Privacy-Awareness of Distributed Data Clustering Algorithms Revisited

Josenildo C. da Silva[1], Matthias Klusch[2], and Stefano Lodi[3]

[1] Instituto Federal do Maranhão (IFMA), Depto. de Informática
Av. Getúlio Vargas, 04, Monte Castelo, CEP 65030-005 São Luís, MA, Brasil
jcsilva@ifma.edu.br
[2] DFKI GmbH Stuhlsatzenhausweg 3, Campus D3.2
D-66123 Saarbrücken
klusch@dfki.de
[3] Dipartimento di Informatica - Scienza e Ingegneria
Viale Risorgimento 2, Bologna
stefano.lodi@unibo.it

Abstract. Several privacy measures have been proposed in the privacy-preserving data mining literature. However, privacy measures either assume centralized data source or that no insider is going to try to infer some information. This paper presents distributed privacy measures that take into account collusion attacks and point level breaches for distributed data clustering. An analysis of representative distributed data clustering algorithms show that collusion is an important source of privacy issues and that the analyzed algorithms exhibit different vulnerabilities to collusion groups.

1 Introduction

The goal of Distributed Data Mining (DDM) is to find patterns or models from a collection of distributed datasets, that is, datasets residing on the nodes of a communication network, possibly under constraints of limited bandwidth and data privacy [16]. In DDM, Distributed Data Clustering (DDC) is the problem of finding groups of similar objects in distributed datasets [4].
Privacy and data ownership play an important role in DDM and in DDC in particular, a role which calls for a privacy preserving solution [13, 6]. Two main approaches have emerged to address this problem: *secure multiparty computation* and *model-based data mining*. With the secure multi-party computation (SMC) approach, all computations are performed by a group of mining parties following a given protocol and using cryptographic techniques to ensure that only the final results will be revealed to the participant, e.g. secure sum, secure comparison [5], secure set union [3]. In the model-based approach, each site computes a partial local model from the local dataset and all local models are aggregated to produce a global model, which is shared with the participants.
Each approach proposes a different privacy measure. These privacy measures, however, either assume that no insider is going to try to infer some sensitive

information, or do not account for particularities of specific data mining tasks. The privacy definition in SMC considers only threats from the outside and does not care about how much an inside party can learn from the protocol output. For example, in a protocol where three parties compute the sum of numbers in a secure multiparty protocol, e.g. secure sum, the process does not leak any input information. However, when two parties collude they can subtract its own input and learn the input of remaining party.

Model-based approaches, on the other hand, define privacy from the perspective of the whole dataset, and not of single points. For example, in [11] dataset privacy is based on the average privacy of all points. Some points will, of course, have privacy level much lower than the average privacy level. Thus, even if a single point has a very low privacy level, this privacy breach may go unnoticed.

In this paper, we present distributed privacy measures that take into account inference attacks from insiders and are able to detect point-level privacy breaches. We follow an information theoretic approach and define a set of properties from which our privacy measures are derived. We also apply the proposed measures to representative algorithms to demonstrate previous undetected privacy issues. As main contribution, this paper: (i) introduces the first privacy measures for DDM in general and DDC in particular with respect to insider and collusion attacks, and single data point privacy; (ii) re-evaluates the privacy preservation of representative DDC algorithms with these measures, and reveals that insider collusion is an important source of privacy breach and (iii) exemplifies the respective privacy analysis for selected algorithms.

2 Privacy Measures for Distributed Data Clustering

Major threats in a distributed mining session come from malicious insiders. Therefore, privacy measures should take into account the presence of collusion groups of malicious peers. Moreover, privacy measures should detect the privacy level of single data points.

In this paper, we will regard privacy measures as functions which, for a given distributed data mining algorithm, map a dataset subset and the maximum size of collusion groups of parties to a real number, and satisfy certain properties. We will call the value of such a measure a *privacy level*.

Let $L_1, \ldots, L_p$ be sites hosting one element of a partition of a dataset D each, and $\mathcal{A}$ be any distributed data mining algorithm running on $L_1, \ldots, L_p$. We will assume that up to $p-1$ sites among $L_1, \ldots, L_p$ are malicious, in that they seek to infer objects of D, or parts thereof, possibly in collusion groups of at most $c < p$ members, by either exchanging information or violating the protocol of $\mathcal{A}$, or both. By *privacy measure* for $\mathcal{A}$ we mean a computable partial function

$$\mathbf{PR}_{\mathcal{A}} \colon (X, c) \in 2^D \times \{0, 1, \ldots, p-1\} \to \mathbf{PR}_{\mathcal{A}[c]}(X) \in [0, \infty) \tag{1}$$

which satisfies one or multiple of the following properties:

P1 (collusion) $\mathbf{PR}_{\mathcal{A}[1]}(X) \geq \mathbf{PR}_{\mathcal{A}[c]}(X)$ when there are at most c malicious peers colluding, for all $c \in \{1, \ldots, p-1\}$ and for all $X \subseteq D$;

P2 (point monotonicity) it is non-increasing from singletons to dataset, i.e., $\mathbf{PR}_{\mathcal{A}[c]}(\{x\}) \geq \mathbf{PR}_{\mathcal{A}[c]}(D)$ for all $c \in \{0, 1, \ldots, p-1\}$.

Property **P1** expresses the decrease of privacy level in scenarios with inference attacks and collusion from malicious parties. Note that $c = 1$ expresses the absence of collusion groups of size two or greater. Therefore, parties are semi-honest, i.e., they adhere to the protocol of $\mathcal{A}$, but may exploit information gathered during execution for inference purposes. When no parties attempt to infer objects, i.e., are honest, there are no inside threats and $c = 0$. Property **P2** constrains $\mathbf{PR}_{\mathcal{A}}$ to behave as a worst-case measure: a greater privacy level than the one at singletons is not attainable for the dataset. We call this property *point-level* awareness.

Throughout this paper, we use the following notation. To indicate explicitly a privacy measure m in the evaluation of a given algorithm $\mathcal{A}$ we use the notation $\mathbf{PR}^m_{\mathcal{A}}$. We indicate the privacy of a given point x, given an algorithm $\mathcal{A}$ and measure m, as $\mathbf{PR}^m_{\mathcal{A}}(x)$; for a dataset D we use $\mathbf{PR}^m_{\mathcal{A}}(D)$. For the sake of simplicity, we omit algorithm, measure, dataset or data point, when they are implicit in a given context.

2.1 Existing Privacy Measures for DDC

In SMC, privacy is equivalent to having a trusted third party perform the computation and erasing all of the input data after the computation [7]. An SMC protocol is said to preserve privacy if one can prove that after the computation no party learns anything but the final results, as it would be the case if there were a trusted third party in the setting. This notion of privacy is known as the *simulation paradigm* [5], and is used to formally define privacy for SMC protocols. For a discussion on proofs for SMC protocols, the reader may refer to [9] and [5]. The privacy measure which is used in SMC protocols will be called *private computation*, and denoted in this paper as $\mathbf{PR}^{PC}_{\mathcal{A}}(D)$.

Model-based approaches work by producing partial local models, which are then aggregated into a global model [8, 10, 11, 14]. Finally, using the global model each party computes the mining results. Examples of models include wavelets coefficients, parametric models, like a mixture of Gaussians, or non-parametric models, like kernel density estimates.

In the model-based approach the *likelihood-based* measure is used in the context of clustering and classification [10]. Let D be a given dataset and $f_\lambda(x)$ be the probability density function associated with a given probabilistic model λ. The privacy $\mathbf{PR}^{like}_{\mathcal{A}}$ of data set D given model λ is defined as the reciprocal of the geometrical mean likelihood of the dataset being generated under model λ and can be expressed as:

$$\mathbf{PR}^{like}_{\mathcal{A}}(D) = 2^{\left(-\frac{1}{|D|}\sum_{x \in D} \log f_\lambda(x)\right)} \tag{2}$$

This measure indicates how likely a sampled data set is to occur given a probability model [10]. If the likelihood is high, then the privacy is low and vice-versa.

2.2 Limitations of Current Measures

In the presence of collusion groups a secure multiparty protocol (SMC) may fail [9] because its privacy definition gives only the privacy level from the outsiders' point of view. Any malicious insiders will receive the correct output, from which they may try to reconstruct sensitive inputs from other parties.

$\mathbf{PR}_{\mathcal{A}}^{PC}(D)$ does not address inference or collusion (¬P1). The private computation measure was designed to detect leaks from the protocol and not from outputs. Consider a protocol where three parties compute the set union in a secure multiparty protocol. The process does not leak any input information, but when two parties collude they can remove its own input sets and learn the input set of remaining party. However, privacy computation does not address this inference attack situation and $\mathbf{PR}_{SMCSum[0]}^{PC}(x) = \mathbf{PR}_{SMCSum[2]}^{PC}(x)$ when they should indicate the decrease in privacy in the presence of 2 malicious parties working in collusion.

$\mathbf{PR}_{\mathcal{A}}^{PC}(D)$ is point-level (P2). By definition, if any point $x \in D$, the dataset of inputs of a given party, is leaked, the protocol is considered not private, i.e. $\forall x \in D : \mathbf{PR}_{\mathcal{A}}^{PC}(x) = 0 \rightarrow \mathbf{PR}_{\mathcal{A}}^{PC}(D) = 0$. Therefore, $\forall x \in D : \mathbf{PR}_{\mathcal{A}}^{PC}(x) \geq \mathbf{PR}_{\mathcal{A}}^{PC}(D)$.

The likelihood-based measure is discussed in the following. Let $D = \{1, 4, 6, 9\}$, a dataset, and a mixture of two Gaussian with the first model be centered at $x_0 = 1$, i.e. it has mean $\mu_1 = 1$ with variance $\sigma_1^2 = 0.1$. The second model models the three remaining points, i.e., it has mean $\mu_2 = 6.3$ and variance $\sigma_2^2 = 1.0$. With probability density function of model denoted by $f(x)$, using Eq. (2) we have: $\mathbf{PR}_{\mathcal{A}}^{like}(D) = 2^{\left(-\frac{1}{|4|}(\log_2(f(1)) + log_2(f(4)) + log_2(f(6)) + log_2(f(9)))\right)} = 13.7326$.

$\mathbf{PR}_{\mathcal{A}}^{like}(D)$ does not address inference or collusion (¬P1). If the mixture of local models represents datasets from participants and a malicious insider has access to all local models, it can try to reconstruct other participants' datasets. In the above example, the attacker could reconstruct the first point with high precision with the first model, which is centered at x_0 with small variance. $\mathbf{PR}_{\mathcal{A}[1]}^{like}(x_0) = 2^{-\log_2 f(x_0)} = 0.5013$. However, even in this case $\mathbf{PR}_{\mathcal{A}[1]}^{like}(D) = 13.7326$, i.e. the drop in privacy due to a insider attack ($c \geq 1$) is not reflected in $\mathbf{PR}^{like}$. Therefore, $\mathbf{PR}_{\mathcal{A}}^{like}(D)$ does not fulfill property P1.

$\mathbf{PR}_{\mathcal{A}}^{like}(D)$ is not point-level (¬P2). When only a few points have a high likelihood of being reconstructed with a high precision, $\mathbf{PR}_{\mathcal{A}}^{like}(D)$ measure will still indicate a high privacy protection. Consider a dataset D above. The geometrical mean in the privacy measure smoothed out the measure for $x_0 = 1$, masking a possible privacy breach. Thus,

$$\mathbf{PR}_{\mathcal{A}}^{like}(x_0) = 2^{-\log_2 f(x_0)} = 0.5013 < \mathbf{PR}_{\mathcal{A}}^{like}(D) = 13.7326$$

Therefore, $\mathbf{PR}_{\mathcal{A}}^{like}(D)$ does not fulfill property P2. Table 1 presents a summary of all studied privacy measures and their properties.

	Reference	Approach	Collusion (P1)	Point-level (P2)
$\mathbf{PR}^{PC}$	[5]	Simulation	no	yes
$\mathbf{PR}^{like}$	[10]	Probability	no	no
$\mathbf{PR}^{range}$	(Def. 1)	Info. theory	yes	yes
$\mathbf{PR}^{rec}$	(Def. 2)	Inference analysis	yes	yes
$\mathbf{PR}^{BK}$	(Def. 3)	Info. theory	yes	yes

Table 1. Summary of privacy measures and properties

3 New Privacy Measures for DDC

In this section, we propose new privacy measures to analyze distributed data clustering algorithms. We assume that the attackers are members of the mining group and that they have access to the resulting cluster map and other information defined by the mining protocol being analyzed.
Our first measure defines the privacy of a cluster as the size of the interval between its maximal and minimal values.

Definition 1 (Cluster range measure). *Given a dataset D and a cluster map $\mathcal{C} = \{C_k\} \subseteq 2^D$, whose elements C_k are pairwise disjoint. We define the cluster privacy of a given point x in a given cluster $C_k \in \mathcal{C}$ as:*

$$\mathbf{PR}^{range}(x) = max\{C_k\} - min\{C_k\}. \tag{3}$$

Extending to the whole dataset:

$$\mathbf{PR}^{range}(D) = \min\{\mathbf{PR}^{range}(x) : x \in D\}. \tag{4}$$

As an example, consider a cluster of data points over the dimension "annual income" ranging from US$ 100 000 to US$ 150 000 reveals the value of each data point with a maximal error of US$ 50 000 and maximal mean error of US$ 25 000 (assuming uniform distribution). Consequently, each point in this cluster is said to have a privacy level of 50 000 dimension units, US$ in this case.
If a reconstruction method is known, it is possible to measure how close the reconstructed data gets to the original sensitive data.

Definition 2 (Reconstruction based measure). *Let $R \subset \mathbb{R}$ denote a set of reconstructed data objects such that each $r_i \in R$ is a reconstructed version of $x_i \in D$. We define the privacy level, given a reconstruction method, by:*

$$\mathbf{PR}^{rec}(x_i) = \mid x_i - r_i \mid . \tag{5}$$

Extending to the whole dataset:

$$\mathbf{PR}^{rec}(D) = \min\{\mathbf{PR}^{rec}(x_i) : x_i \in D, r_i \in R, 1 \leq i \leq N\} \tag{6}$$

where N is the size of the dataset D.

Consider secure k-means algorithm [15]. In this algorithm parties L_1 and L_p hold together the information on the distance $d = |x - \mu_i|$ between a given centroid μ_i and other parties data points x. Thus, attackers can use the inverse of the distance as a reconstruction method to infer data points x. $\mathbf{PR}^{rec}(x)$ will denote the precision of this specific reconstruction method.

A general definition of privacy proposed in the centralized data mining setting is the *bounded knowledge* measure [1], which defines privacy as the length of the interval from which a random variable X is generated. This measure can be expressed in terms of the entropy of X, as follows.

Definition 3 (Bounded Knowledge). *Given a random variable X with probability density function f_X and domain Ω_X, the privacy of X given by its bounded knowledge is:*

$$\mathbf{PR}^{BK}(X) = 2^{h(X)} \tag{7}$$

where $h(X) = -\int_{\Omega_X} f_X(x) log_2 [f_X(x)]\, dx$ is the differential entropy.

As an example, consider a random variable X uniformly distributed between 20 and 70, abbreviated $X \sim U(20, 70)$, has probability density function $f(x) = \frac{1}{50}$, for $20 \leq x \leq 70$, and 0 otherwise. The entropy of X is $h(X) = log_2(50)$. Thus, the privacy provided by X according to bounded knowledge measure is $\mathbf{PR}^{BK}(X) = 2^{log_2(50)} = 50$. This definition is general enough to be used in different data mining contexts, e.g. cluster analysis, association rules, etc. [2].

For a given point $x \in C_i$, a cluster in cluster map $\mathcal{C}$ induced from D, X_i a random variable for values of C_i and a probability density function $f_{X_i}(x)$, let:

$$\mathbf{PR}^{BK}(x) = \mathbf{PR}^{BK}(X_i) = 2^{h(X_i)} \tag{8}$$

with $f_{X_i}(x)$ being zero outside C_i.

In the case of a cluster map, we are interested in the smallest interval size in the said map[4]. Therefore,

$$\mathbf{PR}^{BK}(D) = \min\{\mathbf{PR}^{BK}(x)\} = \min\{2^{h(X_i)}\}. \tag{9}$$

The next definition extends each of the previously defined measures to include collusion groups.

Definition 4. *Let $\mathcal{A}$ be a distributed data clustering algorithm, D be a dataset, and measure $m \in \{rec, range, BK\}$, with collusion groups containing at most c attackers. We define:*

$$\mathbf{PR}^{m}_{\mathcal{A}[c]}(D) = \min\{\mathbf{PR}^{m}_{\mathcal{A}[i]}(D) : 1 < i \leq c\}. \tag{10}$$

$\mathbf{PR}^{m}_{\mathcal{A}[c]}(D)$ represents the minimum privacy level provided to dataset D when the collusion groups have at most c peers, using any privacy measure m. For example, $\mathbf{PR}^{BK}_{\mathcal{A}[2]}(D)$ denotes the privacy level provided by algorithm $\mathcal{A}$ to dataset D when collusion groups are formed with at most 2 malicious peers analyzed with BK measure.

[4] This notion comes from the well-known idea in computer security that defines the security level of a system as the level of its *weakest link*.

Properties Analysis of $\mathbf{PR}^{m}_{\mathcal{A}[c]}(D)$

Lemma 1 (Collusion). *Given an algorithm $\mathcal{A}$, for all dataset D and privacy measures $m \in \{range, BK, rec\}$, and $c > 1$ (presence of non-singleton collusion groups), if there is a collusion scenario decreasing the privacy level of dataset D, then $\mathbf{PR}^{m}_{\mathcal{A}[1]}(D) \geq \mathbf{PR}^{m}_{\mathcal{A}[c]}(D)$.*

Proof. Let $a = \mathbf{PR}^{m}_{\mathcal{A}[1]}(D)$ be the privacy level of dataset D with algorithm $\mathcal{A}$ with no collusion (i.e., $c = 1$), and $b = \mathbf{PR}^{m}_{\mathcal{A}[c]}(D)$ be the privacy level in a collusion scenario with $c > 1$ malicious peers. By definition $\mathbf{PR}^{m}_{\mathcal{A}[c]}(D)$ is the smallest privacy level considering all collusion scenarios. Thus, $\mathbf{PR}^{m}_{\mathcal{A}[c]}(D) = \min\{a, b\}$. Therefore, if the collusion group decreases the privacy level of the $c = 1$ scenario, then $a \geq \min\{a, b\}$. □

Lemma 2 (Point level privacy). $\forall x \in D : \mathbf{PR}^{m}_{\mathcal{A}[c]}(x) \geq \mathbf{PR}^{m}_{\mathcal{A}[c]}(D)$*, for all dataset D and privacy measures $m \in \{range, BK, rec\}$.*

Proof. (Range) Consider a cluster map $\mathcal{C}$ from D, with only two clusters C_a and C_b. Let r_a and r_b denote $r_a = max\{C_a\} - min\{C_a\}$ and $r_b = max\{C_b\} - min\{C_b\}$, the cluster range of C_a and C_b respectively. For a given point $x_a \in C_a$, by definition, $\mathbf{PR}^{range}(x_a) = r_a$ and $\mathbf{PR}^{range}(D) = \min\{r_a, r_b\}$. Therefore, $r_a \geq \min\{r_a, r_b\}$
(Rec) Consider a dataset D and a reconstructed set R. Let x_a be any given point in D and r_a its reconstructed counterpart in R. By definition, $\mathbf{PR}^{rec}_{\mathcal{A}[c]}(x_a)$ is $|x_a - r_a|$ and $\mathbf{PR}^{rec}_{\mathcal{A}[c]}(D) = \min\{|x_i - r_i| : x_i \in D, r_i \in R, 1 \leq i \leq N\}$. Therefore, $|x_a - r_a| \geq \min\{|x_i - r_i| : x_i \in D, r_i \in R\}$.
(BK) Consider a cluster map $\mathcal{C}$ from D, with only two clusters C_a and C_b. Let X_a be a random variable modeling a data point $x_a \in C_a$, and X_b a random variable modeling data points $x_b \in C_b$. By definition, $\mathbf{PR}^{BK}_{\mathcal{A}[c]}(x_a)$ is $2^{h(X_a)}$ and $\mathbf{PR}^{BK}_{\mathcal{A}[c]}(D) = \min\{2^{h(X_a)}, 2^{h(X_b)}\}$.
Therefore, $\mathbf{PR}^{BK}_{\mathcal{A}[c]}(x_a) = 2^{h(X_a)} \geq \min\{2^{h(X_a)}, 2^{h(X_b)}\}$. Similarly, we have that $\mathbf{PR}^{BK}_{\mathcal{A}[c]}(x_b) = 2^{h(X_b)} \geq \min\{2^{h(X_a)}, 2^{h(X_b)}\}$. □

We have thus derived three privacy measures, Cluster Range, Reconstruction, and Bounded Knowledge, that are inspired by different abstractions of privacy, and satisfy the natural properties of collusion and point-level awareness. In contrast, the Private Computation and Likelihood privacy measures fail to capture at least one of such properties. We will now revisit prominent DDC algorithms to examine if and how applying the new measures changes their evaluation, as to the amount of privacy that is guaranteed by each of them.

4 Application to DDC Algorithms

To apply our measures to DDC algorithms, we need to analyze which information is available to each party during the mining session, which collusion groups

can be formed and how they can reconstruct information from available information (including single malicious attacks). In the following, a few algorithms for distributed data clustering are briefly reviewed and their privacy properties are then analyzed in light of our privacy definitions. We selected these algorithms because they are based on prominent methods for distributed data clustering.

4.1 Secure Multiparty k-Means

Vaidya and Clifton [15] proposed an extension of the classic k-means algorithm to the distributed setting, using cryptographic protocols to achieve privacy (VC-kmeans). Data is assumed to be vertically partitioned. The solution is based on a secure sum protocol to find the closest cluster for any given point. It also uses secure permutation and secure comparison. VC-kmeans assumes three trusted parties L_1, L_2 and L_p. Additionally, let L_j be any other non-trusted party in the mining group. It was originally evaluated with $\mathbf{PR}^{PC}$ as private with three trusted parties, but no analysis is presented on how much privacy is preserved under collusion.

Single Insider Attacks. A given party L_j knows only: (i) $\boldsymbol{\mu}_j$, a share of the centroid; (ii) d_{ij}, the distance from the cluster centroid $\boldsymbol{\mu_i}$ to the view of point x_j; (iii) and a random vector $\boldsymbol{v}_j$. L_1 is the party which starts the protocol and knows: (i) a partial view of the cluster centroids, $\boldsymbol{\mu}_1$; (ii) the cluster assignment for each data point x; (iii) a random vector $\boldsymbol{v}$; and (iv) a permutation π of 1 to k, used to preserve the privacy of information in the SMC protocol. L_2 knows $\boldsymbol{T}_2 = \pi(\boldsymbol{v}_2 + \boldsymbol{d}_2)$, the permuted sum of $\boldsymbol{v}_2$ with $\boldsymbol{d}_2$, which is hidden from the other parties but L_p. L_p knows its share of the centroid $\boldsymbol{\mu}_p$, and $\boldsymbol{T}_i = \pi(\boldsymbol{v}_i + \boldsymbol{d}_i)$, $i = 1, 3, 4, \ldots, p$, the permuted sum of $\boldsymbol{v}_i$ with $\boldsymbol{d}_i$ of each party but L_2. Moreover, L_p knows the combined sum of $\boldsymbol{T}_i$ from all parties but L_2, i.e. $\boldsymbol{Y} = \boldsymbol{T}_1 + \sum_{i=3}^{p} \boldsymbol{T}_i$. L_1 is the party holding the most important information, which can be used to reconstruct sensitive data, including the random vector $\boldsymbol{v}$ and the permutation π. However, without the permuted sum of distances $\boldsymbol{Y}_i$ from other parties ($i = 1, 3, 4, \ldots, p$) L_1 will not learn anything, because it cannot reconstruct data points from other parties. Similarly, L_2 and L_p will not learn anything from the information they hold alone.

Let D be a n-dimensional dataset distributed over a network of peers. When there are only single insider attacks, algorithm VC-kmeans produces a cluster map of C from D with a privacy level given by:

$$\mathbf{PR}^{range}_{VCkmeans[1]}(D) = \min\{\max(C_i) - \min(C_i)\} \tag{11}$$

with $\forall C_i \in C$. Any insider attacker working solo can only learn what is disclosed by the cluster map itself – namely, that each point ranges in the interval $min,\ max$, for a given cluster. Contrast this information with the result of an SMC analysis, which only tells us that the protocol is private, but does not quantify it in terms of original data space units.

Attack with Collusion of Insiders L_1 and L_p. Together, L_1 and L_p hold information on the permuted sum of all parties except for L_2. Moreover, they hold

information on the permutation π and the random vector $\boldsymbol{v}$. Therefore, this collusion group may compute the vector $\boldsymbol{d}_i$ using inverse of permutation π:

$$\boldsymbol{d}_i = \pi^{-1}(\boldsymbol{Y}_i) - \boldsymbol{v}_i \tag{12}$$

with $i = 1, 3, 4, \ldots, p$. The vector $\boldsymbol{d}_i$ represents the distance between a given point x and the cluster centroid i with mean μ_i, therefore, with the true distance, every point x can be located with an arbitrary error. Using Eq. (12) as reconstruction method, we apply $\mathbf{PR}^{rec}(D)$.

$$\mathbf{PR}^{rec}_{VCkmeans[2]}(D) = \min\{|x - r| : x \in D, r \in R\} \approx 0 \tag{13}$$

where D is the original dataset and R is a reconstructed dataset. Original evaluation with $\mathbf{PR}^{PC}_{VCkmeans}(D)$ does not inform how much privacy is lost with only one attacker. However, we find in our analysis that a malicious alone can learn no more than the size of each cluster.

4.2 Distributed Data Clustering with Generative Models

Merugu and Ghosh [10] present an algorithm for distributed clustering and classification based on generative models approach (DDCGM). Their algorithm outputs an approximate model $\hat{\lambda}_c$ of a true global model λ_c from a predefined fixed family of models F, *e.g.* multivariate 10-component Gaussian mixtures. DDCGM first computes local models λ_i, from which the average global model $\bar{\lambda}$ is generated by $p_{\bar{\lambda}}(x) = \sum_{i=1}^{n} \nu_i p_{\lambda_i}(x)$ where $p_\lambda(x)$ is the probability density function of a given model λ. The algorithm uses $\bar{\lambda}$ to find a good approximation $\hat{\lambda}_c$ of the true (and unknown) global model λ_c. The model $\hat{\lambda}_c$ is used as cluster map. Original privacy evaluation was based on $\mathbf{PR}^{like}(D)$, with all models in a mixture, regardless of the possible weakness of any component. The new evaluation reflects the weakest model in the mixture.

Single Insider Attacks. In the DDCGM scheme, a central entity receives local generative models and combines them into an average generative model. This entity knows individual generative model from each party. Arbitrary parties know only the global model. Since the models represent clusters, we can apply $\mathbf{PR}^{range}(D)$. Let $p_\lambda(x)$ be a mixture model with k elements. The privacy level provided by DDCGM using $p_\lambda(x)$ and with no collusion is:

$$\mathbf{PR}^{range}_{DDCGM[1]}(D) = \min\{x_{max} - x_{min}\} \tag{14}$$

where x_{max} and x_{min} are inferior and superior elements at the each cluster, according to the model $p_\lambda(x)$.

Assuming that each component model λ_i in the mixture is a Gaussian in a n dimensional data space with covariance matrix Σ_i, the entropy is $h_i(x) = \ln\sqrt{(2\pi e)^n |\Sigma_i|}$, where $|\Sigma_i|$ is the determinant of the covariance matrix of the given model, and consequently, a cluster. Therefore, we can compute:

$$\mathbf{PR}^{BK}_{DDCGM[1]}(D) = \min\left\{2^{h_i(x)}\right\} = \min\left\{2^{\ln\sqrt{(2\pi e)^n |\Sigma_i|}}\right\} \tag{15}$$

	Original assessment	Single attacks	Collusion attacks
VC-kmeans	[15] private (3 trusted)	$\min\{x_{max} - x_{min}\}$	decrease to ≈ 0, $c \geq 2$
DDCGM	[10] $2^{\left(-\frac{1}{\lvert D\rvert}\sum_{x\in D}\log f_\lambda(x)\right)}$	$\min\{2^{\ln\sqrt{(2\pi e)^n\lvert\Sigma_i\rvert}}\}$	same level, $c \geq 1$
ITDDC	[14] N/A	$\min\{x_{max} - x_{min}\}$	same level, $c \geq 1$
EC-kmeans	[12] private (0 trusted)	$\min\{x_{max} - x_{min}\}$	$\min\{x_{max} - x_{min}\}$, $c \geq 2$

Table 2. Summary of privacy preserving distributed data clustering algorithms.

Collusion Attack. Any collusion group must include the central party since there is little information for arbitrary parties, and collusion attacks reduce to single aggregator attack. Thus, $\mathbf{PR}^{BK}_{DDCGM[1]}(D) = \mathbf{PR}^{BK}_{DDCGM[c]}(D)$ with $c \geq 2$.

4.3 Information Theoretical Approach to Distributed Clustering

Shen and Li [14] proposed an information theoretical approach to distributed clustering (ITDDC). They assume a peer-to-peer network where each node solves a local clustering problem and updates its neighbors. The clustering problem is to fit a discriminative model to cluster boundaries that maximize the mutual information between cluster labels and data points. With low communication, local clusters are formed based on global information spread through the network. The algorithm needs several rounds of iterations to converge. When it comes to privacy, the authors do not investigate how the algorithm would behave under inference attacks and do not investigate how much privacy this approach does provide.

Single Insider Attack. Each party in ITDDC knows a set of discriminative models defining the clusters boundaries of points on data sets and from all its direct neighbors. We can apply $\mathbf{PR}^{range}(D_j)$ to compute how much privacy is preserved at local dataset D_j for a given model. Each party estimates $\hat{p}_j(k|x)$, a class label distribution defined by a local discriminative model (for instance, logistic regression). The distribution of x in a given cluster is not disclosed. Thus, each point can only be located in the interval corresponding to its cluster boundaries. The privacy provided by DDCGM using $\hat{p}_j(k|x)$ and with no collusion is:

$$\mathbf{PR}^{range}_{ITDDC[1]}(D) = \min\{x_{max} - x_{min}\} \tag{16}$$

where x_{max} and x_{min} are inferior and superior elements at the each cluster, according to the boundaries defined by model $\hat{p}_j(k|x)$.

Collusion attack. The only information being exchanged among the parties is the local models. Moreover, there is no special central entity holding extra information on data distribution at local datasets. Therefore, even if malicious parties collude against another party, they cannot improve on the single insider attack. Therefore, $\mathbf{PR}^{range}_{ITDDC[c]}(D) = \mathbf{PR}^{range}_{ITDDC[1]}(D)$, with $c \geq 1$ colluding parties.

4.4 Elliptic Curves for Multiparty k-means

Patel and colleagues [12] present a privacy-preserving distributed k-means algorithm based on elliptic curves (EC-kmeans). They assume no trusted party and use elliptic curves to achieve low overhead cryptography. No analysis on inference attack or collusion is presented by the authors.

Single Insider Attack. Each peer knows its own centroids, its own cluster boundaries and the encrypted version of the global centroids and the number of points in a global cluster. Without collusion, a given malicious party does not even know the boundaries of clusters residing on other parties.

Collusion. The initiator knows the information necessary to decrypt data in the mining session. Therefore, a collusion group with the initiator and any party L_i can learn about the centroids and number of points in each cluster on the party L_{i-1}. With the centroids, cluster boundaries of dataset D_j at L_j could be estimated and $\mathbf{PR}^{range}_{ECkmeans[2]}(D_j) = \min\{x_{max} - x_{min}\}$.

4.5 Discussion

Table 2 presents an overview of the studied algorithms. The analysis above shows that collusion is indeed a chief source of privacy breach, and that algorithms can be separated according to their vulnerability to collusion groups and to the malicious behavior of a site with a special role in the protocol, e.g., a central site, or an aggregator, or a protocol initiator. VC-kmeans is almost completely not private if the central site colludes, whereas DDCGM has limited vulnerability to the central site and not to collusion; ITDDC does not use a central site and only disclose cluster ranges, irrespective of collusions. EC-kmeans, finally, is secure and only discloses range information under a collusion attack that involves the initiator.

5 Conclusions

We presented new privacy measures for distributed data clustering, in order to overcome the limitations of existing measures. Starting from a set of formal properties, it was shown that the new measures satisfy the properties and, therefore, improve over previous ones. The new measures were applied to selected representative of privacy-preserving distributed data clustering algorithms. Some identified benefits from the new measures are the ability to detect the vulnerabilities of the representative algorithm to collusion in different scenarios and detect point level privacy breach. In fact, it was shown that collusion is indeed an important source of privacy breach, and that algorithms can be separated according to their vulnerability to collusion groups and to the potential malicious behavior of a site with a special role in the protocol, e.g., a central site, or an aggregator, or a protocol initiator.

Acknowledgment

This work was partly supported by the EU-funded project TOREADOR (contract n. H2020-688797)

References

1. D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proc. of the 20th Symp. on Principles of Database Systems (PODS)*, pages 247–255. ACM, May 2001.
2. E. Bertino, I. Fovino, and L. Provenza. A framework for evaluating privacy preserving data mining algorithms. *Data Mining and Knowl. Discovery*, 11(2), 2005.
3. C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. Tools for privacy preserving data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):28–34, 2002.
4. G. Forman and B. Zhang. Distributed data clustering can be efficient and exact. *SIGKDD Explor. Newsl.*, 2(2):34–38, Dec. 2000.
5. O. Goldreich. *Foundations of Cryptography: Volume 2 – Basic Applications*. Cambridge University Press, 2004.
6. C. Jones, J. Hall, and J. Hale. Secure distributed database mining: Principle of design. In *Advances in Distributed and Parallel Knowledge Discovery*, chapter 10, pages 277–294. AAAI Press / MIT Press, 2000.
7. M. Kantarcioglu. A survey of privacy-preserving methods across horizontally partitioned data. In *Privacy-Preserving Data Mining*, volume 34 of *The Kluwer Intl. Series on Advances in Database Systems*, pages 313–335. Springer, 2008.
8. M. Klusch, S. Lodi, and G. Moro. Agent-based distributed data mining: the KDEC scheme. In *Intelligent Information Agents: the AgentLink perspective*, volume 2586 of *LNCS*. Springer, 2003.
9. Y. Lindell and B. Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1(1), 2009. Article 5.
10. S. Merugu and J. Ghosh. Privacy-preserving distributed clustering using generative models. In *Proc. of the 3rd Intl. Conf. on Data Mining (ICDM)*. IEEE, 2003.
11. S. Merugu and J. Ghosh. A privacy-sensitive approach to distributed clustering. *Pattern Recognition Letters*, 26:399–410, 2005.
12. S. J. Patel, D. Punjani, and D. C. Jinwala. An efficient approach for privacy preserving distributed clustering in semi-honest model using elliptic curve cryptography. *International Journal of Network Security*, 17(3):328–339, 2015.
13. F. Provost. Distributed data mining: scaling up and beyond. In *Advances in Distributed and Parallel Knowledge Discovery*, pages 3–27. AAAI Press, 2000.
14. P. Shen and C. Li. Distributed information theoretic clustering. *IEEE Transactions on Signal Processing*, 62(13):3442–3453, July 2014.
15. J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proc. of the 9th Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 206–215. ACM, 2003.
16. M. J. Zaki. Parallel and distributed data mining: An introduction. In M. J. Zaki and C.-T. Ho, editors, *Large-scale parallel data mining*, pages 1–23. Springer, 2000.