Secure Two-party Agglomerative Hierarchical Clustering Construction*

Mona Hamidi[1], Mina Sheikhalishahi[2] and Fabio Martinelli[2]
[1]*Dipartimento Ingegneria dell Informazione e Scienze Matematiche, Universita di Siena, Siena, Italy*
[2]*Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Pisa, Italy*

Keywords: Privacy, Hierarchical Clustering, Data Sharing, Secure Two-party Computation, Distributed Clustering.

Abstract: This paper presents a framework for secure two-party agglomerative hierarchical clustering construction over partitioned data. It is assumed that data is distributed between two parties *horizontally*, such that for mutual benefits both parties are willing to identify clusters on their data as a whole, but for privacy restrictions, they avoid to share their datasets. To this end, in this study, we propose general algorithms based on *secure scalar product* and *secure hamming distance computation* to securely compute the desired criteria in constructing clusters' scheme. The proposed approach covers all possible secure agglomerative hierarchical clustering construction when data is distributed between two parties, including both *numerical* and *categorical* data.

1 INTRODUCTION

Facing the new challenges brought by a continuous evolving Information Technologies (IT) market, large companies and small-to-medium enterprises found in *Information Sharing* a valid instrument to improve their key performance indexes. Sharing data with partners, authorities for data collection and even competitors, may help in inferring additional intelligence through collaborative information analysis (Martinelli et al., 2016) (Sheikhalishahi and Martinelli, 2017b). Such an intelligence could be exploited to improve revenues, e.g. through best practice sharing, market basket analysis (Oliveira and Zaïane, 2002), or prevent loss coming from brand-new potential cyber-threats (Faiella et al., 2017). Other applications include analysis of medical data, provided by several hospitals and health centers for statistical analysis on patient records, useful, for example, to shape the causes and symptoms related to a new pathology (Artoisenet et al., 2013).

Information sharing, however, independently from the final goal, leads to issues and drawbacks which must be addressed. These issues are mainly related to the information privacy. Shared information might be sensitive, potentially harming the privacy of physical people, such as employee records for business applications, or patient records for medical ones (Martinelli et al., 2016). Hence, the most desirable strategy is the one which enables data sharing in a secure environment, such that it preserves the individual privacy requirement while at the same time the data are still practically useful for analysis.

Clustering is a very well-known tool in unsupervised data analysis, which has been the focus of significant researches in different studies, spanning from information retrieval, text mining, data exploration, to medical diagnosis (Berkhin, 2006). Clustering refers to the process of partitioning a set of data points into groups, in a way that the elements in the same group are more similar to each other rather than to the ones in other groups.

The problem of data clustering becomes challenging when data is distributed between two (or more) parties and for privacy concerns the data holders refuse to publish their own dataset, but still they are interested to shape more accurate clusters, identified on richer sets of data. For a motivating example suppose that a *hospital* and a *health center* hold different datasets with the same set of attributes. Both centers are interested to shape clusters on the whole data, which brings the benefits of identifying the trends and patterns of diseases on a larger set of samples. How would it be possible to learn about clusters' shapes without disclosing patients' records ?

To this end, in this study, we address the problem of securely constructing hierarchical clustering algorithms between two parties. The participating parties are willing to model an agglomerative hierarchical clustering on the whole dataset without revealing

*This work has been supported by the H2020 EU funded project C3ISP [GA #700294].

Hamidi, M., Sheikhalishahi, M. and Martinelli, F.
Secure Two-party Agglomerative Hierarchical Clustering Construction.
DOI: 10.5220/0006659704320437
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 432-437
ISBN: 978-989-758-282-0

their own dataset. Both scenarios of data being described in *numerical* and *categorical* attributes is addressed in this study. For each scenario, we propose secure two-party computation protocols which can be exploited as a general tool to construct securely all possible agglomerative hierarchical clustering algorithms between two parties.

At the end, each data holder finds the structure of hierarchical clusters on the whole data, without knowing the records of the other party. In this study, as in general cases, it is assumed that clustering on joint datasets produces better result rather than clustering on individual dataset.

The contribution of this paper can be summarized as the following. A framework is proposed which serves as a tool for two parties to detect the cluster structures on the whole dataset, in terms of agglomerative hierarchical clustering, without revealing their data, in two different scenarios of data being described either numerically or categorically.

Two secure computation protocols, named *secure scalar product* and *secure hamming distance* protocols, are exploited to propose new algorithms such that each party is able to find the closest points (or clusters) for agglomeration.

The rest of the paper is structured as follows. Related work, on the two concepts of privacy preserving data clustering and secure hierarchical clustering construction, is presented in Section 2. Section 3 presents some preliminary notations exploited in this study, including *agglomerative hierarchical clustering*, *secure scalar product*, and *secure hamming distance computation*. Section 4 presents the system model in three subsections: horizontal distributed data framework, secure Euclidean distance computation, and finally agglomerative hierarchical clustering. Finally Section 5 briefly concludes proposing future research directions.

2 RELATED WORK

The problem of privacy preserving data clustering is generally addressed for the specific case of *k*-means clustering, either when data is distributed between two parties (Bunn and Ostrovsky, 2007) or more than two parties (Jha et al., 2005).

In (Su et al., 2007), *Document clustering* has been introduced, and a cryptography based framework has been proposed to do the privacy preserving document clustering among the two parties. It is assumed that each has her own private documents, and wants to collaboratively execute agglomerative document clustering without disclosing their private content. The main idea is to find which documents have many words in common, and place the documents with the most words in common into the same groups and then build the hierarchy bottom up by iteratively computing the similarity between all pairs of clusters and then merging the most similar pair. In the proposed approach, differently from our technique, the problem when data are described through numerical attributes, is not addressed. In (De and Tripathy, 2013) a secure hierarchical clustering approach over vertically partitioned data is provided which increases the accuracy of the clusters over the existing approaches. However, in our study, we address the problem of hierarchical clustering construction when data is distributed horizontally.

In (Sheikhalishahi and Martinelli, 2017a), the problem of secure divisive hierarchical clustering construction is addressed when data is distributed between two parties horizontally and vertically. However, the criteria for divisive hierarchical clustering, discussed in this study, is different from agglomerative hierarchical clustering algorithms.

In all the aforementioned studies, differently from our approach, or data is distributed between more than two parties, or the problem is addressed for divisive hierarchical clustering, or the problem is addressed for vertical partitioned data. To the best of our knowledge the problem of secure two-party data clustering is a topic which is required to be explored deeper.

3 PRELIMINARY NOTATIONS

In this section, we present some background knowledge which are exploited in our proposed framework.

3.1 Hierarchical Clustering

Clustering algorithm partitions a set of objects into smaller groups such that all objects within the same group (cluster) are more similar or close to each other rather than the objects in different clusters (Jagannathan and Wright, 2005). There have been a large number of algorithms developed to solve the various formulations of data clustering. *Hierarchical clustering*, as a specific form of data clustering, generates a hierarchical decomposition of the given set of data objects, which can be either *agglomerative* or *divisive* based on how the hierarchical decomposition is formed. The agglomerative approach successively merges the objects or clusters which are close to one another, until all of the clusters are merged into one or until a termination condition holds. The divisive approach starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until finally each object is in one cluster, or until a termination condition holds (De and Tripathy, 2013). Hierarchical methods, differently from many other algorithms, have the ability to both discover clusters of arbitrary shape and deal with different data types.

3.2 Secure Scalar Product

Scalar product is a useful technique in data mining such that many data mining algorithms can be reduced to computing the scalar product.
For secure two-party scalar product computation, assume that two parties, named *Alice* and *Bob*, each has a vector of cardinality n, e.g $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$, respectively. Then, both are interested in securely obtaining the scalar product of the two vectors, i.e. $\sum_{i=1}^{n} x_i \cdot y_i$, without revealing their own vectors. There are many different solutions with varying degrees of accuracy, communication cost and security. The solution which we have applied is the one proposed in (Vaidya and Clifton, 2002), in which the key is to use linear combinations of random numbers to make vector elements, and then do some computations to eliminate the effect of random numbers from the result. Algorithm 1 details the process.

Algorithm 1: *Sec.Scalar()*: Secure Scalar Product.

```
Data: Alice and Bob have vectors $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$, respectively.
Result: Alice and Bob obtain securely $S = X \cdot Y$
initialization;
Alice and Bob together decide on random $n \times \frac{n}{2}$ matrix C
for Alice do
    Alice generates a random vector $R$ of cardinality $\frac{n}{2}$, $R = (r_1, \dots, r_{\frac{n}{2}})$
    Alice generates the $n \times 1$ matrix $Z$, where $Z = C \times R$
    Alice generates $X_1 = X + Z$
    Alice sends $X_1$ to Bob
end
for Bob do
    Bob generates the scalar product $S_1 = \sum_{i=1}^{n} x_{1i} \cdot y_i$
    Bob also generates the $n \times 1$ matrix, where $Y_1 = C^T \times Y$
    Bob sends both $S_1$ and $Y_1$ to Alice
end
for Alice do
    Alice generates $S_2 = \sum_{i=1}^{n} Y_{1i} \cdot R_i$
    Alice generates the scalar product $S = S_1 - S_2$
    Alice reports the scalar product $S$ to Bob
end
```

3.3 Secure Hamming Distance Computation

In the case that *Alice*'s and *Bob*'s vectors are described through *categorical* attributes, the distance between vectors is computed with the use of secure hamming distance. The secure communication between two parties for obtaining hamming distance is on the base of *oblivious transfer*. A 1-out-2 *oblivious transfer*, denoted by OT_1^2, is a two party protocol where one party (the sender) inputs n-bit strings $X_1, X_2 \in \{0,1\}^n$, and the other party (the receiver) inputs a bit y. At the end of the protocol, the receiver obtains X_b but learns nothing about X_{1-b}, while the sender learns nothing about b (Bringer et al., 2014).

In (Bringer et al., 2013), the secure computation of the Hamming distance has been presented based on oblivious transfer. It is assumed that two parties , say *Alice* and *Bob*, hold bit strings of the same length n, $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_n)$, respectively. Both are interested in jointly computing the *Hamming Distance* between X and Y, i.e. $D_H(X,Y) = \sum_{i=1}^{n} (x_i \oplus y_i)$ without revealing X and Y. Algorithm 2 details the process.

4 SYSTEM MODEL

In this section we explain in detail 1) what we mean by horizontal distributed data, 2) how Euclidean distance can be computed securely between two vectors owned by two different parties, and 3) how the proposed algorithms can be exploited to construct any agglomerative hierarchical clustering securely.

4.1 Horizontal Distributed Data

Suppose that two data holders are interested in detecting the structure of clusters (through agglomerative hierarchical clustering) on their datasets as a whole. However, for privacy concerns, they are not willing to publish or share the main dataset. As mentioned before, it is assumed that clustering on both datasets as a whole (as in general cases) produces better results comparing to clustering on individual dataset. In this study it is assumed that data is distributed *horizontally* between two parties. This means that each data holder has information about all the features but for different collection of objects. More precisely, let $\mathcal{A} =$

Algorithm 2: *Sec.Hamming()*: Secure Hamming Distance Computation.

```
Data: Alice and Bob have n-bit strings
      $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$,
      respectively.
Result: Alice and Bob obtain securely the
        hamming distance between $X$ and $Y$.
initialization;
Alice generates $n$ random values
  $(r_1, \ldots, r_n) \in_R Z_{n+1}$ and computes $R = \sum_{i=1}^{n} r_i$
for each $i = 1, \ldots, n$, Alice and Bob engage in
  a $OT_1^2$ do
    Alice acts as the sender and Bob as the
      reciever
    Bob's selection bit is $y_i$
    Alice's input is $(r_i + x_i, r_i + \bar{x}_i)$ where $x$ is a
      bit value and $\bar{x}$ denotes $1 - x$
    The output obtained by Bob is
      consequently $t_i = r_i + (x_i \oplus y_i)$
end
Bob computes $T = \sum_{i=1}^{n} t_i$ and sends $T$ to Alice
Alice computes and outputs $T - R$
```

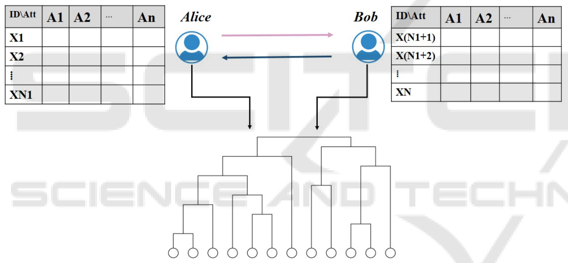

Figure 1: Hierarchical Clustering over Horizontal Partitioned Data.

$\{A_1, A_2, \ldots, A_n\}$ be the set of n attributes all used to express each record of data. Therefore, each record is an n dimensional vector $X_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_n})$, where $v_{i_j} \in A_j$.

Figure 1 depicts a higher level representation of hierarchical clustering construction on horizontal distributed framework. *Alice* and *Bob*, holding respectively datasets D_a and D_b, are the two parties interested in constructing hierarchical clustering on $D_a \cup D_b$, without knowing the data information of the other party. As it can be observed, the two tables are described with the same set of attributes, but on different objects. To discover the structure of the desired agglomerative clustering algorithm, *Alice* and *Bob* communicate through secure computation protocols. Depending on if data in both sides is described through *numerical* or *categorical* attributes, secure *Euclidean* or *Hamming* distance computation algorithms are exploited inside communication algorithms, respectively.

4.2 Secure Euclidean Distance Computation

Suppose *Alice* and *Bob* own vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$, respectively. We assume that $n > 2$, and $x_i, y_i \in \mathbb{R}$ for all i, i.e. both vectors contain numerical elements. Both participated parties are interested in obtaining securely the result of Euclidean distance between X and Y, i.e. $D_E(X,Y) = (\sum_{i=1}^{n} (x_i - y_i)^2)^{\frac{1}{2}}$. Algorithm 3 details the process of secure Euclidean distance computation with the use of secure scalar product (presented in Algorithm 4).

Algorithm 3: *Sec.Euclidean()*: Secure Euclidean Distance Computation.

```
Data: Alice and Bob have numerical vectors
      $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$,
      $n > 2$, respectively.
Result: Alice and Bob compute securely
        Euclidean Distance of $X$ and $Y$
$D(X,Y) = (\sum_{i=1}^{n} (x_i - y_i)^2)^{\frac{1}{2}} =$
$(\sum_{i=1}^{n} (x_i)^2 + (y_i)^2 - 2(x_i \cdot y_i))\frac{1}{2}$
initialization;
Alice reports $X^2 = \sum_{i=1}^{n} {x_i}^2$
Bob reports $Y^2 = \sum_{i=1}^{n} {y_i}^2$
$X \cdot Y \leftarrow$ Sec.Scalar(X,Y)
return $D_E(X,Y) = (X^2 + Y^2 - 2X \cdot Y)^{\frac{1}{2}}$
```

Theorem 1. *Euclidean distance computation as proposed in Algorithm 3 does not reveal the information of each party's data, but the distance.*

Proof. Due to the fact that it is assumed $n > 2$, hence X^2 and Y^2 will not reveal something about the specific amount of each component, but the squared result of their vectors. Moreover, $X \cdot Y$ is computed through secure scalar product protocol proven to be secure in (Clifton et al., 2002) (Vaidya and Clifton, 2002). □

4.3 Secure Agglomerative Hierarchical Clustering

In the beginning of the agglomerative hierarchical clustering process, each element is located in a cluster by its own. Afterwards, the set of N objects to be clustered are grouped into successively fewer than N sets, arriving eventually at a single set containing all N objects (Day and Edelsbrunner, 1984). According to different distance measures between

Table 1: Specification of Hierarchical Clustering Methods.

Hierarchical Clustering	Lance-Williams	Distance Metric
Single Link	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = -0.5$	$d(i \cup j,k) = \frac{1}{2}d(i,k)$ $+\frac{1}{2}d(j,k) - \frac{1}{2}\lvert d(i,k) - d(j,k)\rvert$
Complete Link	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0.5$	$d(i \cup j,k) = \frac{1}{2}d(i,k)$ $+\frac{1}{2}d(j,k) + \frac{1}{2}\lvert d(i,k) - d(j,k)\rvert$
Group average(UPGMA)	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$ $\beta = 0$ $\gamma = 0$	$d(i \cup j,k) = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}d(i,k) + \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}d(j,k)$
Weighted group average (WPGMA)	$\alpha_i = 0.5$ $\beta = 0$ $\gamma = 0$	$d(i \cup j,k) = \frac{1}{2}d(i,k) + \frac{1}{2}d(j,k)$
Median method	$\alpha_i = 0.5$ $\beta = -0.25$ $\gamma = 0$	$d(i \cup j,k) = \frac{1}{2}d(i,k)$ $+\frac{1}{2}d(j,k) - \frac{1}{4}d(i,j)$
Centroid method	$\alpha_i = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}$ $\beta = -\frac{\lvert i\rvert\lvert j\rvert}{(\lvert i\rvert+\lvert j\rvert)^2}$ $\gamma = 0$	$d(i \cup j,k) = \frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}d(i,k)$ $+\frac{\lvert i\rvert}{\lvert i\rvert+\lvert j\rvert}d(j,k) - \frac{\lvert i\rvert\lvert j\rvert}{(\lvert i\rvert+\lvert j\rvert)^2}d(i,j)$
Ward method	$\alpha_i = \frac{\lvert i\rvert+\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$ $\beta = -\frac{\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}$ $\gamma = 0$	$d(i \cup j,k) = \frac{\lvert i\rvert+\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}d(i,k)+$ $\frac{\lvert i\rvert+\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}d(j,k) - \frac{\lvert k\rvert}{\lvert i\rvert+\lvert j\rvert+\lvert k\rvert}d(i,j)$

clusters, all agglomerative hierarchical methods have been divided into seven methods with the use of a formula named *Lance-Williams* (Murtagh and Contreras, 2012). These seven categories of agglomerative hierarchical clustering algorithms are named *single link*, *complete link*, *average link*, *weighted average link*, *Ward's method*, *centroid method*, and *the median* (Gan et al., 2007).

More precisely, in agglomerative hierarchical clustering algorithms, the Lance-Williams formula is used to calculate the dissimilarity between a cluster and the other cluster formed by merging two other clusters (Gan et al., 2007). Formally, if objects i and j are agglomerated into cluster $i \cup j$, then the new dissimilarity between the cluster and all other objects of cluster k is required to be specified as the following:

$$\begin{aligned} d(i \cup j,k) =& \alpha_i\, d(i,k) + \alpha_j\, d(j,k) \\ &+ \beta\, d(i,j) + \gamma\, \lvert d(i,k) - d(j,k)\rvert \end{aligned} \quad (1)$$

where α_i, α_j, β and γ defines the agglomerative parameters (Murtagh and Contreras, 2012). The value for each of these coefficients in different algorithms has been listed in Table 1. Thence, if *Alice* and *Bob* are able to find these different distance metrics securely, they are both able to construct all possible agglomerative hierarchical clustering algorithms without revealing their own data.

As it can be observed from Table 1, it is enough that the two participating parties obtain securely the distance of each pair of elements. This means that it is required for them to construct securely the *dissimilarity matrix* on whole elements without disclosing the data.

Algorithm 4 presents how *Alice* and *Bob* are able to construct securely the dissimilarity matrix on all records with the use of secure Hamming and Euclidean distance computations, presented respectively in Algorithms 2 and 3.

Algorithm 4: *Sec.Matrix():* Secure Distance Matrix Computation.

```
Data: Alice and Bob have information of
      records $X_1, \ldots, X_k$ and $X_{k+1}, \ldots, X_N$,
      respectively.
Result: Secure distance matrix computation
initialization;
for $1 \le t, t' \le k$ do
    if $X_t, X_{t'}$ are numerical then
        Alice reports $(M(t,t') \leftarrow D_E(X_t, X_{t'}))$
    else
        Alice reports $(M(t,t') \leftarrow D_H(X_t, X_{t'}))$
    end
end
for $k+1 \le s, s' \le N$ do
    if $X_s, X_{s'}$ are numerical then
        Bob reports $(M(s,s') \leftarrow D_E(X_s, X_{s'}))$
    else
        Bob reports $(M(s,s') \leftarrow D_H(X_s, X_{s'}))$
    end
end
for $1 \le t \le k$ do
    for $k+1 \le s \le N$ do
        if $X_t, X_s$ are numerical then
            $M(t,s) \leftarrow Sec.Euclidean(X_t, X_s)$
        else
            $M(t,s) \leftarrow Sec.Hamming(X_t, X_s)$
        end
    end
end
for $1 \le t, s \le N$ do
    return $M(t,s)$
end
```

Theorem 2. *Algorithm 4 reveals no information to other party except the distances of all elements.*

Proof. The proof is straightforward from secure computation of Euclidean and Hamming distance proven to be secure in Theorem 1 and in (Bringer et al., 2013), respectively. □

5 CONCLUSION

In this work we proposed a framework which can be exploited for two parties to construct any agglomerative hierarchical clustering algorithm on their data as a whole, without revealing the original datasets. To this end, secure two-party computation algorithms are proposed to obtain the required criteria for detecting the clusters on the whole data. Two scenarios of data being described numerically and categorically have been addressed. In future direction we plan to analyze the efficiency of proposed approach on benchmark dataset clustering to evaluate communication cost in reality. Moreover we plan to address the problem when data is distributed among more than two parties either horizontally or vertically.

REFERENCES

Artoisenet, C., Roland, M., and Closon, M. (2013). Health networks: actors, professional relationships, and controversies. In *Collaborative Patient Centred eHealth*, volume 141. IOSPress.

Berkhin, P. (2006). A survey of clustering data mining techniques. In Kogan, J., Nicholas, C., and Teboulle, M., editors, *Grouping Multidimensional Data*, pages 25–71. Springer Berlin Heidelberg.

Bringer, J., Chabanne, H., Favre, M., Patey, A., Schneider, T., and Zohner, M. (2014). Gshade: Faster privacy-preserving distance computation and biometric identification. In *Proceedings of the 2Nd ACM Workshop on Information Hiding and Multimedia Security*, pages 187–198, New York, NY, USA.

Bringer, J., Chabanne, H., and Patey, A. (2013). SHADE: secure hamming distance computation from oblivious transfer. In *Financial Cryptography and Data Security - FC 2013 Workshops, USEC and WAHC 2013, Okinawa, Japan, April 1, 2013, Revised Selected Papers*, pages 164–176.

Bunn, P. and Ostrovsky, R. (2007). Secure two-party k-means clustering. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 486–497, NY, USA. ACM.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., and Zhu, M. Y. (2002). Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.*, 4(2):28–34.

Day, W. H. E. and Edelsbrunner, H. (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1(1):7–24.

De, I. and Tripathy, A. (2013). A secure two party hierarchical clustering approach for vertically partitioned data set with accuracy measure. In *Recent Advances in Intelligent Informatics - Proceedings of the Second International Symposium on Intelligent Informatics, ISI 2013, August 23-24 2013, Mysore, India*, pages 153–162.

Faiella, M., Marra, A. L., Martinelli, F., Mercaldo, F., Saracino, A., and Sheikhalishahi, M. (2017). A distributed framework for collaborative and dynamic analysis of android malware. In *25th Euromicro International Conference on Parallel, Distributed and Network-based Processing, PDP 2017, St. Petersburg, Russia, March 6-8, 2017*, pages 321–328.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Jagannathan, G. and Wright, R. N. (2005). 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*, KDD '05, pages 593–599, New York, NY, USA. ACM.

Jha, S., Kruger, L., and McDaniel, P. (2005). *Privacy Preserving Clustering*, pages 397–417. Springer Berlin Heidelberg, Berlin, Heidelberg.

Martinelli, F., Saracino, A., and Sheikhalishahi, M. (2016). Modeling privacy aware information sharing systems: A formal and general approach. In *15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*.

Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 2(1):86–97.

Oliveira, S. R. M. and Zaïane, O. R. (2002). Privacy preserving frequent itemset mining. In *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining - Volume 14*, CRPIT '14, pages 43–54.

Sheikhalishahi, M. and Martinelli, F. (2017a). Privacy preserving clustering over horizontal and vertical partitioned data. In *2017 IEEE Symposium on Computers and Communications, ISCC 2017, Heraklion, Greece, July 3-6, 2017*, pages 1237–1244.

Sheikhalishahi, M. and Martinelli, F. (2017b). Privacy-utility feature selection as a privacy mechanism in collaborative data classification. In *The 26th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, Poznan, Poland*.

Su, C., Zhou, J., Bao, F., Takagi, T., and Sakurai, K. (2007). Two-party privacy-preserving agglomerative document clustering. In *Proceedings of the 3rd International Conference on Information Security Practice and Experience*, ISPEC'07, pages 193–208, Berlin, Heidelberg. Springer-Verlag.

Vaidya, J. and Clifton, C. (2002). Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 639–644, New York, NY, USA. ACM.