Efficient and Privacy-Preserving k-Means Clustering for Big Data Mining

Zakaria Gheid, Yacine Challal

Efficient and Privacy-Preserving k-means clustering For Big Data Mining

Zakaria Gheid*§, Yacine Challal*‡
*Ecole nationale supérieure d'informatique
Laboratoire des Méthodes de Conception des Systèmes, Algiers, Algeria
‡Centre de Recherche sur l'Information Scientifique et Technique, Algiers, Algeria
Email: §z_gheid@esi.dz , ‡y_challal@esi.dz

***Abstract*—Recent advances in sensing and storing technologies have led to big data age where a huge amount of data are distributed across sites to be stored and analysed. Indeed, cluster analysis is one of the data mining tasks that aims to discover patterns and knowledge through different algorithmic techniques such as k-means. Nevertheless, running k-means over distributed big data stores has given rise to serious privacy issues. Accordingly, many proposed works attempted to tackle this concern using cryptographic protocols. However, these cryptographic solutions introduced performance degradation issues in analysis tasks which does not meet big data properties. In this work we propose a novel privacy-preserving k-means algorithm based on a simple yet secure and efficient multi-party additive scheme that is cryptography-free. We designed our solution for horizontally partitioned data. Moreover, we demonstrate that our scheme resists against adversaries passive model.**

***Index Terms*—big data, horizontally partitioned data, k-means clustering, privacy, efficiency.**

1. Introduction

Over the last years, world's generated data have grown in their scope and size, shifting from centralized processing to distributed environments, and leading so to the big data era. Big data is a term used to designate large or complex data sets that are beyond the ability of traditional data processing and methods. The emergence of big data sets have created exciting opportunities to maximize the knowledge available for analysts, researchers and business people, allowing to integrate big data analytics in decision making by uncovering hidden patterns, unknown correlations and other insights.

Moreover, big data analytics is performed throw several advanced data mining techniques such as clustering. This analysis task consists of discovering patterns from a set of data objects, then affecting each object to the closest pattern. Working on automating such a task has produced several methods known as unsupervised learning [1] such as k-means algorithm [2]. Indeed, big data clustering is now a keystone requirement for several areas of life like healthcare, social science, business and marketing. For instance, in cancer diagnosis we can take known samples of cancerous and non-cancerous data sets and apply clustering algorithms over patients' medical records (PMR) to identify cancerous data [3], [4]. For this, maximizing the number of sample sets by establishing a collaboration between several hospitals may significantly improve the accuracy of the results. Another practical scenario may consist of different telephone companies that need to set up towers in a common region they acquired. The optimum location of these common towers can be found through clustering algorithms so as to maximize the signal strength for each network's users. In social network analysis, clustering users' publications may help in recognizing communities with dense friendships internally and sparse friendships externally. Social network clustering can help in designing marketing plans [5], identifying terrorist cells [6] and other useful applications. Hence, putting data sets got from different social networks together should deliver more value to the analysis.

Further, the effective integration of big data mining has given rise to privacy issues surrounding disclosing personal private data during analysis process in distributed environments. In fact, personal opinions, political interests, healthcare records and other private data are being shared between institutions and service providers to improve accuracy of the clustering task.

From the perspective of research, many proposed works attempted to reach a partial privacy protection by adding some noise before sharing data [7], [8], [9]. However, the minimum error rate raised by the added noise is intolerable for applications needing a high accuracy level, such as healthcare. From another side other works have implemented homomorphic encryption schemes or oblivious transfert protocols [10] in order to securely share personal data for clustering. Nevertheless, these latter security measurements had inevitably result in serious degradation of performance especially for big data sets [11], [12].

In this work, we propose an efficient k-means protocol that aims to ensure total privacy protection under a given security model without using any cryptographic scheme. The contribution of this work can be summarized as follows

- We present (Π-sum): a privacy-preserving and efficient multiparty additive scheme based on the famous Clifton's secure sum [13] and modified so as to escape the "maximum" barrier assumpted by [13].

- We sketch (sk-means): a secure implementation of the k-means algorithm over horizontally partionned data and based on Π-sum.
- We evaluate the security of our proposals using the real/ideal standard paradigm [14] and we proof their performance efficiency compared to other methods through different experimental tests.

The rest of the paper is organized as follows. In section 2 we present preliminaries and notations used to introduce our proposals. In section 3 we highlight the privacy concern in k-means process and we present two privacy-peserving protocols in order to efficiently tackle this issue. In section 4 we give a formal security proof of our solution using the real/ideal paradigm and we devote section 5 to the performance analysis through different experiments. We give a literature survey of related works in section 6 and we conclude by summarizing the contributions this work.

2. Preliminaries & terminology

In this section we introduce preliminaries and notations used later to implement and analyse our proposal.

2.1. k-means clustering algorithm

Recall that data clustering is a task of data mining that consists of partitioning a collection of data sets into separated groups called clusters in a way that maximizes the similarity of the objects in each cluster. Euclidian distance and cosine similarity [15] are some metrics used to evaluate similarity between objects in a cluster. K-means [2] is one of the most widely used algorithm [16] to produce automatically k clusters from a collection of data sets in a simple way. A brief description of k-means implementation is presented in Algorithm1.

Algorithm 1: k-means clustering

1: Randomly select k cluster centers $\{c_1, ...c_k\}$.
2: **repeat**
3: Assign each data entity to the closest cluster center c_i.
4: Replace each cluster center c_i by the mean of the cluster i.
5: **until** cluster centers do not change.

2.2. Big data properties

Big data mining requires additional constraints than traditional data mining tasks in order to better handle its main characteristics, which are illustred in Figure 1.

2.2.1. High volume. Big data is a great amount of datasets. Thus, using a method which performs a privacy-preserving clustering without inducing a high computation cost is a keystone requirement.

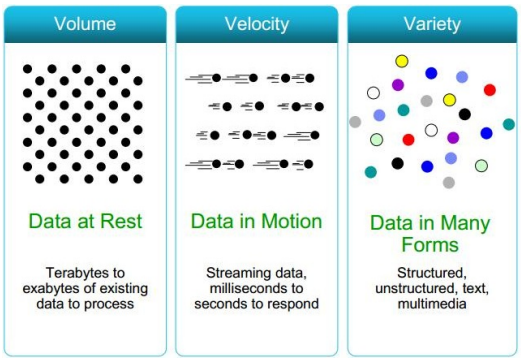


Figure 1. Big data main characteristics: volume, variety and velocity

2.2.2. High velocity. Big data are subjected to a high velocity stream of input and output queries which requires in some contexts real time interactions. Thus, introducing privacy preserving mechanisms should not induce unaffordable computation overhead.

2.2.3. High variety. Big data stores involve data sets of different sizes that originate from several sources. So, we need to address this variety in the big data clustering analysis.

2.3. Data distribution model

In a distributed environment, a data store is splitted over sites into multiple units called fragments. data fragmentation can be done in three ways, namely, horizontal, vertical and mixed distributions [17]. Let S be a set of k data objects $(d_1, ..., d_k)$ each of which has n attributes $(a_1, ..., a_n)$

Definition 1. *A horizontal distribution produces fragments S_i of m_i ($m_i < k$) data objetcs each of which has n attributes $(a_1, ..., a_n)$. Each fragment S_i can be get by the selection operation (σ) of the relational algebra using a criterion c_i as $\sigma_{c_i}(S)$.*

Definition 2. *A vertical distribution of S produces fragments S_i of k data objects $(d_1^i, ..., d_k^i)$ having a subset of the n original attributes. Each S_i can be get by the projection operation (π) of the relational algebra using a subset of attributes a_i ($1 \leq i \leq n$) such as $\pi_{a_1,a_2,...,a_m(m \leq n)}(S)$.*

Definition 3. *A mixed distribution of S is obtained using a horizontal distribution followd by a vertical distribution or a vertical destribution followed be a horizontal distribution. Using the precedent examples we can define it in the relational algebra as $\sigma_{c_i}(\pi_{a_1,a_2,...,a_m(m \leq n)}(S))$ or $\pi_{a_1,a_2,...,a_m(m \leq n)}(\sigma_{c_i}(s))$.*

In this work, we will propose security for the horizontal distribution model.

2.4. Multiparty computation

Let us consider a set of participants that want to jointly compute the value of a public function f relying on their private data. Let P_1,...,P_n denote the participants and v_1,...,v_n

their private data respectively. We call MPC model the running process of $f(v_1, ..., v_n)$ [18]. Let Π denote a multiparty protocol executed by n participants (P_1,...,P_n) in order to evaluate the function f and let v denote the set of inputs $(v_1, ..., v_n)$. We consider n as security parameter and we will give security proof relying on the assumption: $n > 2$.

Notation 1. *Let $view_X^{\Pi}(v,n)_i$ denote the set of messages get by the party $P_{i\in\{1,...,n\}}$ during the execution X of Π on the set of inputs v and security parameter n.*

Notation 2. *Let $out_X^{\Pi}(v,n)_i$ denote the output of the party $P_{i\in\{1,...,n\}}$ by the execution X of the protocol Π on the set of inputs v and security parameter n. Let $out_X^{\Pi}(v,n)$ denote the global output of all collaborating parties from the same execution of Π, where*

$$out_X^{\Pi}(v, n) = \cup_{i=1}^{n} out_X^{\Pi}(v, n)_i$$

In next section, we introduce a novel privacy-preserving clustering protocol built on a secure multiparty additive scheme. We will use these MPC notations later, to prove the security of our proposal.

3. A novel efficient and privacy-preserving k-means clustering

3.1. Privacy issue in k-means algorithm

In a horizontal distributed environment (see Definition1), sites wanting to participate in a k-means clustering task need to collaborate (see task 4, Algorithm 1) in order to compute means of the clusters that may involve data objects from different sites. Assume a cluster i includes n data objects $\{d_1, ..., d_n\}$ originating from m sites $\{P_1, ..., P_m\}$. To get the mean of the cluster i (which will be the novel cluster center c_i), participants need to evaluate the following computation

$$c_i = \frac{\sum_{j=1}^{m} (\sum_{d_i \in P_j} d_i)}{\sum_{j=1}^{m} card(i, j)} \quad (1)$$

where $card(i, j)$ denotes the number of data objetcs originating from the participant P_j. This collaborative computation requires from participants to send their sum of data objects (numerator) as well as their cardinalities (denominator) to each other, which may cause a privacy breach when it comes to private data such as medical records. Thus, to preserve privacy in a horizontal distributed k-means execution we will present a privacy-preserving protocol that securely assesses this mean computation (equation (1)) based on a simple multiparty additive scheme.

3.2. sk-means: a privacy-preserving and efficient k-means protocol for horizontally partitioned data

Throughout the litterature survey we present later in this paper, only few works ave targeted horizontally partitioned data. Contrary to precedent works [19] that aim to secure the whole fraction (see equation 1), The main idea of our scheme consists of splitting the calcution of each cluster mean into two sum operations (numerator and denominator). Thus, we reduce the need for secure multiparty mean computation to the need for secure multiparty sum. This reduction will induce a high level of efficiency that copes with big data properties. Let us assume m participants $\{P_1, ..., P_m\}$ in a k-means clustering task, each of which has $n_{i(1\leq i\leq m)}$ data objects. Let $sum(i, j)$ denote the sum of data objects involved in the cluster i and originating from P_j and let $card(i, j)$ denote the number of these data objects. Using the secure mutiparty addition primitive named Π-sum and detailed in section 3.3, we present sk-means, a secure k-means protocol implemented in Algorithm 2.

Algorithm 2: sk-means, a privacy-preserving and efficient k-means protocol

Variables:
k: number of clusters
m: number of participants
i: the cluster index
$sum(i, j)$: sum of data objects of P_j involved in the cluster i
$card(i, j)$: number of data objects of P_j involved in the cluster i

1: Randomly select k cluster centers $\{c_1, ...c_k\}$.
2: **repeat**
3: Assign each data object to the closest cluster center c_i (Performed by each P_j locally).
4: **for** $(i = 1; i \leq k; i++)$ **do**
5: $sum_1 \leftarrow \Pi$-sum $(sum(i, 1), ..., sum(i, m))$
6: $sum_2 \leftarrow \Pi$-sum $(card(i, 1), ..., card(i, m))$
7: $c_i \leftarrow \frac{sum1}{sum2}$.
8: **end for**
9: Sharing the novel k cluster centers $\{c_1, ...c_k\}$.
10: **until** cluster centers do not change.

3.3. Π-sum: a privacy-preserving and efficient multiparty additive scheme

In what follows we present Π-sum, a privacy-preserving and efficient multiparty sum protocol, which is free from cryptographic operations. Π-sum is built on the secure sum protocol proposed by Clifton et al. [13], working in the same security model that we explore in the next section and having less restrictions on data inputs in order to cope with big data high variety. Assume m parties (P_1,...,P_m) having respectively (v_1,...,v_m) private values and wanting to evaluate

$$v = \sum_{i=1}^{m} v_i \quad (2)$$

Assuming no collusion, Clifton's secure sum [13] operates by hiding $v1$ with a random number r chosen uniformely from the range $[0..n]$ by P_1, then, sending $(r + v_1 \bmod n)$

to P_2, which adds v_2 and sends the result to P_3 and so on until the end. The protocol could work only under the assumption given in [13] that the value v is known to be in the range $[0..n]$. This is a hard restriction for big data stores that may involve uncontrollable values. Through Π-sum that we present in Algorithm 3, we show that sending a number r randomly chosen from $\mathbb{R}$ is sufficient for securing this type of scheme and we prove its security in the next section.

Algorithm 3: Π-sum (v_1,...,v_m), a privacy-preserving and efficient multiparty addition

Input : v_i: the private value of P_i, $v_{i(1\leq i\leq m)} \in \mathbb{R}^n$
m: the number of participant parties
S_i: the secret of P_i
r: random variable $\in \mathbb{R}^n$ generated by P_1
Output: $v = \sum_{i=1}^{k} vi$
Step 1 by P_1
1: Generates a random $r \in \mathbb{R}^n$
2: $S_1 \leftarrow r$
3: P_1 shares S_1 with P_2
Step 2 by $\cup_{i=2}^{m} P_i$
4: **for** $i = 2$ to $m - 1$ **do**
5: $S_i \leftarrow S_{i-1} + v_i$
6: P_i shares S_i with P_{i+1}
7: **end for**
8: P_m shares S_m with P_1
Step 3 by P_1
9: $S_1 \leftarrow S_m + (v_1 - r)$
10: **return** S_1 in broadcast.

4. Security analysis

In this section we prove the security of our scheme in a given security model according to the real/ideal simulation paradigm [14].

4.1. Adversarial model

In multiparty computation, there are two main types of adversaries, namely, passive and active, according to the allowed behaviours of their corrupted participants in the computation [18]. In this work we give security proof against passive adversaries.

4.1.1. Passive adversary. (Also called semi-honest) In this model of adversary, corrupted parties follow the protocol specifications but they are allowed to learn information from the messages they receive during the execution of the protocol.

4.1.2. Active adversary. (Also called malicious) For this model, there are no suppositions on the behaviour of corrupted parties and they are allowed to randomly deviate from the protocol according to the adversary's instructions.

4.2. Security model

In this subsection we introduce the real/ideal simlation paradigm [14]. Let Π denote a mutipaty protocol executed by m participants (P_1,...,P_m) in order to evaluate a function f. Let B denote the class of adversary that may corrupt participants in Π such as $B \in \{active, passive\}$. Let R and L denote respectively the real and the ideal executions of Π on the set of inputs v and security parameter m.

During a **real execution** (R) we consider the presence of an adversary denoted A that behaves according to the class B while corrupting a set of participants $P_{i(1\leq i\leq m)}$. At the end of R, uncorrupted parties output whatever was specified in Π and the corrupted P_i outputs any random functions of their $view_R^{\Pi}(v,m)_i$.

During an **ideal execution** (L) we consider the presence of a trusted incorruptible party denoted T, which receives the set of inputs v from all participants in order to evaluate the function f in the presence of an adversary denoted S. We assume S corrupts the same P_i as the adversary A of the correspondant real execution, and behaves according to the same class B before sending inputs to T. By the end of L, uncorrupted participants output what was received from T and the corrupted P_i output any random functions of their $view_L^{\Pi}(v,m)_i$.

Definition 4. *Let* Π *and* f *be as above. We consider* Π *a secure multiparty protocol if for any real adversary* A *having a class* B *and attacks the protocol* Π*, there exists an adversary* S *in the ideal execution having the same class* B *and that can emulate any effect achieved by* A*. Let* $\stackrel{d}{\equiv}$ *denote the distribution equality. We formalize this security definition as follows*

$$\{out_R^{\Pi}(v,m)\} \stackrel{d}{\equiv} \{out_L^{\Pi}(v,m)\} \tag{3}$$

4.3. Real/ideal proof

In what follows we give simulations for real and ideal execution to prove the security of Π-sum prtocol. Then, we deduce the security of sk-means protocol.

4.3.1. Π-sum security proof.

Theorem 1. *Assume* m ($m > 2$) *participants* (P_1,...,P_m) *having respectively* (v_1,...,v_m) *private values and assuming no collusion. Then,* Π*-sum* (v_1,...,v_m) *is a secure MPC protocol in the presesence of a passive adversary.*

Proof. We consider restricting the adversarial class (B) to passive adversary and that is allowed to corrupt one participant at a time since no collusion is assumed. Let A, S and T be as above. Let Π denote Π-sum protocol and v denote the set of inputs (v_1,...,v_m) where $v_{i(1\leq i\leq m)} \in \mathbb{R}^n$. In this adversarial model (passive) the simulation is trivial, since any corrupted $P_{i(1\leq i\leq m)}$ is assumed following Π.

Assume A corrupts a $P_{i(1\leq i\leq m)}$, then, S will just handle P_i's input and sends it to T, thereby, completing the simulation. By the end, T performs the sum computation of all

received inputs and sends back the result to each participant. The security proof of this simulation relies on four different cases as follows

- **Case 1**: P_1 is corrupted. In this case the views of P_1 are described as follows

$$view_R^{\Pi}(v,m)_1 = \{r, v_1, v, S_1, S_m\} \quad (4)$$
$$view_L^{\Pi}(v,m)_1 = \{r, v_1, v\} \quad (5)$$

  Since we defined the security parameter $m > 2$, we have according to Π (Algorithm 3)

$$S_m = S_{m-1} + v_m \quad (6)$$

  So, S_m will not reveal any information for P_1 since it involves at least two unknowns (S_{m-1} and v_m). Likewise, S_1 will not involve additional information than the random value r. Then, we can reduce (4) as follows

$$view_R^{\Pi}(v,m)_1 = \{r, v_1, v\} \quad (7)$$

  Thus, according to (5) and (7), we can deduce

$$\{out_R^{\Pi}(v,m)\} \stackrel{d}{\equiv} \{out_L^{\Pi}(v,m)\} \quad (8)$$

- **Case 2**: P_2 is corrupted. In this case the views of P_2 are described as follows

$$view_R^{\Pi}(v,m)_2 = \{v_2, v, S_1, S_2\} \quad (9)$$
$$view_L^{\Pi}(v,m)_2 = \{v_2, v\} \quad (10)$$

  As S_1 involves the random value r, then

$$v = S_m - S_1 + v_1 \quad (11)$$

  But, since we define the security parameter $m > 2$, S_1 will be useless for P_2 to disclose v_1 because (11) will involve at least two unknowns ($S_{m \notin \{1,2\}}$ and v_1). Consequently, S_2 will not involve additional information for P_2. Then, we can reduce (9) as follows

$$view_R^{\Pi}(v,m)_2 = \{v_2, v\} \quad (12)$$

  Thus according to (10) and (12) we can deduce the same (8).
- **Case 3**: P_m is corrupted. Then, we describe the views of P_m as follows

$$view_R^{\Pi}(v,m)_m = \{v_m, v, S_m, S_{m-1}\} \quad (13)$$
$$view_L^{\Pi}(v,m)_m = \{v_m, v\} \quad (14)$$

  But since we defined security parameter $m > 2$, S_{m-1} will be other than S_1, then $v = S_m - S_1 + v_1$ will not reveal any information for S_m since S_1 and v_1 are two unknowns. Consequently, S_m as well as S_{m-1} will not include additional information for P_m. Thus, we can reduce (13) from S_m and S_{m-1}, then, we deduce the same (8).
- **Case 4**: $P_{i(2<i<m)}$ is corrupted. In this case the views of P_i are described as follows

$$view_R^{\Pi}(v,m)_i = \{v_i, v, S_i, S_{i-1}\} \quad (15)$$
$$view_L^{\Pi}(v,m)_i = \{v_i, v\} \quad (16)$$

  With the same reasoning logic as case 3, we can reduce (15) from S_i and S_{i-1} in ordr to deduce the same (8).

Throughout these possible corruption cases, we have proved that any information that can be output (learned) by a corrupted participant $P_{i(1\leq i\leq m)}$ in a real execution of Π-sum can also be output (learned) in ideal execution according to the passive adversarial model. □

Note 1. *Notice that we have not considered outputs of uncorrupted parties during the security proof of the diferent cases, because they are never affected in the passive adversarial model.*

4.3.2. sk-means security proof.

Corollary 1. *Assume m ($m > 2$) participants (P_1,...,P_m) in a multiparty clustering task and assuming no collusion. Then, running sk-means is a secure MPC in the presence of a passive adversary.*

Proof. As the call to Π-sum is the only multiparty task within sk-means (see Algorithm 2), we can deduce the security of sk-means relying on Theorem 1 proved above. □

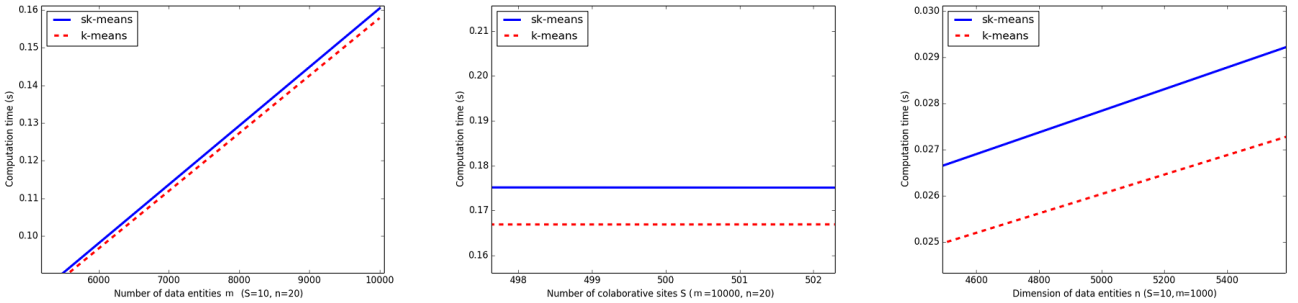
5. Performance evaluation

In this section we will demonstrate the efficiency of our propsed sk-means protocol thus, its suitability for big data mining.

5.1. Simulation model and scenarios

In order to prove the efficiency of the sk-means protocol (Algorithm 2) based on Π-sum (Algorithm 3) we will evaluate the impact of big data volume (number of data objects) and big data variety (size and source of data objects) on sk-means running time and we compare it to the most recent secret sharing-based protocol for horizontally data distribution proposed by Patel et al. [11]. We have avoided to test protocols based on oblivious transfer and homomorphic encryption because of their inadequacy for big data analytics as was reported by different works [11], [12]. For experimental purpose and without loss of generality, we will simulate the computation of only one cluster mean rather than simulating the hole sk-means clustering process.

We assume a cluster i involves m data objects (d_1,...,d_m) each of which has the size of n attributes and originating all from s participant sites. For simplicity reason we assume each site has $\frac{m}{s}$ data entities involved in the cluster i. We perform three experiments, namely, $E1$, $E2$ and $E3$ in order to evaluate respectively the impact of: a) the number of data objects (m), b) the number of collaborative sites (s) and



(a) Number of data objects(m) (b) Number of collaborative sites (S) (c) Size (dimension) of data objects (n)

Figure 2. Impact of (m), (S) and (n) parameters on the running time of sk-means protocol compared to the native k-means execution

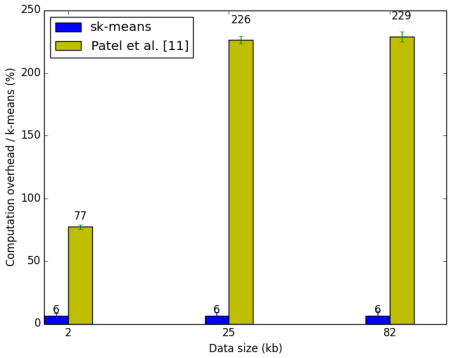


Figure 3. Running time percentage with respect to the native k-means

c) the size of data objects (n) on running time. In figure 2 we plot results of $E1$, $E2$ and $E3$ besides the running time of a native k-means clustering process (without any security measurments) as a reference time. We plot results of comparison with the protocol proposed by Patel [11] in figure 4 and we give more detailed results in table 1 and table 2.

Regarding the evaluation environment, we make each experiment on the same data sets through a simulator built in Python and executed in an Intel i5-2557M CPU running at 1.70 GHz and having a 4 GB of RAM. We stress that despite the low communication cost of our proposal, we are not considering communication costs in comparison.

5.2. Results and discussion

5.2.1. Results of E_1. In E_1 we attempted to test the impact of big data volume on running time of sk-means compared to the standard k-means. For this, we fixed s and n parameters to 10 and 20 respectively and we vary the number of data entities (m) in the range [100,10000]. Results that are shown in table 1 and illustrated in figure 2a reveal the efficiency of sk-means with an almost constant distance from k-means running time. The overhead rate induced by sk-means remains stable in the neiberhood of (1.0x) $\times$ (k-means running time).

Table 1.

EXPERIMENTAL EVALUATION RESULTS

Data entities number (m) with s=10 and n=20	100	1000	5000	10000
k-means running time (second)	0.0016	0.017	0.081	0.157
sk-means running time (second)	0.0017	0.018	0.082	0.159
Ratio	1.06	1.06	1.01	1.01
Number of sites (S) with m=10000 and n=20	20	150	200	250
k-means running time (second)	0.17	0.17	0.17	0.17
sk-means running time (second)	0.170	0.173	0.177	0.177
Ratio	1	1.01	1.04	1.04
Dimension of data entities (n) with s= 10 and m=1000	50	100	500	5000
k-means running time (second)	0.016	0.017	0.018	0.026
sk-means running time (second)	0.017	0.018	0.019	0.028
Ratio	1.06	1.06	1.06	1.07

Table 2.

RUNNING TIME PERCENTAGE INCREASE WITH RESPECT TO K-MEANS

Data size (Kb)	2	25	82
S Patel [11] computation overhead (%)	77.26	226.23	229.05
sk-means computation overhead (%)	6.25	6.25	6.25

5.2.2. Results of E_2. Through E_2 we have evaluated the impact of big data variety in data sources on running time of k-means and sk-means. To do this, we varied the number of participant sites (s) in the range [20,600] and we fixed m and n to 10000 and 20 respectively. Results that are presented in table 1 and illustrated in figure 2b reveal the efficiency of sk-means with an almost constant distance from k-means. Augmenting s has no effect on sk-means running time because in the Π-sum protocol, random number (r) is generated once only by the participant P_1. The augmentation in sk-means running time observed is due to running

environment and it is not significant, hence, the overhead rate induced by sk-means is still stable in the neiberhood of (1.0x) $\times$ (k-means running time).

5.2.3. Results of E_3. In E_3 we aimed to evaluate the impact of big data variety in objects' size. We fixed m and s to 1000 and 10 respectiveley and we have varied the size of data objects (n) in the range [50,10000]. Results shown in table 1 and illustrated in figure 2c are showing the efficiency of sk-means with a stable and not significant computation overhead compared to the native k-means.

5.2.4. sk-means vs. Patel et al. [11]. We have compared sk-means running time to the recent privacy-preserving k-means protocol proposed by Patel et al. [11]. We took data sets of the same size that considered in the evaluation presented in [11] under the assumption that each data attribute is coded in one byte, i.e. each data object requires n bytes. We make comparaison of the computation overhead rate of the two protocols with respect to the native k-means protocol. We note that we have taken evaluation performance of the protocol [11] when executed under the passive adversarial model as reported by Patel et al. [20]. Results that are shown in table 2 and illustrated in figure 4 reveal the high level of efficiency provided by sk-means. We observe that for small data objects, sk-means protocol outperforms [11] by a magnitude of (12$\times$), this outperformance increases with the increase of the size of data objects until reaching (36$\times$). We note that the stable rate of sk-means overhead is due to its independance from the size of data objects, which has been proved through E_2.

6. Related works

Privacy preserving data mining methods had been surveyed in different works, we cite as examples Verykos et al. [21], Vaidya et al. [22], Aggrawal et al. [23] and a more recent one given by Meskine et al. [24]. In this state-of-the art section, we make focus on the privacy protection in the data clustering task performed across k-means algorithm in a distributed environment.

From a literature survey we observe that all proposed works could be classified according to their privacy level into two categories: those proposing a partial privacy protection and others achieving a complete privacy protection.

The first category proceeds by introducing a noise to the shared data [7], then, clusters the noisy data with a minimum error rate. Works of Bunn et al. [8] and Oliveira et al. [9] present some contributions that fall in this category. Even these methods do not induce computation overhead, they are compromising privacy with the trustworthiness of the result. Such a compromise is unacceptable for applications where a complete accuracy or a complete privacy are critical.

The second category attain a complete privacy using three main tools that consist of: oblivious transfer protocols [10], homomorphic cryptosystems and secure sharing schemes. The two first techniques have been largely implemented because of the high level of security they provide. Vaidya et al. [25] gave the first common cited work in which they attempted to secure vertical data distributions. For this, they implemented the secure permutation of Du et al. [26], the Yao's evaluation circuit [27], besides some homomorphic encryption schemes. This pile of primitives resulted in a high computation cost and reduced the scalability of this method for big data sets. Jha et al. [28] proposed two protocols for vertically partitioned data based on oblivious polynomial evaluation [10] and homomorphic encryption schemes. Each of these techniques has a high communication and computation cost when executed on large datasets [11], [24], in addition, they operate only on two parties and can not be extended for multiple participants. Contrary to precedent works, Jaganhatan et al. [29] targeted mixed partitioned data by using the Yao's circuit and the secure scalar product [30]. Because they used these security subroutines expensively [24] their method seems to be impractical for big data sets. Bunn et al. [8] proposed a more efficient protocol for mixed data distributions. Yet, they implement some time expensive operations such as the Paillier cryptosystems [31]. Samet et al. [19] proposed two protocols for both vertically and horizontally data distributions. They implement a secure multi-party addition primitive and present an application example for two or more parties. Nonetheless, we can simply demonstrate that any multiparty addition scheme is unsecure for two-party execution because of the bijective property of the addition (a participant secret can be got from the result by subtracting the local value).

Furthermore, few solutions have been proposed based on secure sharing paradigm. Doganay et al. [32], Upmanyu et al. [33] and Jinwala et al. [34] are almost the only works that belong to this category. They proposed different protocols based on some secure sharing schemes in order to preserve privacy under a semi-honest model of adversaries. Recently, Patel et al. [11] proposed a privacy protection that resists to malicious adversaries based on the shamir's secret sharing scheme [35].

By summarizing we can deduce the lack of techniques that can provide a complete privacy while remaining efficient to cope with big data properties. It has been proved [11], [12] that encryption-based techniques are not suitable for big data sets while the secret sharing schemes are more promising for mining large datasets because of their low and efficient computation and communication costs [11].

In this paper we presented a simple, efficient and privacy-preserving k-means protocol based on a multiparty additive scheme. We targeted horizontally partitioned data under the passive adversarial model and we tackle the problem of big data high requirements in an efficient way.

7. Conclusion

In this paper we proposed sk-means: a novel efficient and privacy-preserving protocol for k-means clustering founded heaviliy on a secure and simple multiparty additive protocol named Π-sum. Through different evaluations we proved the security of sk-means and Π-sum as well as their simplicity

and efficiency compared to other propositions. These preliminary results demonstrate that our solution suites better to big data properties and scales to large data sets as shown in experimental tests where we demonstrate that performance (computation overheades) of our solution is independent from data entity sizes and the induced computation overhead does not exceed the ratio of (1.0x) of the native k-means, yet providing privacy-preserving of cluster centers computation.

References

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Unsupervised learning*. Springer, 2009.

[2] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[3] XY Wang and Jon M Garibaldi. A comparison of fuzzy and non-fuzzy clustering techniques in cancer diagnosis. In *Proceedings of the 2nd International Conference in Computational Intelligence in Medicine and Healthcare, BIOPATTERN Conference, Costa da Caparica, Lisbon, Portugal*, page 28, 2005.

[4] Mark Girolami and Chao He. Probability density estimation from optimally condensed data samples. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1253–1264, 2003.

[5] Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E Tarjan. Clustering social networks. In *Algorithms and Models for the Web-Graph*, pages 56–67. Springer, 2007.

[6] Valdis Krebs. Uncloaking terrorist networks. *First Monday*, 7(4), 2002.

[7] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 99–106. IEEE, 2003.

[8] Paul Bunn and Rafail Ostrovsky. Secure two-party k-means clustering. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 486–497. ACM, 2007.

[9] Stanley RM Oliveira and Osmar R Zaiane. Privacy preserving clustering by data transformation. *Journal of Information and Data Management*, 1(1):37, 2010.

[10] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 245–254. ACM, 1999.

[11] Sankita Patel and Devesh C Jinwala. Privacy preserving distributed k-means clustering in malicious model, 2013.

[12] Rongxing Lu, Hui Zhu, Ximeng Liu, Joseph K Liu, and Jun Shao. Toward efficient and privacy-preserving computing in big data era. *Network, IEEE*, 28(4):46–50, 2014.

[13] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y Zhu. Tools for privacy preserving distributed data mining. *ACM Sigkdd Explorations Newsletter*, 4(2):28–34, 2002.

[14] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of CRYPTOLOGY*, 13(1):143–202, 2000.

[15] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, pages 58–64, 2000.

[16] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[17] S Navathe, Kamalakar Karlapalem, and Minyoung Ra. A mixed fragmentation methodology for initial distributed database design. *Journal of Computer and Software Engineering*, 3(4):395–426, 1995.

[18] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1):5, 2009.

[19] Saeed Samet, Ali Miri, and Luis Orozco-Barbosa. Privacy preserving k-means clustering in multi-party environment. In *SECRYPT*, pages 381–385, 2007.

[20] Sankita Patel, Viren Patel, and Devesh Jinwala. Privacy preserving distributed k-means clustering in malicious model using zero knowledge proof. In *Distributed Computing and Internet Technology*, pages 420–431. Springer, 2013.

[21] Vassilios S Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *ACM Sigmod Record*, 33(1):50–57, 2004.

[22] Jaideep Vaidya. A survey of privacy-preserving methods across vertically partitioned data. In *Privacy-preserving data mining*, pages 337–358. Springer, 2008.

[23] Charu C Aggarwal and S Yu Philip. *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[24] Fatima Meskine and Safia Nait Bahloul. Privacy preserving k-means clustering: a survey research. *Int. Arab J. Inf. Technol.*, 9(2):194–200, 2012.

[25] Jaideep Vaidya and Chris 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*, pages 206–215. ACM, 2003.

[26] Wenliang Du and Mikhail J Atallah. Privacy-preserving cooperative statistical analysis. In *Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual*, pages 102–110. IEEE, 2001.

[27] Andrew Yao. How to generate and exchange secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986.

[28] Somesh Jha, Luis Kruger, and Patrick McDaniel. Privacy preserving clustering. In *Computer Security–ESORICS 2005*, pages 397–417. Springer, 2005.

[29] Geetha Jagannathan and Rebecca N Wright. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 593–599. ACM, 2005.

[30] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. On private scalar product computation for privacy-preserving data mining. In *Information Security and Cryptology–ICISC 2004*, pages 104–120. Springer, 2005.

[31] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*, pages 223–238. Springer, 1999.

[32] Mahir Can Doganay, Thomas B Pedersen, Yücel Saygin, Erkay Savaş, and Albert Levi. Distributed privacy preserving k-means clustering with additive secret sharing. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, pages 3–11. ACM, 2008.

[33] Maneesh Upmanyu, Anoop M Namboodiri, Kannan Srinathan, and CV Jawahar. Efficient privacy preserving k-means clustering. In *Intelligence and Security Informatics*, pages 154–166. Springer, 2010.

[34] Neha B Jinwala and Gordhan B Jethava. Privacy preserving using distributed k means clustering for arbitrarily partitioned data. In *International Journal of Engineering Development and Research*, volume 2. IJEDR, 2014.

[35] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.