is nearer to  $p_1$  than  $p_2$ . The theorem is proved.  $\square$

#### 4. PPK-Means Construction

Intuitively, a distance-recoverable encryption (DRE) [32] can be used to construct our PPK-means scheme, as the distance between two data points can be recovered according to their corresponding ciphertexts. For example, if  $E$  denotes a DRE, given two data points  $x$  and  $y$ , then  $\text{dist}(x, y) = \text{dist}(E(x, K), E(y, K))$ , where  $K$  is a key of  $E$ . However, it is shown to be not secure in practice [33]. In this paper, inspired by Wong et al.'s work, we utilize the secure scalar-product-preserving Encryption to construct the PPK-means against different threat models.

##### 4.1. PPK-Means\_1

In this subsection, we construct a basic framework of the PPK-means scheme PPK-means\_1, which is secure against Known Ciphertext Model. Based on the framework, we will improve the security and propose PPK-means schemes under Known Background Model.

**Setup.** In the initialization phase, the system generates a  $(D + 1) \times (D + 1)$  invertible random matrix  $M$  as the symmetric key  $SK$ , which is secretly shared by the data owner and the data user.

**ModelEnc**( $P, SK$ ). Let  $p_i$ , denoted by a column vector with  $D$  dimensions, be the  $i$ -th central point in  $P$ . The data owner first extends  $p_i$  to a  $(D + 1)$ -dimensional vector  $p_i^{(D+1)}$  by setting the last element to be  $-0.5\|p_i\|^2$ , as shown in Equation (3), and then vector  $p_i^{(D+1)}$  is encrypted under the key  $M$  as the ciphertext of the center point  $p_i$ , as shown in Equation (4)

$$p_i^{(D+1)} = (p_i^T, -0.5\|p_i\|^2)^T \quad (3)$$

$$E(M, p_i) = p'_i = M^T \cdot p_i^{(D+1)} \quad (4)$$

**DataEnc**( $x, SK$ ).  $x$  is the data user's input data denoted by a column vector with  $D$  dimensions. The data user first extends  $x$  to a  $(D + 1)$ -dimensional vector  $x^{(D+1)}$  by setting the last element to be 1 and chooses a random number  $r$ , and randomizes the extended vector by Equation (5)

$$x^{(D+1)} = r(x^T, 1)^T \quad (5)$$

After generating  $x^{(D+1)}$ , the data user uses key  $M$  to further encrypt the vector as the ciphertext of input data  $x$ , as shown in Equation (6)

$$E(M^{-1}, x) = x' = M^{-1} \cdot x^{(D+1)} \quad (6)$$

**Decision**( $E(SK, P), E(SK, x)$ ). Upon receiving the encrypted data  $x'$  from the data user, the cloud server is responsible for performing decision over encrypted trained model without knowing actual

value about the trained model and user's input vector. In other words, according to Algorithm 1, the cloud server determines which center point the encrypted vector  $x'$  should be clustered into according to a privacy-preserving distance comparison *Dist*. Specifically, without loss of generality, let  $p'_i$  and  $p'_j$  be two encrypted center points of  $p_i$  and  $p_j$ , the cloud server can determine whether  $x$  is nearer to  $p_i$  than  $p_j$  by checking whether the condition *Dist* :  $(p'_i - p'_j) \cdot x' > 0$  holds. The fact can be verified based on Equation (2) and SPE technique as follows.

$$\begin{aligned}
 & (p'_i - p'_j) \cdot x' \\
 &= (p'_i - p'_j)^T \cdot x' \\
 &= (M^T \cdot p_i^{(D+1)} - M^T \cdot p_j^{(D+1)})^T \cdot M^{-1} \cdot x^{(D+1)} \\
 &= (p_i^{(D+1)} - p_j^{(D+1)})^T \cdot MM^{-1} \cdot x^{(D+1)} \\
 &= (p_i - p_j)^T \cdot (rx) + (-0.5\|p_i\|^2 + 0.5\|p_j\|^2)r \\
 &= 0.5r(\|p_j\|^2 - \|p_i\|^2 + 2(p_i - p_j)^T \cdot x)
 \end{aligned} \tag{7}$$

Therefore, according to Equation (2) and Theorem 1,  $(p'_i - p'_j) \cdot x' = \text{discom}((p_i, p_j), x)$ , if  $(p'_i - p'_j) \cdot x' > 0$ , then  $\text{discom}((p_i, p_j), x) > 0$ , therefore  $x$  is nearer to  $p_i$  than  $p_j$ .

**Example.** We present a concrete example to help understand the decision processes and computation details of Equation (7). Assume that there exist two center points with two dimensions in the trained model  $p_i = (2, 4)^T$  and  $p_j = (1, 2)^T$ , given a data vector  $x = (3, 4)^T$ , all of them are denoted by column vector. We first compare their Euclidean distances in the plaintext environment. Since  $d(p_i, x) = 1$  and  $d(p_j, x) = \sqrt{8}$ ,  $d(p_i, x) < d(p_j, x)$ , such that  $x$  is nearer to  $p_i$  than  $p_j$ , where  $d(\cdot)$  denote a Euclidean distance. Next, we encrypt  $p_i, p_j, x$  and verify whether  $(p'_i - p'_j) \cdot x' > 0$  holds or not according to Equation (7).

We first choose a key with  $3 \times 3$  invertible matrix  $M = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$  and its invert matrix

$M^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 1 \end{bmatrix}$ . Then, we compute ciphertexts of  $p_i$  and  $p_j$  according to Equation (4) as  $p'_i = M^T \cdot (2, 4, -0.5 \times 20)^T = (12, 6, 16)^T$  and  $p'_j = M^T \cdot (1, 2, -0.5 \times 5)^T = (-1.5, 3, 0.5)^T$ , where  $\|p_i\| = \sqrt{20}$  and  $\|p_j\| = \sqrt{5}$ . By Equation (5), we choose a random number  $r = 2$  and compute the ciphertext of  $x$  as  $x' = M^{-1} \cdot (6, 8, 2)^T = (-2, 4, 4)^T$ . Finally, we compute  $(p'_i - p'_j) \cdot x'$  as follows.

$$\begin{aligned}
 (p'_i - p'_j) \cdot x' &= \left( \begin{bmatrix} 12 \\ 6 \\ 16 \end{bmatrix} - \begin{bmatrix} -1.5 \\ 3 \\ 0.5 \end{bmatrix} \right)^T \cdot \begin{bmatrix} -2 \\ 4 \\ 4 \end{bmatrix} \\
 &= [10.5, 3, 15.5] \cdot \begin{bmatrix} -2 \\ 4 \\ 4 \end{bmatrix} \\
 &= 53 > 0
 \end{aligned}$$

In the above example, since  $(p'_i - p'_j) \cdot x' > 0$ ,  $x$  is nearer to  $p_i$  than  $p_j$ . The conclusion is consistent with the distance comparison over plaintext data. Therefore, our encryption can provide an effective and privacy-preserving distance comparison.

#### 4.2. PPK-Means\_2

Assume that the cloud server has some background knowledge, such as the plaintexts of several center points or a set of plaintexts of data user's input as well as their corresponding encrypted values.



If they are linearly independent, the cloud server can easily recover the key  $M$ . For example, there are  $m$  points  $\{p_1, \dots, p_m | p_i \in \mathbb{R}^D\}$  and the corresponding encrypted values  $p'_i (1 \leq i \leq m)$ , we can set up equations  $M^T \cdot p_i = p'_i$  from  $i = 1$  to  $m$  to recover  $M$  if  $p_i (1 \leq i \leq m)$  are linearly independent.

To solve the above problem, we propose PPK-means\_2 that uses Artificial Dimensions Extension and Random Asymmetric Splitting to make it very difficult to set up the equations.

**Setup.** The system generates two  $(D+2) \times (D+2)$  invertible random matrices  $M_1, M_2$  and a bit string  $S$  of  $(D+2)$ , the key  $SK$  is a three-tuple denoted by  $SK = \{M_1, M_2, S\}$ .

**ModelEnc**( $P, SK$ ). The data owner first chooses a random number  $r'$  and extends each center point  $p_i$  into a  $(D+2)$  dimensional vector as shown in Equation (8).

$$p_i^{D+2} = (p_i, -0.5||p_i||^2, r') \quad (8)$$

Then  $p_i^{D+2}$  is split into two vectors  $(p'_i, p''_i)$  according to bit string  $S$ . If  $S[j] = 0$ ,  $p'_i[j]$  and  $p''_i[j]$  are both equal to  $p_i^{D+2}[j]$ ; if  $S[j] = 1$ ,  $p_i^{D+2}[j]$  is randomly split such that  $p'_i[j] + p''_i[j] = p_i^{D+2}[j]$ . Finally, the data owner encrypts  $p_i$  as  $\{\hat{p}'_i = M_1^T p'_i, \hat{p}''_i = M_1^T p''_i\}$ .

**DataEnc**( $x, SK$ ). The data user first chooses a random number  $r$  and extends the data vector  $x$  as shown in Equation (9).

$$x^{D+2} = r(x^T, 1, 1)^T \quad (9)$$

Then  $x^{D+2}$  is split into two vectors  $(x', x'')$ . If  $S[j] = 0$ ,  $x^{D+2}[j]$  is randomly split such as  $x'[j] + x''[j] = x^{D+2}[j]$ ; if  $S[j] = 1$ ,  $x'[j]$  and  $x''[j]$  are set to be the same value as  $x^{D+2}[j]$ . Finally, the data point  $x$  is encrypted as  $\{\hat{x}' = M_1^{-1} x', \hat{x}'' = M_2^{-1} x''\}$ .

**Decision**( $E(SK, P), E(SK, x)$ ). The cloud server determines whether  $x$  is nearer to  $p_i$  than  $p_j$  by checking whether  $Dist : (p'_i - p'_j) \cdot x' + (p''_i - p''_j) \cdot x'' > 0$ . We verify the correctness according to Equation (2) and SPE technique as follows.

$$\begin{aligned} & (\hat{p}'_i - \hat{p}'_j) \cdot \hat{x}' + (\hat{p}''_i - \hat{p}''_j) \cdot \hat{x}'' \\ &= (\hat{p}'_i - \hat{p}'_j)^T \cdot \hat{x}' + (\hat{p}''_i - \hat{p}''_j)^T \cdot \hat{x}'' \\ &= (M_1^T p'_i - M_1^T p'_j)^T M_1^{-1} x' + (M_2^T p''_i - M_2^T p''_j)^T M_2^{-1} x'' \\ &= (p_i^{(D+2)} - p_j^{(D+2)})^T \cdot x^{D+2} \\ &= (p_i - p_j)^T \cdot (rx) + (-0.5||p_i||^2 + 0.5||p_j||^2)r + (rr' - rr') \\ &= 0.5r(||p_j||^2 - ||p_i||^2) + 2(p_i - p_j)^T \cdot x \end{aligned} \quad (10)$$

Therefore, according to Equation (2) and **Theorem 1**,  $(p'_i - p'_j) \cdot x' + (p''_i - p''_j) \cdot x'' = discom((p_i, p_j), x)$ , if  $((p'_i - p'_j) \cdot x' + (p''_i - p''_j) \cdot x'' > 0)$ , then  $discom((p_i, p_j), x) > 0$ , therefore  $x$  is nearer to  $p_i$  than  $p_j$ .

#### 4.3. PPK-Means\_3

Though the random value  $r'$  is introduced to extend the dimensions of the original vector in PPK-means\_2, it is only an unknown constant value for the attacker. To eliminate the deterministic factor, we propose a security-enhanced scheme PPK-means\_3.

PPK-means\_3 is similar with PPK-means\_2, the only difference is that PPK-means\_3 extends the original center points and a data user's input data into  $D+3$  dimensions by introducing new random numbers. Correspondingly, the key  $M_1$  and  $M_2$  are  $(D+3) \times (D+3)$  invertible matrices and  $S$  is a bit string of  $D+3$ . Specifically, give a center point  $p_i$  and user's data  $x$ , they are extended as  $p_i^{D+3} = (p_i^T, -0.5||p_i||^2, 0.5r'_i, 1)^T$  and  $x^{(D+3)} = r(x^T, 1, 1, 0.5r''^T)^T$ , where  $r'_i$  and  $r''$  are two random numbers for  $p_i$  and  $x$ , respectively. In the decision phase, the cloud server determines whether  $x$

is nearer to  $p_i$  than  $p_j$  by checking whether  $\text{Dist} : (p'_i - p'_j) \cdot x' + (p''_i - p''_j) \cdot x'' > 0$ . We verify the correctness according to Equation (2) and SPE technique as follows.

$$\begin{aligned}
 & (\hat{p}'_i - \hat{p}'_j) \cdot \hat{x}' + (\hat{p}''_i - \hat{p}''_j) \cdot \hat{x}'' \\
 &= (\hat{p}'_i - \hat{p}'_j)^T \cdot \hat{x}' + (\hat{p}''_i - \hat{p}''_j)^T \cdot \hat{x}'' \\
 &= (M_1^T p'_i - M_1^T p'_j)^T M_1^{-1} x' + (M_2^T p''_i - M_2^T p''_j)^T M_2^{-1} x'' \\
 &= (p_i^{(D+3)} - p_j^{(D+3)})^T \cdot x^{D+3} \\
 &= (p_i - p_j)^T \cdot (rx) + (-0.5||p_i||^2 + 0.5||p_j||^2)r + \\
 &\quad 0.5r(r''_i - r''_j) + 0.5r(r''' - r''') \\
 &= 0.5r(||p_j||^2 - ||p_i||^2 + 2(p_i - p_j)^T \cdot x + (r''_i - r''_j)) \\
 &= 0.5r(d(p_j, x) - d(p_i, x) + (r''_i - r''_j))
 \end{aligned} \tag{11}$$

Obviously, since  $(r''_i - r''_j)$  is introduced as a part of the distance comparison, the final clustering result may not be as accurate as that of the original  $k$ -Means scheme. For the consideration of clustering accuracy, we let the random number  $r''_i$  in  $p_i^{D+3}$  follow a normal distribution  $N(u, \sigma^2)$ , where  $\sigma^2$  as a flexible trade-off parameter between accuracy and security. Detailed evaluation on accuracy will be given in Section 6.

## 5. Privacy Analysis

In our paper, the core technique used to design PPK-means is scalar-product-preserving encryption, SPE, whose security has been proven in [32]. Our design does not change the encryption construction of SPE except for introducing a different number of random numbers against different threat models. Therefore, as long as SPE is semantically secure, our scheme is secure, the an adversary cannot recover the plaintext from its ciphertext. To avoid duplication, we ignore the security proof of original SPE and only provide a privacy analysis for our proposed PPK-means under different threat models.

### 5.1. PPK-Means\_1

The security of PPK-means\_1 mainly relies on the symmetric key  $M$ . As long as the key  $M$  is kept secret from the cloud server, PPK-means\_1 is secure against **Known Ciphertext Model**, i.e., the cloud server knows nothing except the encrypted center points and the user's data.

However, PPK-means\_1 is not secure under **Known Background Model**. The cloud server may be able to recover the symmetric key  $M$  according to some known background knowledge. For example, if the cloud server knows that there are  $m$  points  $\{p_1, \dots, p_m | p_i \in \mathbb{R}^D\}$  and the corresponding encrypted values  $p'_i (1 \leq i \leq m)$ , it can set up equations  $M^T \cdot p_i = p'_i$  from  $i = 1$  to  $m$  to recover  $M$  if  $p_i (1 \leq i \leq m)$  are linearly independent.

For the user's data  $x$ , though introducing a random number  $r$  to scale  $x^{D+1}$ , the only randomness does not provide sufficient nondeterminacy in the final distance comparison.

### 5.2. PPK-Means\_2

To prevent the cloud server from correctly setting up equations to solve key  $M$ , PPK-means\_2 uses Artificial Dimensions Extension and Random Asymmetric Splitting to guarantee the security of encrypted model and data under **Known Background Model**. By introducing an extended random number  $r'$  and a split vector  $S$ , both of which are secret values from the view of the cloud server, it is difficult for the cloud server to obtain the plaintext of the outsourced model and user's data even though the cloud server has some useful background knowledge. The fact has been proven in [32,33].

### 5.3. PPK-Means\_3

In PPK-means\_2, though dimension extension with the random number  $r'$  and random splitting are used to hamper the known background knowledge attack, for given  $p_i$ ,  $p_j$ , and  $x$ , the cloud server can still obtain the value  $0.5r(d(p_j, x) - d(p_i, x))$  during the decision processes, which is only randomized by  $r$ . To improve the privacy guarantee of distance comparison, PPK-means\_3 further extends the dimensions of center points and user's data from  $D + 2$  to  $D + 3$  and introduce more random numbers compared with PPK-means\_2. According to Equation (11), the ultimate distance comparison metric  $0.5r(d(p_j, x) - d(p_i, x) + (r''_i - r''_j))$  is further obscured by  $(r''_i - r''_j)$ , where random number  $r''_i$  and  $r''_j$  are in  $p_i^{D+3}$  and  $p_j^{D+3}$ , respectively. Obviously, the same as PPK-means\_2, PPK-means\_3 is secure against **Known Background Model**.

## 6. Scheme Evaluation

### 6.1. Time Complexity Analysis

We analyze the time complexity of our scheme from two aspects: encryption and clustering.

**Time complexity of encryption.** As for encryption, we perform two kinds of operations: one is the multiplication between key  $M$  (Resp.  $M^{-1}$ ) and center point  $p$  (Resp. the user's data  $x$ ) ( $(D + 1)$ -dimension in PPK-means\_1,  $(D + 2)$ -dimension in PPK-means\_2,  $(D + 3)$ -dimension in PPK-means\_3), the time complexity is  $\mathcal{O}(D^2)$ ; the other is the Euclidean distance computation, the time complexity is  $\mathcal{O}(D)$ . Therefore, the total time complexity of encryption is  $\mathcal{O}(D^2) + \mathcal{O}(D)$ .

**Time complexity of clustering.** As for clustering, the main operation is the distance comparison one-by-one between user's data and all  $k$  center points. The time complexity of the distance comparison between the data and one central points is  $\mathcal{O}(1)$ , thus, when the data is clustered into the correct cluster, the time complexity is  $\mathcal{O}(k)$ . If the number of user's data is  $n$ , the total time complexity is  $\mathcal{O}(kn)$ .

### 6.2. Experimental Evaluation

In this section, we implement a prototype and present several experiments to show security and efficiency of our scheme.

#### 6.2.1. Setup

We evaluate our scheme in a client-server environment. The client is a Windows 7 desktop system with 3.20GHz Intel Core (TM) i5-6500 and 8-GB RAM and The server is Windows 7 desktop system with 3.60-GHz Intel Core (TM) i7-7700 CPU and 8-GB RAM. The client environment is used to evaluate performances of trained model encryption for the data owner and input data encryption for the data user. The server environment with more power simulates the cloud server to perform encrypted data clustering. All programs are developed by Java programming language with JRE 1.8.

#### 6.2.2. Evaluation with Simulation data-set

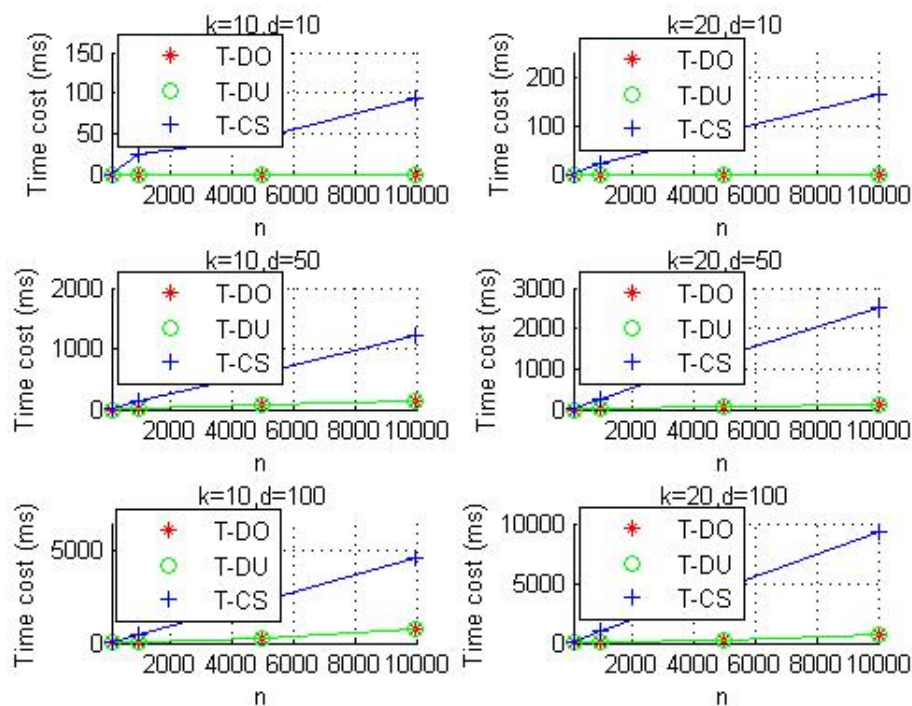
**data-sets.** To adequately evaluate the time cost of our proposed scheme, we simulate several different data-sets to test the time cost with taking different parameters  $(k, n, D)$  into consideration, where  $k$  is the number of clusters,  $n$  denotes the number of user's input data, and  $D$  specifies the number of dimensions of the center point and the user's input data.

**Time cost evaluation.** Table 1 shows the time cost of our scheme when fixing the parameters  $k = 10$ ,  $n = 10,000$ , and  $D = 10$ , where T-DO, T-DU, and T-CS denote the time cost on center points encryption for the data owner, data encryption for the data user, and data clustering for the cloud server, respectively. From Table 1, we can see that PPK-means\_2 and PPK-means\_3 need to spend more time compared to PPK-means\_1 to perform data clustering for cloud server due to Random Splitting, while the time costs of the three schemes on T-DO and T-DU is almost 0 millisecond.

**Table 1.** The time costs of the three PPK-means schemes.

Scheme	T-DO	T-DU	T-CS
PPK-means_1	<1 ms	<1 ms	53 ms
PPK-means_2	<1 ms	<1 ms	87 ms
PPK-means_3	<1 ms	<1 ms	87 ms

Next, we perform several group experiments to test the time cost by adjusting parameters ( $k, n, D$ ). Due to the highest security strength yet most time-consuming clustering performance, we mainly consider PPK-means\_3 as our evaluation object. The average time cost according to ten times of experimental results is shown in Figure 3. The six groups of experimental results show that the time cost is linear to the number of the user's data ( $n$ ). By comparing each row of the results, we can observe that the time cost linearly increases with the number of clusters ( $k$ ), and by comparing each column of the results, we can observe that the time cost is also linearly increases with the number of dimensions ( $D$ ) in each data.

**Figure 3.** The time costs in different groups of experiments.

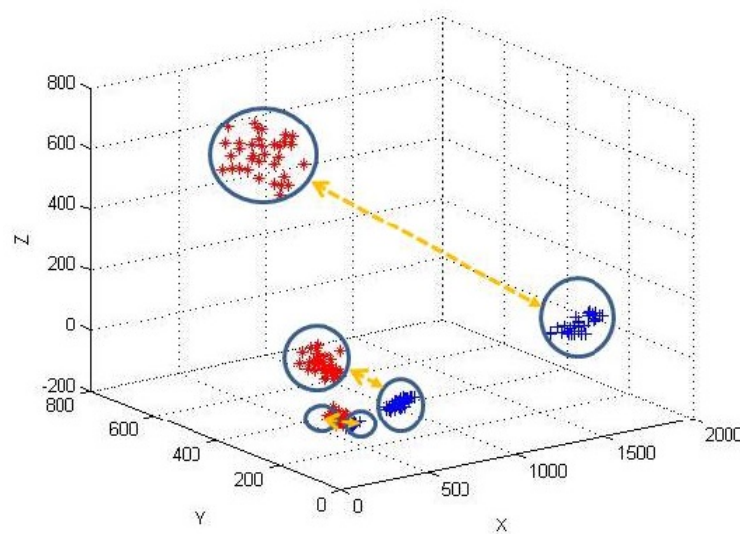
**Functionality evaluation.** To demonstrate the correctness of our scheme by functionality evaluation, we check whether the encrypted user's data are accurately clustered into the correct clusters as that of plaintext data clustering using  $k$ -means. The evaluation result with six-group experiments is shown in Table 2.

The accuracies of PPK-means\_1 and PPK-means\_2 show that all of the encrypted data are correctly clustered. Specifically, for PPK-means\_3, the accuracy is almost 100 % when we set the variance  $\sigma$  to be 1. From Table 2, we can see that the larger the number of dimensions in each data is, the higher the accuracy of clustering is.

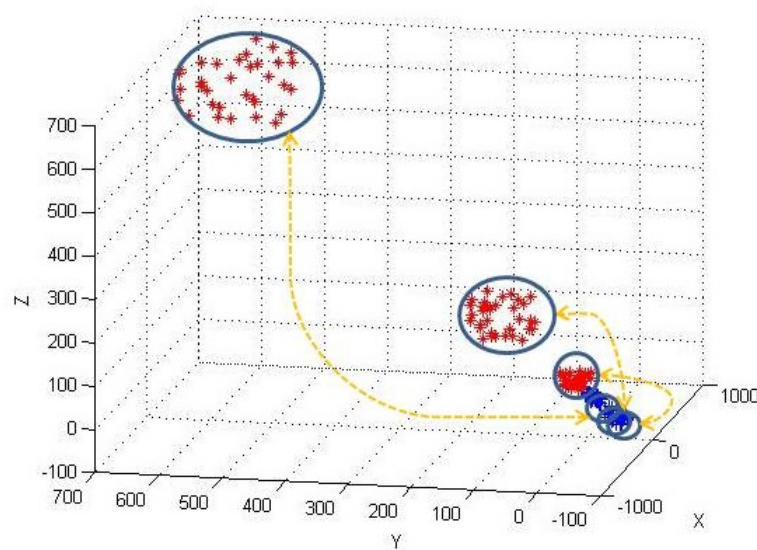
**Table 2.** The proportions of the correctly clustered encrypted points.

Group ID	Accuracy		
	PPK-Means_1	PPK-means_2	PPK-Means_3
Group 1. ( $k = 10, D = 10, n=10,000$ )	100%	100%	99.96%
Group 2. ( $k = 20, D = 10, n=10,000$ )	100%	100%	99.96%
Group 3. ( $k = 10, D = 50, n=10,000$ )	100%	100%	99.98%
Group 4. ( $k = 20, D = 50, n=10,000$ )	100%	100%	99.98%
Group 5. ( $k = 10, D = 100, n=10,000$ )	100%	100%	100%
Group 6. ( $k = 20, D = 100, n=10,000$ )	100%	100%	100%

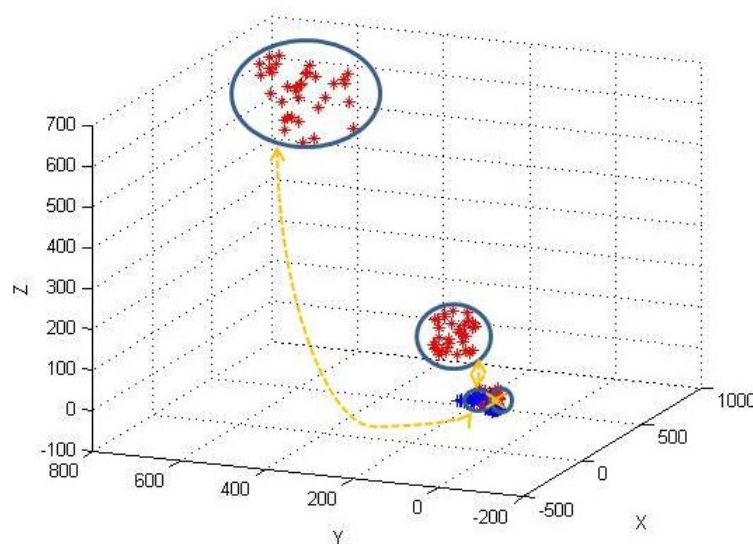
**Visualization.** To intuitively show the privacy-preserving clustering results over encrypted data, we simulate a data-set, in which we represent each point as a 3-dimension vector  $(X, Y, Z)$  ( $D = 3$ ). The data-set contains three clusters ( $k = 3$ ) and 100 data points ( $n = 100$ ). By using our PPK-means\_1 (PPK-means\_2, PPK-means\_3) and original  $k$ -Means, we cluster the encrypted data points and the original data points simultaneously. The clustering cases of three schemes are shown in Figures 4–6, respectively. In these figures, blue points represent encrypted points and red points represent original data points found by traditional  $k$ -means. The results demonstrate that, by PPK-means, the encrypted data points can be correctly clustered in different position spaces with original data. On the other hand, we can see that the relations between encrypted data and corresponding unencrypted data are more undistinguishable for PPK-means\_2 and PPK-means\_3, compared with PPK-means\_1, as they introduce *Random Asymmetric Splitting* and more random numbers. PPK-means\_3 in particular, we hardly label the relation between an original data cluster and its corresponding encrypted data clusters manually, when the original data are adequately randomized.

**Figure 4.** The examples of the clustered original points (red '\*') and the encrypted points (blue '+') by using PPK-means\_1.





**Figure 5.** The examples of the clustered original points (red ‘\*’) and the encrypted points (blue ‘+’) by using PPK-means\_2.



**Figure 6.** The examples of the clustered original points (red ‘\*’) and the encrypted points (blue ‘+’) by using PPK-means\_3.

### 6.2.3. Evaluation with Real-Life data-set

**data-sets.** To evaluate the practicability of our scheme, we use several different real-life data-sets from the Kaggle website [34] to run our prototype, which are used for heart disease expectations, cell classification, and airport locations. The details of these data-sets are shown in Table 3.

In our experiments, we split each data-set into two parts, one is used to train and the other is for detection. Each part contains half of the points (randomly selected) in the data-set. For the training data-sets, we use  $k$ -means to cluster data and generate  $k$  central points. For the detection data-sets, we use PPK-means to cluster the encrypted data.



**Table 3.** The real-life data-sets.

data-set	Number of Records	Number of Dimensions
heart disease	383	14
cells	15,608	6
airport locations	1435	2

**Time cost evaluation.** Table 4 shows the time cost of PPK-means\_3 on T-DO, T-DU, T-CS in these three real data-sets. The results show that our scheme is efficient and practical. For example, when the size of data-set achieves 15608 (cells data-set), the time of data clustering is only 47 milliseconds, while the time cost of model encryption and data encryption is almost 0. It is acceptable in the real application.

**Table 4.** The time cost evaluation of the real data-sets.

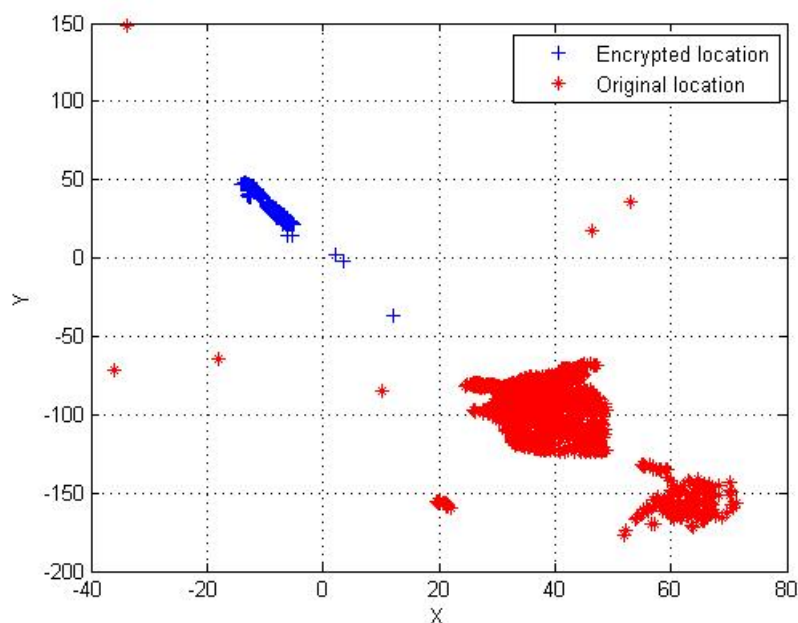
data-set	T-DO	T-DU	T-CS
heart disease	<1 ms	<1 ms	12 ms
cells	<1 ms	<1 ms	41 ms
airport locations	<1 ms	<1 ms	11 ms

**Functionality evaluation.** We evaluate the effectiveness of three schemes in real data-sets, as shown in Table 5, where the accuracy means the proportion of the encrypted data which is clustered into the correct clusters as that of plaintext data clustering using *k*-means. we can observe that, although PPK-means\_3 sacrifices a bit accuracy (less than 0.1%), it is still effective enough.

**Table 5.** The proportions of the correctly clustered encrypted points.

data-set	Accuracy		
	PPK-Means_1	PPK-Means_2	PPK-Means_3
heart disease	100%	100%	100%
cells	100%	100%	99.97%
airport locations	100%	100%	99.86%

**Visualization.** Since the numbers of dimensions of heart disease and cell data-sets are bigger than 3, we only print the original points and the encrypted points by PPK-means\_3 in the airport location data-set with dimension number 2, as shown in Figure 7. We can see that plaintext data and encrypted data are clustered in different position spaces, respectively. Moreover, it is very difficult to determine which one encrypted data corresponds to which one plaintext data in the whole clustering space. The result demonstrates that the original airport location data are protected well by our encryption scheme while achieving accurate data clustering.



**Figure 7.** The examples of the clustered original points (red ‘\*’) and the encrypted points (blue ‘+’) of the airport location.

## 7. Limitation and Conclusions

In this section, we state the limitations of our proposed scheme and conclude this paper.

### 7.1. Limitation

Although our scheme can achieve efficient clustering over encrypted data, there is one major limitation: in our system model, the data user does not send data to the data owner, so that the data owner cannot update the trained model by re-training with the data from the data user on the cloud server.

### 7.2. Conclusions

In this paper, we propose a privacy-preserving  $k$ -Means clustering scheme over encrypted multi-dimensional cloud data, called PPK-Means. The basic idea of PPK-Means is to employ scalar-product-preserving encryption to construct a privacy-preserving distance comparison between an encrypted trained model and encrypted user data, by which the cloud server is able to cluster the user’s data into correct clusters without knowing any useful information about the model and user data. As for the different attack model, we propose three PPK-Means schemes to guarantee the security of data under the different attack model. Detailed Security analyses and extensive experimental evaluations in simulation data-set and demonstrate using a real-life data-set, that our scheme is secure, correct, and practical. Finally, we give the limitations of our scheme. For the future, we will design a mechanism for privacy-preserving active re-training so that the PPK-Means can be continuously updated with the encrypted user’s data.

**Author Contributions:** Conceptualization, H.Y. and Y.X.; methodology, J.Z. and X.H.; software, J.Z. and Y.X.; validation, H.Y. and J.Z.; formal analysis, H.Y. and T.D.; investigation, H.Y. and J.Z.; resources, H.Y.; data curation, J.Z. and X.H.; H.Y. and T.D.; visualization, J.Z.; supervision, J.Z. and Y.X.; project administration, H.Y.

**Funding:** This research was funded by the Natural Science Foundation of Hunan Province under Grant No. 2017JJ2292; Science and Technology Key Projects of Hunan Province under Grant No. 2016JC2012; Outstanding Youth Research Project of Provincial Education Department of Hunan under Grant No. 17B030; Science and Technology Planning Project of Changsha under Grant No. K1705018, ZD1601042 and K1705031.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ren, K.; Wang, C.; Wang, Q. Security Challenges for the Public Cloud. *IEEE Internet Comput.* **2012**, *16*, 69–73. [[CrossRef](#)]
2. Kamara, S.; Lauter, K. *Cryptographic Cloud Storage*; Springer: Berlin/Heidelberg, Germany, 2010.
3. Graepel, T.; Lauter, K.; Naehrig, M. *ML Confidential: Machine Learning on Encrypted Data*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1–21.
4. Nikolaenko, V.; Weinsberg, U.; Ioannidis, S.; Joye, M.; Boneh, D.; Taft, N. Privacy-preserving ridge regression on hundreds of millions of records. In Proceedings of the 2013 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 19–22 May 2013; pp. 334–348.
5. Hu, S.; Wang, Q.; Wang, J.; Chow, S.S.M.; Zou, Q. Securing Fast Learning! Ridge Regression over Encrypted Big Data. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 19–26.
6. Bost, R.; Popa, R.A.; Tu, S.; Goldwasser, S. Machine Learning Classification over Encrypted Data. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 23–26 February 2014.
7. Samanthula, B.K.; Elmehdwi, Y.; Jiang, W. K-Nearest Neighbor Classification over Semantically Secure Encrypted Relational Data. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 1261–1273. [[CrossRef](#)]
8. Vaidya, J.; Clifton, C. Privacy-preserving k-means clustering over vertically partitioned data. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 206–215.
9. Jagannathan, G.; Wright, R.N. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 21–24 August 2005; pp. 593–599.
10. Beye, M.; Erkin, Z.; Lagendijk, R.L. Efficient privacy preserving K-means clustering in a three-party setting. In Proceedings of the IEEE International Workshop on Information Forensics and Security, Iguacu Falls, Brazil, 29 November–2 December 2011; pp. 1–6.
11. Gheid, Z.; Challal, Y. Efficient and Privacy-Preserving k-Means Clustering for Big Data Mining. In Proceedings of the IEEE Trustcom/BigdataSE/ISPA, Tianjin, China, 23–26 August 2017; pp. 791–798.
12. Dwork, C. Differential privacy: A survey of results. In Proceedings of the International Conference on Theory and Applications of MODELS of Computation, Xi'an, China, 25–29 April 2008; pp. 1–19.
13. Yan, S.; Pan, S.; Zhao, Y.; Zhu, W.T. Towards Privacy-Preserving data mining in Online Social Networks: Distance-Grained and Item-Grained Differential Privacy. In Proceedings of the Australasian Conference on Information Security and Privacy, Melbourne, Australia, 4–6 July 2016; pp. 141–157.
14. Xiong, X.; Chen, F.; Huang, P.; Tian, M.; Hu, X.; Chen, B.; Qin, J. Frequent Itemsets Mining with Differential Privacy over Large-scale Data. *IEEE Access* **2018**, *6*, 28877–28889. [[CrossRef](#)]
15. Yun, Y.E.; Shi, C.C.; Yong, Y.U.; Meng-Di, H.; Lin, W.M.; Peng, G. Privacy-Preserving Distributed Naive Bayes data mining. *J. Appl. Sci.* **2018**, *35*, 1–10.
16. Wright, R.; Yang, Z. Privacy-preserving Bayesian network structure computation on distributed heterogeneous data. In Proceedings of the International Conference on Knowledge Discovery and data mining (SIGKDD), Seattle, WA, USA, 22–25 August 2004; pp. 713–718.
17. Vaidya, J.; Kantarcoglu, M.; Clifton, C. Privacy-preserving naive bayes classification. *Int. J. Very Large Data Bases* **2008**, *17*, 879–898. [[CrossRef](#)]
18. Du, W.; Han, Y.S.; Chen, S. Privacy-Preserving Multivariate Statistical Analysis: Linear Regression And Classification. In Proceedings of the SIAM International Conference on data mining, Anaheim, CA, USA, 26–28 April 2012; pp. 222–233.
19. Liu, F.; Ng, W.K.; Zhang, W. Encrypted Gradient Descent Protocol for Outsourced data mining. In Proceedings of the IEEE International Conference on Advanced Information NETWORKING and Applications, Gwangju, Korea, 24–27 March 2015; pp. 339–346.
20. Liu, F.; Ng, W.K.; Zhang, W. Encrypted SVM for Outsourced data mining. In Proceedings of the IEEE International Conference on Cloud Computing, New York, NY, USA, 27 June–2 July 2015; pp. 1085–1092.

21. Sanil, A.P.; Karr, A.F.; Lin, X.; Reiter, J.P. Privacy preserving regression modelling via distributed computation. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 677–682.
22. Karr, A.F.; Lin, X.; Sanil, A.P.; Reiter, J.P. Secure regression on distributed databases. *J. Comput. Graph. Stat.* **2005**, *14*, 263–279. [CrossRef]
23. De Cock, M.; Nascimento, A.C.; Lin, S.C.N. Fast, Privacy Preserving Linear Regression over Distributed Datasets based on Pre-distributed Data. In Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, Denver, CO, USA, 16 October 2015; pp. 3–14.
24. MacQueen, J. Some Methods for classification and Analysis of Multivariate Observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21 June–18 July 1965; pp. 281–297.
25. Basu, S.; Banerjee, A.; Mooney, R.J. Active Semi-Supervision for Pairwise Constrained Clustering. In Proceedings of the International Conference on Data Mining, Brighton, UK, 1–4 November 2004; pp. 333–344.
26. Ahmadyfard, A.; Modares, H. Combining PSO and k-means to enhance data clustering. In Proceedings of the International Symposium on Telecommunications, Tehran, Iran, 27–28 August 2008; pp. 688–691.
27. K-Means. Available online: [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering) (accessed on 7 November 2018).
28. Wang, C.; Cao, N.; Li, J.; Ren, K.; Lou, W. Secure ranked keyword search over encrypted cloud data. In Proceedings of the 2010 IEEE 30th International Conference on Distributed Computing Systems, Genova, Italy, 21–25 June 2010; pp. 253–262.
29. Cao, N.; Wang, C.; Li, M.; Ren, K.; Lou, W. Privacy-preserving multi-keyword ranked search over encrypted cloud data. In Proceedings of the IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 829–837.
30. Li, H.; Yang, Y.; Luan, T.H.; Liang, X.; Zhou, L.; Shen, X. Enabling Fine-grained Multi-keyword Search Supporting Classified Sub-dictionaries over Encrypted Cloud Data. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 312–325. [CrossRef]
31. Yin, H.; Qin, Z.; Ou, L.; Li, K. A query privacy-enhanced and secure search scheme over encrypted data in cloud computing. *J. Comput. Syst. Sci.* **2017**, *90*, 14–27. [CrossRef]
32. Wong, W.; Cheung, D.; Kao, B.; Mamoulis, N. Secure knn computation on encrypted databases. In Proceedings of the ACM International Conference on Management of Data, SIGMOD, Providence, RI, USA, 29 June–2 July 2009; pp. 139–152.
33. Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement. *IEEE Trans. Parallel and Distrib. Syst.* **2016**, *27*, 2546–2559. [CrossRef]
34. Kaggle. 2018. Available online: <http://www.kaggle.com> (accessed on 15 September 2018).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).