Rings for Privacy: An Architecture for Large Scale Privacy-Preserving Data Mining

Maria Luisa Merani, Daniele Croce, and Ilenia Tinnirello

Abstract—This article proposes a new architecture for privacy-preserving data mining based on Multi Party Computation (MPC) and secure sums. While traditional MPC approaches rely on a small number of aggregation peers replacing a centralized trusted entity, the current study puts forth a distributed solution that involves all data sources in the aggregation process, with the help of a single server for storing intermediate results. A large-scale scenario is examined and the possibility that data become inaccessible during the aggregation process is considered, a possibility that traditional schemes often neglect. Here, it is explicitly examined, as it might be provoked by intermittent network connectivity or sudden user departures. For increasing system reliability, data sources are organized in multiple sets, called rings, which independently work on the aggregation process. Two different protocol schemes are proposed and their failure probability, i.e., the probability that the data mining output cannot guarantee the desired level of accuracy, is analytically modeled. The privacy degree, the communication cost and the computational complexity that the schemes exhibit are also characterized. Finally, the new protocols are applied to some specific use cases, demonstrating their feasibility and attractiveness.

Index Terms—Privacy, secret sharing, data mining, secure multi-party computation, C-means

1 Introduction

Nowadays, the problem of privacy-preserving data mining is crucial for extracting knowledge from users' data while protecting their privacy. On one side, contemporary societies are witnessing an unprecedented availability of data, including personal information from social networks, environmental measurements from smart sensors, state information from systems of different complexity, such as autonomous cars, robots, domestic appliances, Internet of Things (IoT) sensors. On the other side, data owners are often independent individuals or entities that are likely not to trust each other, even if the value coming from data aggregation could be beneficial for all the actors involved in the process of data knowledge extraction. The situation is even more critical for personal data, as new bylaws such as the General Data Protection Regulation (GDPR) of the European Union [1], impose precise obligations on data processing and control.

Although, intuitively, one might expect that processing the informational elements provided by multiple users requires to access the data of each source, a very interesting solution that overcomes such privacy issue is represented by Multi-Party Computation (MPC). Indeed, MPC allows a set of parties to jointly compute a mutually agreed function of their data, while keeping their input data private. Only the final result becomes known to all participants, under some conditions on the maximum number of colluding participants trying to attack the system.

Despite the fact that MPC has been considered impractical for many years, because control on data comes at the price of increasing the storage requirements and the overhead of communication among participants, recent years have witnessed the deployment of several real systems based on MPC. Examples include the set-up of an auction mechanism in Denmark relying on three collaborative parties (rather than on the usual centralized trusted entity) [2], as well as a system for tax fraud detection in Estonia based on the correlation of multiple data sources provided by independent databases [3]. A limitation of current deployments is still the usage of few aggregation parties: for instance, in the auction case, all bidders have to trust that at least two out of the three parties running the auction process are not malicious.

Instead, this paper investigates on the possibility of building a distributed architecture for large-scale MPC, where both the number of data sources and also the number of data aggregators is large [4]. In greater detail, it is assumed that each node providing data also works as an aggregation peer, with the only help of a server storing the intermediate gathered results. Two alternative MPC schemes based on secure sums are proposed, which exploit partial collections and which are completely distributed, therefore necessitating no trusted servers. Additionally, data sources are logically organized in multiple groups called rings, which independently perform partial aggregations. This increases the resiliency of the schemes, that effectively cope with data losses.

The contribution that the paper offers is two-fold:

1) first, it introduces a novel metric to measure the reliability of privacy preserving schemes in the presence of node failures, i.e., when users unpredictably depart

- *Maria Luisa Merani is with the Dipartimento di Ingegneria "Enzo Ferrari", University of Modena and Reggio Emilia, 41121 Modena, Italy, and also with the Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 92092 Puteaux, France. E-mail: marialuisa.merani@unimore.it.*
- *Daniele Croce and Ilenia Tinnirello are with the Engineering Dept., University of Palermo, 90133 Palermo, Italy, and also with the Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 92092 Puteaux, France. E-mail: {daniele.croce, Ilenia.Tinnirello}@unipa.it.*

*Manuscript received 5 Feb. 2020; revised 6 Nov. 2020; accepted 20 Dec. 2020. Date of publication 5 Jan. 2021; date of current version 20 Jan. 2021.
(Corresponding author: Daniele Croce.)
Recommended for acceptance by B. Veeravalli.
Digital Object Identifier no. 10.1109/TPDS.2021.3049286*

or experience intermittent network connectivity. This metric is the *failure probability*, defined as the probability that the data mining scheme leads to inaccurate estimates because of data losses. Indeed, previous studies usually considered the network perfectly reliable [5], [6], or claimed that links can be insecure [4]. Unlike these investigations, the current work assumes data sources are not always available, and lines up with [20], that considers node dropouts as common events.

2) Second, the present paper analytically determines the failure probability that the two proposed schemes exhibit. Moreover, their privacy degree is evaluated, as well as their communication and computational cost. In doing so, a complete theoretical picture of the performance the schemes achieve is offered, in terms of communication cost and computational complexity, as well as privacy and resiliency to failures.

The proposed aggregation protocols are applied to some specific use cases and the findings can be summarized as follows: there exists a trade-off between the failure probability and the privacy degree that the users are guaranteed by the two solutions: the scheme that better protects sensitive data exhibits the highest failure probability. Furthermore, the scheme that is more robust against data losses also displays the highest communication cost.

The rest of the paper is organized along these lines: Section 2 presents some related work and Section 3 provides an introductory background on secure MPC protocols. The privacy-preserving schemes are put forth in Section 4. Their privacy level, robustness, communication and computational costs are analytically determined in Sections 5, 6 and 7, respectively. The proposed protocols are applied to distinctive use cases in Section 8 and their behavior is analyzed in Section 9; the conclusions are gathered in Section 10.

2 Related Work

For privacy-preserving data mining [7], several approaches exist: (i) altering the data before their delivery to the data miner in such a way that the aggregation results are not compromised; (ii) relying upon more sites that have to cooperate to obtain the mining results; (iii) resorting to a machine learning setting where many users collaboratively train a model without exposing their data, as the recently proposed federated learning approach indicates. Data alteration solutions may introduce mining errors, if the alteration is based on random noise [9], while federated learning may result extremely complex in case of homomorphic data encryption and often requires the presence of a central orchestration server [8]. Instead, this work undertakes the pathroad delineated by point (ii) above, and sits among the studies on distributed cooperation mechanisms, that employ secure MPC to warrant users the desired privacy.

TABLE 1
Comparison of Related Works and Their Main Characteristics

Scheme	Circuit	No. of nodes	TTP	Security
P4P [6]	Hybrid	many	Yes	Active
Yao [10]	Boolean	2	No	Passive
GMW [11]	Bool./arithm.	many	No	Active
BGW [12]	Arithmetic	many	No	Active
Cramer [13]	Arithmetic	many	No	Active
FairMP [14]	Boolean	many	Emulated	Passive
VIFF [15]	Arithmetic	many	No	Passive
Sharemind [16]	Hybrid	3	Yes	Passive
Erkin [18]	Arithmetic	many	No	Passive
Vaidya [19]	Arithmetic	many	Yes	Passive
Bonawitz [20]	Arithmetic	many	Yes	Active

The first work proving the feasibility of secure two-party computation is the seminal paper by Yao [10]; the approach has been generalized to multiple parties and different adversary models by Goldreich *et al.* [11], to arithmetic circuits rather than logic circuits by Ben-Or *et al.* [12], and to any linear secret sharing scheme by Cramer *et al.* [13]. Secure computation has been commonly considered too arduous for practical applications: this changed with an implementation of Yao's protocol that Ben-David *et al.* have generalized in the FairplayMP framework [14]. In this respect, VIFF by Damgard *et al.* is another notable example [15]: it is built for asynchronous networks with no notion of rounds, i.e., it guarantees security when calculations are performed in arbitrary order. Similarly to [15], the current work assumes that network nodes are not synchronized: however, its logical ring-based architecture allows to simplify the secure computation protocols, as messages are sent from one node to the next downstream user, thus inherently guaranteeing sequential operations. MPC has also been investigated in Sharemind [16] and P4P [6], by Bogdanov and Duan, respectively. Yet, both works enforce solutions where very few sites collect the users' sensitive data: unfortunately, few and well-known miners might cooperate against users; on the contrary, in the proposed distributed scenario finding colluding mates can turn out very difficult or even impossible. Extensions of MPC schemes introduce the possibility to work simultaneously on multiple data or to verify data integrity of the shares and/or keys. A survey of the approaches proposed so far, which also quantifies their computation and transmission costs, is provided in [17].

Privacy-preserving clustering is also faced in [18] and [19]: these strategies focus on a specific clustering technique and introduce burdensome cryptographic tools besides secure sums. In [18], Erkin *et al.* used homomorphic encryption to privately process the data, while in [19], Vaidya *et al.* employed secure circuits for comparing data vectors under random permutations. Table 1 summarizes the main characteristics of the above approaches, reporting: (i) number of nodes; (ii) presence of Trusted Third Parties (TTP); (iii) type of guaranteed security, that is, whether semi-honest (passive) or malicious (active) attacks are examined. However, differently from the current study, none of the protocols in, e.g., [6], [16], [18] or [19] pay due attention to the reliability issue, that is, to the possibility that users suddenly depart from the network and their data are no longer available.

Finally, the federated learning solution in [20] deserves a mention, as its authors are well-aware of the issues raised by data that become inaccessible during the mining process and explicitly refer to dropped-out users. Yet, their work has no notion of failure probability, which we first introduce in the current study. Moreover, their system is heavily based on the presence of a common server. Rather, our solution preserves privacy in a totally distributed manner, in that conceptually

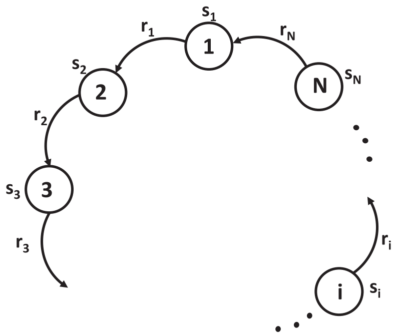


Fig. 1. SSC implementation example.

resembling the fully decentralized/peer-to-peer approach mentioned in [19].

3 Background

A succinct overview of Secure Multi-Party Computation (SMPC) is first provided, in order to ease the understanding of the proposal. SMPC is a field of cryptography devised to compute a function $f(s_1, s_2, \ldots, s_N)$ of the data s_i's (named secrets from now on) collected from N independent nodes, without revealing any information except for the final value of the function. Secure Sum Computation (SSC) [21] is an SMPC example, where the secrets of N parties are privately summed. To implement it in a distributed fashion, one can refer to the scenario depicted in Fig. 1, where each node owns its secret s_i. Assuming that the final sum to be computed, $\sum_{i=1}^{N} s_i$, lies in the range $[0, q)$, which is known, the protocol works as follows:

1) a designated node, e.g., node 1 in Fig. 1, creates a random value r uniformly distributed in $[0, q)$, computes $r_1 = (r + s_1) mod\ q$ and sends it to node 2;
2) node 2 computes $r_2 = (r_1 + s_2) mod\ q$ and sends it to node 3.
3) the generic, ith node computes r_i as $(r_{i-1} + s_i)\ mod\ q$ and sends it to the next downstream node;
4) last node sends $r_N = (r + \sum_{i=1}^{N} s_i)\ mod\ q$ to node 1, which subtracts the initial random value r and determines the sum $\sum_{i=1}^{N} s_i$.

The SSC scheme is asynchronous: it is the reception of the message from node $i-1$ that triggers node i to provide its data. Moreover, the solution is robust for the *honest-but-curious* attack model, i.e., when users try to draw some information through their observations, but they still adhere to the protocol rules. However, it is not secure in case of malicious attacks: if nodes $i-1$ and $i+1$ collude against node i, secret s_i is revealed; similarly, if nodes $i-1$ and i are violated, secret s_i is disclosed too.

An interesting solution to solve this problem is to privately compute the sum employing a linear secret sharing (SS) scheme. An SS scheme is a cryptographic method that allows to split each secret s_i into N_c multiple shares $sh_1(s_i), sh_2(s_i), \ldots, sh_{N_c}(s_i)$ and to recover the original secret if a given number of shares is available. SSC architectures that employ SS methods rely upon the presence of N_c computation peers, which receive different shares of the data provided by the nodes. Each computation peer aggregates the shares that it obtains and makes the resulting sum of shares available to, e.g., a central server. The computation is still secure, owing to the *homomorphic property* that stems from the linearity of the share generation method: as the share of the sum of two secrets $s_i + s_j$ is equal to the sum of their shares, the result of any linear function $f(s_i, s_j)$ can be computed collecting a sufficient number of sums of shares k, without disclosing the original secrets. Hence, for SSC the central server can recover the sum of the secrets $s_1, s_2, \ldots s_N$; depending on the generation scheme, the number of required sums of shares k can be either lower than or equal to N_c.

Trivial Secret Sharing (TSS) is the simplest scheme to generate N_c shares of the secret s_i by using modular sums. In TSS, the ith node with secret s_i randomly selects the first $N_c - 1$ shares with uniform probability in $[0, q)$, then computes the last share as $(s_i - sh_1(s_i) - sh_2(s_i) - \cdots sh_{N_c - 1}(s_i))\ mod\ q$. The method is an (N_c, N_c)-threshold scheme, as all N_c shares distributed to the computation peers are necessary to recover s_i from the modular sum of all shares. Alternative techniques to generate secret shares have been independently proposed by Shamir [22] and Blakley [23] in 1979. In Shamir's solution, node i randomly selects a polynomial $p_i(x)$, whose degree is $k-1$ (with $k \leq N_c$) and whose known term is the secret s_i, that is to say,

$$p_i(x) = s_i + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1}\,, \tag{1}$$

with a_1, a_2, ..., a_{k-1} being the known polynomial coefficients. N_c random shares are generated by the node in the form $sh_x(s_i) = \{x, p_i(x)\ mod\ q\}$, where x is an arbitrary integer and q a prime number greater than both s_i and N_c; usually, x is the identifier of the computation peer who will receive the share. Collecting at least k shares, it is possible to recover the polynomial coefficients by interpolation and to determine s_i as $p_i(0)$. The scheme is classified as a (k, N_c)-threshold scheme, given that k shares are sufficient to recover s_i. Note that both TSS and Shamir's technique are unconditionally secure, i.e., their security does not depend on the computational complexity of a hard problem: rather, their security is guaranteed by information theory.

4 Scenario and Proposal

This Section describes the distributed, privacy-preserving protocol solutions that are proposed to mine the data of a group of users, while respecting their privacy. The focus is on those statistical learning strategies whose update laws require linear operations such as vector addition: on one hand, this allows to exploit the homomorphic property that the previously introduced secret sharing strategies display; on the other, the linear feature is exhibited by several popular data mining algorithms, and therefore does not represent a severe restriction.

It is assumed that a central server, i.e., the data miner, is interested in knowing the average, aggregate behavior of the users, i.e., the sum of their secrets, and that this knowledge has to be acquired without disclosing the single user data. Relying on a privacy-preserving data mining process turns out very useful for both the data miner and the users whose data are being mined; no privacy leakage occurs, an attractive feature given mining techniques typically collect personal, sensitive data, and at the same time accomplish the profiling of the users.

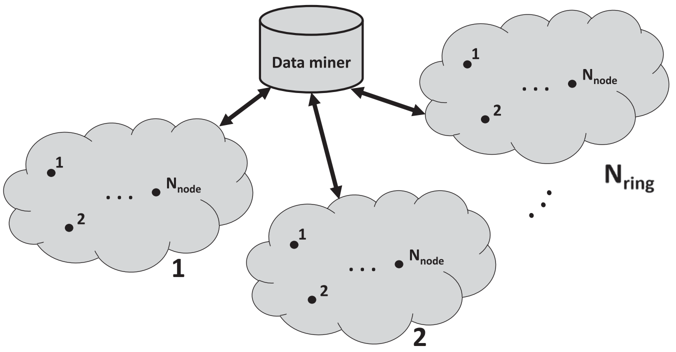


Fig. 2. Multi-ring scenario.

As anticipated before, rather than the large field operations required by public-key cryptography or homomorphic encryption, Shamir's secret sharing scheme is adopted to protect users' data from external and internal attackers.

Let us therefore refer to a multi-ring scenario where, as depicted in Fig. 2, we have:

- one central server, that acts as the data miner and takes on the role of clocking the mining protocol;
- N users (interchangeably termed nodes in what follows) grouped in multiple rings on the basis of their geographical location: we will denote by N_{ring} the number of rings and by $N_{node} = N/N_{ring}$ the number of nodes in each ring (with no loss in generality, N is assumed to be a multiple of N_{ring}).

In this model, the central server is honest-but-curious, so it follows the protocol without cheating, but it is not totally trusted: it might want to access the users' data for its own purposes. Users are modeled as honest-but-curious parties too: each of them can collude with other nodes within the same ring and/or with the server, against one or more victims. Note, however, that the proposed schemes can be easily generalized to cope with other types of malicious attacks through the introduction of well-known cryptographic techniques (verifiable secret sharing), as explained in [17], [24]. For example, data integrity and peer honesty can be assured by using secret and share signatures: secret signatures help verify the correctness of secrets (if shares are corrupted, reconstructed secrets do not match with their signatures), while share signatures could be used to check correctness of shares before reconstructing the secrets. However, the drawback of employing such cryptographic techniques is that the complexity of the secret sharing scheme is increased. Also, note that the security of the proposed schemes is assured by the use of the well-known Shamir scheme.

Importantly, we do not assume that nodes are permanently connected to the network: indeed, (i) users may independently fail and (ii) there is a non-null probability p that the node status is *off*, due to either intermittent network connectivity or sudden departure of the user from the network. Examples are nodes connected through wireless links or users of a peer-to-peer overlay network [25], [26].

The unstable network scenario that has just been depicted can severely impair the effectiveness of the proposed solutions and, in general, can affect any MPC-based scheme: we therefore evaluate not only their privacy degree, but also assess their robustness against the loss of a fraction of users' data. The communication and computational costs have to be determined too, in order to understand the price to be paid to guarantee the desired privacy level. Finally, note that being robust to departures has the positive side-effect to provide an intrinsic protection from malicious users that might partially compromise communications or data, offering the possibility to exclude compromised or missing data from the final result.

4.1 Base Scheme

In this protocol scheme, that we call the Base Scheme, we propose to employ Shamir's technique and take advantage of its homomorphic property separately in each ring. The scheme is composed of two phases: during the first phase, called Distribution Phase (DP), N_{node} shares are created and distributed among the users belonging to the same ring; during the second phase, termed Collection Phase (CP), k sums of shares are delivered to the server, $k \leq N_{node}$, reconstructing the aggregated data without privacy leakage.

4.1.1 Distribution Phase

The DP starts when the server randomly triggers a node within each ring. We denote this node as node i and detail the operations it performs in the following:

1) node i makes N_{node} shares of its secret data s_i following Shamir (k, N_{node})-threshold scheme, using the identifiers of the nodes in its ring, i.e., $j = 0, 1, \ldots, i, \ldots, (N_{node} - 1)$, as seeds;
2) it keeps for itself share $sh_i(s_i) = [i, p_i(i) \ mod \ q]$;
3) it sends share $sh_j(s_i) = [j, p_i(j) \ mod \ q]$ to node j, $\forall j$, $j = 0, 1, \ldots, (N_{node} - 1)$, $j \neq i$.

When receiving the share from node i, every other node in the ring learns that it is time to compute the shares of its secret: it therefore behaves as node i, retaining the share computed with its identifier as seed and sending the remaining shares to the proper nodes within the ring.

Once node i has received the $(N_{node} - 1)$ shares from the nodes within its same ring, it sums them up and determines $Sh_i = \sum_{j=0}^{N_{node}-1} sh_i(s_j)$ which, owing to the homomorphic property, is a share of the sum of the secrets. Similarly, every node in the ring receives $(N_{node} - 1)$ shares from all other nodes and determines a different share.

4.1.2 Collection Phase

Now the CP begins. So, node i sends its contribution, Sh_i, to its downstream node, $i + 1$, that concatenates Sh_{i+1} and forwards the output to the next node. This phase goes on until k node contributions are concatenated; if a downstream node is *off* or has not received all the shares from the other nodes, it will be skipped. The node that concatenates the kth share sends them all to the central server, that can therefore successfully determine S_{ring}, the sum of the secret data for the N_{node} users belonging to the examined ring,

$$S_{ring} = \sum_{i=1}^{N_{node}} s_i \, . \tag{2}$$

It is straightforward to take into account the presence of more rings: the contributions of all rings have to be gathered, where each of them is in the form of (2),

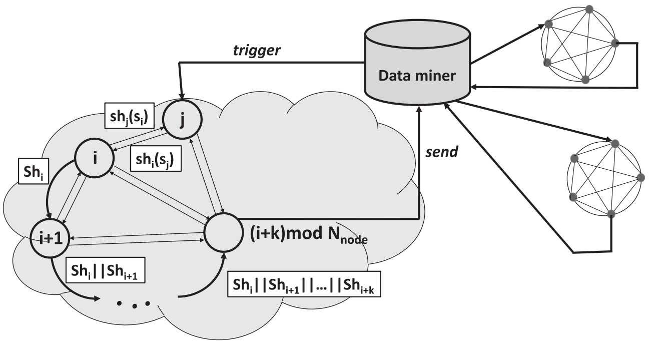


Fig. 3. Base scheme.

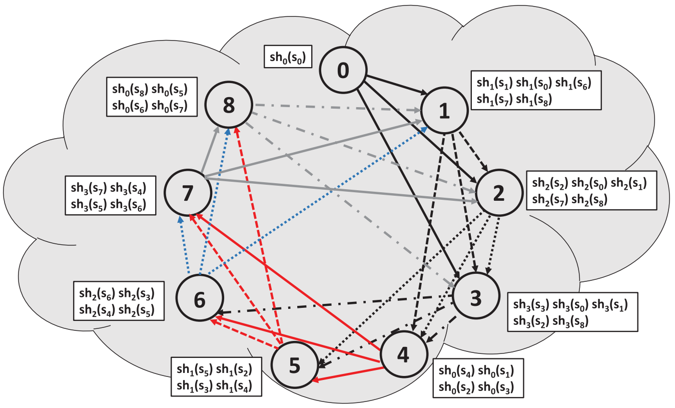

$sh_0(s_0)$
0
$sh_0(s_8)$ $sh_0(s_5)$ $sh_0(s_6)$ $sh_0(s_7)$
8
1
$sh_1(s_1)$ $sh_1(s_0)$ $sh_1(s_6)$ $sh_1(s_7)$ $sh_1(s_8)$
$sh_3(s_7)$ $sh_3(s_4)$ $sh_3(s_5)$ $sh_3(s_6)$
7
2
$sh_2(s_2)$ $sh_2(s_0)$ $sh_2(s_1)$ $sh_2(s_7)$ $sh_2(s_8)$
$sh_2(s_6)$ $sh_2(s_3)$ $sh_2(s_4)$ $sh_2(s_5)$
6
3
$sh_3(s_3)$ $sh_3(s_0)$ $sh_3(s_1)$ $sh_3(s_2)$ $sh_3(s_8)$
$sh_1(s_5)$ $sh_1(s_2)$ $sh_1(s_3)$ $sh_1(s_4)$
5
4
$sh_0(s_4)$ $sh_0(s_1)$ $sh_0(s_2)$ $sh_0(s_3)$

Fig. 4. Enhanced scheme: DP example when $N_{node} = 9$ and $z = 4$.

$$S_{out} = \sum_{j=1}^{N_{ring}} S_{ring_j}\,. \tag{3}$$

The whole procedure is graphically summarized in Fig. 3.

4.2 Enhanced Scheme

When the number of nodes within each ring grows, the DP of the Base Scheme becomes considerably burdensome. To relieve it, we observe that it is not necessary that the generic node sends its secret data shares to every other node belonging to its same ring. Accordingly, we modify the previous proposal and employ a (k', z)-threshold scheme, with $z < N_{node}$ and $k' \leq z$, that we name Enhanced Scheme.

4.2.1 Distribution Phase

In every ring, we group nodes in z different sets I_r, $r = 0, 1, \ldots, (z-1)$, based on the node identifier, j, so that set I_r includes all nodes such that their identifier satisfies

$$I_r = \{\text{all nodes with identifier } j \,|\, j \; mod \; z = r\}. \tag{4}$$

As a toy example, when $N_{node} = 9$ and $z = 4$, we have $I_0 = \{0, 4, 8\}$, $I_1 = \{1, 5\}$, $I_2 = \{2, 6\}$ and $I_3 = \{3, 7\}$. Now we require node i to interact with a reduced number of nodes. In particular, node i:

1) makes z shares, using the set identifier r, $0 \leq r \leq (z-1)$, as seed: $sh_r(s_i) = [r, p_i(r) \; mod \; q]$;
2) keeps the share evaluated in $r = i \; mod \; z$;
3) sends $sh_r(s_i)$ to a randomly selected node within set $I_r, \forall r \in [0, (z-1)]$ and $r \neq i \; mod \; z$;

Resuming the previous example, node 1 of Fig. 4 keeps for itself $sh_1(s_1)$ and might choose to send the remaining $z - 1 = 3$ shares as follows: share $sh_0(s_1)$ to node 4 in I_0, share $sh_2(s_1)$ to node 2 in I_2 and share $sh_3(s_1)$ to node 3 in I_3, as the red lines in Fig. 4 indicate; this is a possible example, but any combination fulfilling the previous constraints is equally acceptable. At the end of the DP, nodes within the same set I_r will possess only shares evaluated in $r = i \; mod \; z$. Note that some nodes within a set might receive no shares at all (as for node 0 in Fig. 4).

4.2.2 Collection Phase

Next, the CP begins: within set I_r, each node sums to the share it kept for itself the shares that it might have received, to compute a partial sum, Sh_i for node i. Then, such partial sums Sh_i's have to be collected, in order to determine $Sh_{I_r} = \sum_{i \in I_r} Sh_i = \sum_{i=0}^{N_{node}-1} sh_r(s_i)$. To achieve this, a possible choice is to accumulate them in a round robin manner, starting from a node triggered by the central server, that transmits its contribution to the next downstream node, and gradually covering all other nodes within the set. Continuing with the previous toy example, given that within set I_0 node 0 is triggered, this node sends Sh_0 (which is the sum of its share $sh_0(s_0)$ plus the shares that it might have received during the previous DP) to node 4; node 4 then adds Sh_4 and forwards everything to node 8, which finally adds Sh_8. It is up to node 8 to deliver the partial sum Sh_{I_0} that it has computed to the server. The remaining sets behave similarly, as portrayed in Fig. 5. It is then sufficient that any k' of such sums of shares computed in k' distinct sets of the same ring be transferred to the central server for it to recover S_{ring}, the sum of the N_{node} users' data belonging to the examined ring.

In analogy to the Base Scheme, the Enhanced Scheme procedure is replicated in every ring and the aggregated sum S_{out} of all data users is finally collected at the central server. Note that, as desired, in each ring the sum of the data is retrieved without disclosing any single contribution to the server, thus protecting users' privacy.

5 Privacy Analysis

As regards privacy, we begin by observing that for the Base Scheme, during the DP and CP each user owns N_{node} shares

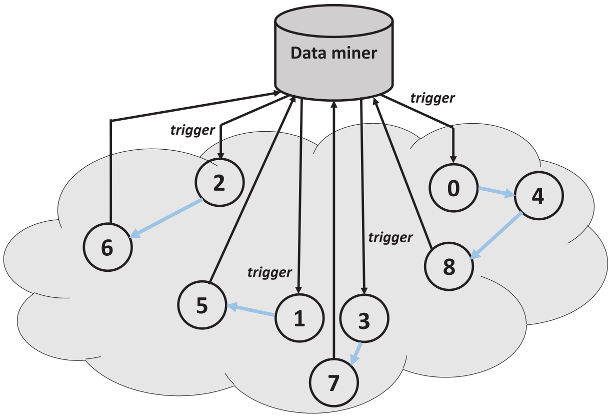


Fig. 5. Enhanced scheme: CP example from a generic ring when $N_{node} = 9$, $z = 4$ and $k' = 2$.

of N_{node} different secrets: not enough to recover any valuable information about other users. The server only knows partial sums and cannot recover the data of the single user either. However, if at least k nodes in a ring collude in a coalition, they can collectively aggregate k shares from each honest party, and gain access to the private data of the $N_{node} - k$ honest users. In other words, for the Base Scheme the probability P_v of disclosing the secrets of v honest nodes in presence of ϕ colluding nodes in a ring is

$$P_v = \begin{cases} 0\,, & \phi < k \\ 1\,, & \phi \geq k\,, \quad \forall v\,, \quad v = 0, 1, \ldots, N_{node} - k \end{cases}. \tag{5}$$

The value of k has therefore to be carefully chosen, as higher k values guarantee a higher confidentiality degree. On the other hand, such values result in a scheme that is weaker with respect to the loss of shares, as next Section will point out.

Regarding the Enhanced Scheme, to discover all secret data in a ring with probability 1, the minimum condition to fulfill is that all nodes of $k^{'}$ arbitrary sets collude: if, for the sake of clarity, we examine the circumstance where the sets have equal cardinality, given by $|I_r| = \frac{N_{node}}{z}$, $\forall r$, $r = 0, 1, \ldots, z-1$, then $k^{'} \cdot |I_r|$ *strategically* placed nodes are needed. More generally, ϕ colluding nodes in a ring of the Enhanced Scheme, with $\phi \geq k'$, will be able to disclose the secrets of v honest nodes, $1 \leq v \leq (N_{node} - \phi)$, with probability P_v, $P_v \leq 1$, that depends on nodes deployment inside the ring. It is therefore interesting to determine such probability, that coincides with the probability that each of the v honest nodes sends at least k' shares to k' colluding nodes. Assuming that the colluding nodes are uniformly distributed within the z sets, so that the probability $\tilde{p}$ that an honest node sends a share to a colluding user is a constant, then the probability $\bar{P}$ that an honest node sends at least $k^{'}$ shares to $k^{'}$ colluding nodes may be computed as

$$\bar{P} = \sum_{w=k^{'}}^{z-1} \binom{z-1}{w} \tilde{p}^{w} \cdot (1-\tilde{p})^{z-1-w}\,; \tag{6}$$

It follows that P_v can be expressed as

$$\tilde{P}_v = \binom{N_{node} - \phi}{v} \bar{P}^{v} (1-\bar{P})^{N_{node} - \phi - v}\,. \tag{7}$$

Unfortunately, there exists no clean, closed-form expression for P_v when the sets are not uniformly polluted by malicious nodes. However, in the Numerical Results Section P_v behavior will be assessed through a simulative approach, taking into account different k' values and different, random placements of the colluding nodes; furthermore, $\tilde{P}_v$ will be compared against P_v.

6 Failure Probability

In this Section, the focus is shifted on the reliability that the two proposed schemes exhibit in scenarios with intermittent network connectivity or sudden user departures. As a matter of fact, during the scheme execution, it might happen that the contribution of a node cannot be included in the final sum because: (i) some of the shares have not been received due to a communication failure; (ii) the node itself has gone down due some hardware failure; (iii) the node has voluntarily departed from its ring. Thus, we examine the condition when the number of nodes whose data cannot be included in the final sum is greater than or equal to N', where $N' - 1$ indicates the maximum tolerated number of missing data. As a general example, we state that, when the output of the mining process provides an estimate $\mathbf{E}^{'} = [e_1^{'}, e_2^{'}, \ldots, e_n^{'}]$ in the euclidean space $\mathcal{R}^n$, whose a-norm distance d from the estimate $\mathbf{E} = [e_1, e_2, \ldots, e_n]$ drawn from the totality of the users' data is greater than or equal to a fixed δ,

$$d = \left(\sum_{k=1}^{n} |e_k - e_k^{'}|^a \right)^{\frac{1}{a}} > \delta\,, \tag{8}$$

then the mining output is no longer acceptable. We therefore define the failure probability P_{fail} as

$$P_{fail} = Pr\{\text{number of lost data} \geq N^{'}\}\,, \tag{9}$$

and proceed to its evaluation.

We first begin by determining the probability $P_{fail-ring}$ that the data of the users belonging to the same ring cannot be retrieved. In doing so, it is assumed that the data of the single ring are either totally lost or entirely recovered. In other words, we do not consider the circumstance where a partial recovery of the data from a ring occurs. In particular, we observe that

- if the DP fails, the CP does not even begin;
- if the DP does not fail, the CP might in turn fail.

Let us indicate by $P_{fail-DP}$ the probability that the DP fails and similarly by $P_{fail-CP}$ the probability that the CP fails. Assuming that the two phases are independent, we can compute the probability $P_{fail-ring}$ that the partial sum provided by a generic ring is not available at the data miner as

$$P_{fail-ring} = P_{fail-DP} + (1 - P_{fail-DP}) \cdot P_{fail-CP}, \tag{10}$$

where $P_{fail-DP}$ and $P_{fail-CP}$ take on different expressions for the Base and the Enhanced schemes.

Next, it is observed that the data delivered to the data miner from each ring allows to recover a partial sum, S_i for the ith ring, where such sum takes into account the contributions stemming from N_{node} distinct users. In the ideal case, the sum of N data will be available at the data miner; yet, as the partial sums of some rings might be missing, in turn S_{out} reduces to the sum of $N - N_{node}$, $N - 2N_{node}$, $\ldots$, 0 data, if $1, 2, \ldots, N_{ring}$ rings do not provide their contribution, respectively.

It follows that P_{fail} is expressed by

$$P_{fail} = \sum_{i=n_{mul}}^{N_{ring}} Pr\{\text{the data of } i \text{ rings are unavailable}\}\,, \tag{11}$$

where

$$n_{mul} = \left. \frac{N^{'} - 1}{N_{node}} \right\rfloor, \tag{12}$$

and

$$Pr\{\text{the data of } i \text{ rings are unavailable}\} = \binom{N_{ring}}{i} P^{i}_{fail-ring}(1-P_{fail-ring})^{N_{ring}-i}, \tag{13}$$

with $P_{fail-ring}$ given by (10). In the following subsections, $P_{fail-DP}$ and $P_{fail-CP}$ will be specialized for the two proposed privacy-preserving schemes as a function of the probability p that a node fails or the communication is interrupted during the aggregation process (from now on, p is named *off* probability).

6.1 Base Scheme

For this scheme, observe that $P_{fail-DP}$ has the following meaning:

$$P^{(base)}_{fail-DP} = 1 - Pr\{\text{at least } k \text{ nodes within the ring receive the shares from all nodes}\}, \tag{14}$$

and it is therefore given by

$$P^{(base)}_{fail-DP} = 1 - \sum_{i=k}^{N_{node}} \binom{N_{node}}{i} (1-p)^{N_{node}\cdot i}(1-(1-p)^{N_{node}})^{N_{node}-i}. \tag{15}$$

On the other hand, when it comes to the CP of the Base Scheme, $P^{(base)}_{fail-CP}$ is

$$P^{(base)}_{fail-CP} = \sum_{i=N_{node}-(k-1)}^{N_{node}} \binom{N_{node}}{i} p^{i}(1-p)^{N_{node}-i}. \tag{16}$$

6.2 Enhanced Scheme

For this scheme, $P_{fail-DP}$ specializes to the following definition:

$$P^{(en)}_{fail-DP} = 1 - Pr\{\text{at least } k' \text{ of the } z \text{ subsets receive all shares from all nodes}\}, \tag{17}$$

and therefore turns out to be

$$P^{(en)}_{fail-DP} = 1 - \sum_{i=k'}^{z} \binom{z}{i} (1-p)^{N_{node}\cdot i}(1-(1-p)^{N_{node}})^{z-i}, \tag{18}$$

whereas for the CP of the Enhanced Scheme, $P^{(en)}_{fail-CP}$ is provided by

$$P^{(en)}_{fail-CP} = \sum_{i=z-(k'-1)}^{z} \binom{z}{i} p^{i}_{set}(1-p_{set})^{z-i}, \tag{19}$$

where p_{set} is the probability that the collection of shares within a set does not succeed. Since the shares are collected in a round-robin manner, with the active participation of all nodes in a set, the collection fails if at least one node fails, hence,

$$p_{set} = 1-(1-p)^{\frac{N_{node}}{z}}. \tag{20}$$

7 Communication Cost and Complexity

Lastly, communication cost and computational complexity of both schemes are determined. With reference to the former cost, we measure it by the number of connections required for all intra-ring and server-to-ring communications.

7.1 Base Scheme

Regarding the communication cost of the (k, N_{node})-threshold Base Scheme, in each ring the DP needs:

1) one connection to trigger one randomly chosen node within the ring;
2) $\frac{N_{node}(N_{node}-1)}{2}$ connections to distribute the shares among all nodes in the ring.

During the CP, it is simply necessary to take into account $k-1$ connections to collect the shares of the sum inside the ring and 1 connection to send them to the server. The communication cost of the Base Scheme, C_{base}, is therefore

$$C_{base} = \left[1 + \frac{N_{node}(N_{node}-1)}{2} + k\right] N_{ring}. \tag{21}$$

As regards computational complexity, during the DP each node has first to compute N_{node} shares of the secret, then it has to sum one of these with the $N_{node}-1$ shares received by the other nodes. These operations are briefly recalled below, along with their contribution to complexity:

1) evaluate a $(k-1)$-order polynomial for an integer value, so as to determine one share of the secret. The complexity is $O(k-1)$;
2) repeat the previous operation N_{node} times;
3) sum one of the shares with the $N_{node}-1$ shares the node received from the other nodes of the ring. The complexity is $O(log(N_{node}-1))$.

Thus, for the single node the computational complexity is approximated by $O(N_{node}k)$, so that the complexity of the entire DP within a ring is $O(N^2_{node}k)$. During the following CP, the server has to compute one polynomial over k points; here the computational complexity is $O(k)$, which is negligible with respect to the previous term. Since the above operations have to be repeated for each ring, the overall computational complexity of the base scheme becomes $O(N_{ring}N^2_{node}k)$.

7.2 Enhanced Scheme

The Enhanced Scheme exhibits a different communication cost, C_{en}, that can be lower than C_{base}, depending on the z and k' values employed. To determine C_{en}, it is observed that during the DP, in each ring

1) the server has to trigger all nodes to have them start distributing their shares: this requires N_{node} connections;
2) then, $(z-1)\cdot N_{node}$ connections are needed for the distribution of the shares from the N_{node} nodes to $z-1$ different sets.

In the Enhanced Scheme, observe that nodes have no knowledge about the actual number of shares they will receive. So, CP will begin after an adequately long time interval, when

1) the server contacts k' nodes in k' different sets, in order to trigger the share collection within each set: this requires k' connections;

2) next, all nodes within a set will be contacted and their partial sums gathered: this requires $|I_r| - 1$ connections in the generic set I_r;
3) finally, k' connections will be employed to return to the server the sum of the contributions of each triggered set.

The communication cost of the Enhanced Scheme, C_{en}, is therefore written as

$$C_{en} = \left[N_{node} \cdot z + 2k^{'} + \sum_{r \in D^{(k^{'})}} (|I_r| - 1) \right] \cdot N_{ring}, \tag{22}$$

where $D^{(k^{'})}$ identifies the $k^{'}$ sets that the server selected.

As C_{en} cost depends on $k^{'}$, to pursue an easy-to-interpret comparison against C_{base}, the maximum value of C_{en}, C^{*}_{en}, is examined, where C^{*}_{en} corresponds to $k' = z$

$$\begin{aligned} C^{*}_{en} &= \left[z \cdot (N_{node} + 2) + \sum_{r \in D^{(z)}} (|I_r| - 1) \right] \cdot N_{ring} \\ &= \Big[z \cdot (N_{node} + 2) + (N_{node} - z) \Big] \cdot N_{ring} \, . \end{aligned} \tag{23}$$

We outline that C^{*}_{en} takes on a lower value than C_{base} if

$$z(N_{node} + 1) + N_{node} \; < \; 1 + \frac{N_{node}(N_{node} - 1)}{2} + k \, , \tag{24}$$

that, after a few algebraic passages, leads to

$$z < \frac{2(k+1) + N^2_{node} - 3N_{node}}{2(N_{node} + 1)} \, . \tag{25}$$

If last inequality is satisfied, the Enhanced Scheme always warrants a reduced communication cost than the base solution; the saving is the smallest when $k^{'} = z$, then it increases for increasing values of $k^{'}$, as the evaluations reported in the Numerical Results Section will quantify.

To evaluate the computational complexity of the enhanced scheme, in analogy to the base scheme, observe that the operations each node has to perform are:

1) evaluate a $(k^{'} - 1)$-order polynomial to determine one share. The complexity is $O(k^{'} - 1)$;
2) repeat operation 1) z times;
3) sum the node's own share with the shares received from the other subsets. Such shares vary between a minimum of 0 and a maximum of $z - 1$. In the worst-case, the node has to compute a sum of z shares, whose complexity is $O(log(z - 1))$.

During the DP, the complexity that a node has to budget is therefore approximated by $O(zk^{'})$, and the complexity of the entire DP is $O(N_{node}zk^{'})$. During the CP, in each subset the sum of all shares is computed, the complexity ranging between 0 and $O(log(z - 1))$, that corresponds to the previous worst-case; next, z polynomials over $k^{'}$ points have to be computed, whose complexity is $O(z \cdot log(k^{'} - 1))$, negligible with respect to the DP complexity. Thus, if multiple rings are used, the overall complexity becomes $O(N_{ring}N_{node}zk^{'})$, which is always lower than the base scheme complexity, as $z < N_{node}$ and $k^{'} < k$.

TABLE 2
Per Node Communication Cost and Computational Complexity

	Base	Enhanced
Communication cost	$O(N_{node})$	$O(z)$
Computational complexity	$O(kN_{node})$	$O(k^{'}z)$

Table 2 summarizes the per-node communication cost and computational complexity for the base and the enhanced schemes. Although in line with other state of art techniques [17], one advantage of the proposed solutions is that they evenly distribute the computational load among all nodes, without relying on computational servers.

Lastly, we compare the costs of the proposed schemes to those displayed by the federated learning solution in [20]. Since this scheme relies on a centralized server, the solution displays distinct communication and computational costs for the nodes and for the server. For a node in [20], the communication and complexity costs are $O(N_{node})$ and $O(N^2_{node})$ respectively, much higher than the proposed schemes (for the sake of simplicity, the comparison neglects m, the length of the data). Significant burden is also placed on the server, with communication cost of $O(N^2_{node})$ and computational complexity also $O(N^2_{node})$. This is far higher than the server cost of the proposed base scheme, that amounts to a negligible $O(1)$ for communication (only 2 connections needed) and to $O(k)$ for complexity, while, for the enhanced scheme, the server costs are $O(k^{'})$ and $O(zk^{'})$, respectively.

8 Applying the Scheme

8.1 Fuzzy C-Means With Privacy

As a concrete example of data mining tool that can successfully leverage the privacy preserving schemes discussed before, we consider one of the most widely used clustering algorithms, Fuzzy C-Means (FCM for short) [27]. FCM is an unsupervised soft clustering strategy, that attempts to partition a finite collection N of data elements into K fuzzy clusters. Without loss of generality, assume that each of the data elements is an array of size $1 \times M$ and indicate by $\mathbf{d}_i = [d_{i1}d_{i2} \dots d_{iM}]$ the data element provided by the ith user (from now onward the usage of bold will point to either a vector or a matrix). The FCM algorithm returns:

- K centroids, $\mathbf{c}_1$, $\mathbf{c}_2$, …, $\mathbf{c}_K$, $\mathbf{c}_j = [c_{j1}c_{j2} \dots c_{jM}]$, where $\mathbf{c}_j$ is the representative element of the jth cluster, and
- matrix $\mathbf{U}$, $N \times K$ in size, whose generic element u_{ij}, $u_{ij} \in [0, 1]$, represents the membership degree of the $\mathbf{d}_i$ data element to the jth cluster.

The ith row of the membership matrix $\mathbf{U}$, that we indicate by the symbol $\mathbf{u}_i = [u_{i1}u_{12} \dots u_{iK}]$, holds the memberships of the ith user to the various clusters. We refer to it as to the ith user membership vector and observe that the constraint $\sum_{j=1}^{K} u_{ij} = 1$ has to be respected.

The steps that FCM goes through are succinctly described next:

1) Select the number of clusters K $(2 \leq K < N)$, the fuzziness parameter f (in literature, the value $f = 2$ is often encountered) and the termination criterion ϵ. Set the iteration index t to $t = 0$ and randomly initialize the membership matrix $\mathbf{U} \rightarrow \mathbf{U}^{(\mathbf{0})}$.

2) at step t, compute the centroids as

$$\mathbf{c}_j^{(t)} = \frac{\sum_{i=1}^{N} \left(u_{ij}^{(t)}\right)^f \cdot \mathbf{d}_i}{\sum_{i=1}^{N} \left(u_{ij}{}^{(t)}\right)^f} \quad \text{for every j , } j = 1, 2, \ldots, K\,; \tag{26}$$

3) for every i and j pair, also update the generic element $u_{ij}^{(t)}$ of matrix $\mathbf{U}^{(t)}$ as follows:

$$u_{ij}^{(t)} = \frac{1}{\sum_{k=1}^{K} \left(\frac{\|\mathbf{d}_i - \mathbf{c}_j^{(t-1)}\|}{\|\mathbf{d}_i - \mathbf{c}_k^{(t-1)}\|}\right)^{\frac{2}{f-1}}}\,, \tag{27}$$

where $||\cdot||$ indicates the euclidean norm;

4) for $t \geq 1$, if $\|\mathbf{U}^{(t+1)} - \mathbf{U}^{(t)}\| \leq \epsilon$, stop; otherwise set $t = t + 1$ and return to step 2.

Once the algorithm execution has come to an end, $\mathbf{d}_i$, hence the ith user, is assigned to the cluster whose membership degree is the highest, that is, $\max_j\{u_{ij}\}$.

Given the final aim is to profile the users without violating their privacy, the most conservative approach is taken and it is therefore assumed that the profiling server be exclusively interested in the centroids determination, whereas it is not allowed to access either the users' data or their membership vectors. Accordingly, a possible implementation of the algorithm that is totally trusted and privacy respectful is discussed next. To this end, the profiling server is required to be responsible for updating the centroid vectors only, as step 2 of the algorithm mandates, and for broadcasting them to the nodes, that are responsible for the $\mathbf{U}^{(t)}$ matrix update at step 3; for the same privacy reasons, it is node i only that reads and modifies its membership vector $\mathbf{u}_i$. To allow the server to iteratively compute the centroid vectors, from (26) we observe that at step t of the algorithm the elements that the ith user has to convey to the server are those grouped in the following matrix $\mathbf{s}_i^{(t)}$:

$$\mathbf{s}_i^{(t)} = \begin{pmatrix} (u_{i1}^{(t)})^f d_{i1} & \ldots & (u_{iK}^{(t)})^f d_{i1} \\ \vdots & \ddots & \vdots \\ (u_{i1}^{(t)})^f d_{iM} & \cdots & (u_{iK}^{(t)})^f d_{iM} \\ (u_{i1}^{(t)})^f & \ldots & (u_{iK}^{(t)})^f \end{pmatrix}; \tag{28}$$

which represents the ith user secret to be protected.

We conclude noting that this privacy-oriented FCM implementation implies that a fraction of its computational burden is placed on nodes, that are responsible for updating their membership vector at every step, until convergence is reached. However, as the profiling server has to only perform additions on the terms it receives, our proposed schemes can be profitably employed, being based on SSC.

8.2 FCM Loss Tolerance

To obtain a reasonable estimate for the maximum number of missing data $N^{'}$ that defines the failure condition in (9), this subsection investigates how robust the FCM centroid evaluation is with respect to data losses. A criterion is first established to determine $N^{'}$; then, several data sets are considered, that display different features in terms of sparseness and cardinality, and the distinctive $N^{'}$ value of each of them is computed.

The criterion works as follows: the centroids are calculated over the complete data set; the set is then reduced by a fixed number α of randomly chosen data samples; the new centroids, $\mathbf{c}_j^{\alpha}$, are determined and so are the $\mathcal{R}^2$ euclidean distances $||\mathbf{c}_j - \mathbf{c}_j^{\alpha}||$, $j = 1, 2, \ldots, K$. As the FCM output can be affected by the specific selection of the removed data, for a given α the procedure is repeated for a sufficiently large number of trials $\mathcal{G}$ and then the ensemble averages $E_{\mathcal{G}}[||\mathbf{c}_j - \mathbf{c}_j^{\alpha}||]$, $j = 1, 2, \ldots, K$, are evaluated. The loss of fewer than $\alpha = N'$ data is deemed acceptable, where N' corresponds to the first value that violates the condition $\max_j E_{\mathcal{G}}[||\mathbf{c}_j - \mathbf{c}_j^{\alpha}||] \leq \delta$, with $\delta = 10^{-2}$; from this point onward, $N^{'}$ is referred to as the FCM loss tolerance threshold. Furthermore, the ϵ value ruling the FCM stop condition is set equal to 10^{-16}, i.e., $\epsilon << \delta$.

The choice of the data sets deserves a few words: two of them are commonly encountered in the literature that deals with clustering and feature extraction, namely, the so-called Iris data set [28] and the banana data set [29]. The Iris data set contains random samples of flowers belonging to three species of iris flowers. Fifty observations of sepal length, sepal width, petal length and petal width are recorded for each species, wherefrom $N = 150$ quadruples are available: as an indicative example, Fig. 6a shows the normalized sepal width on the x-axis and the petal length on y-axis, along with the chromatic indication of the clusters and the centroids that FCM returns; in Fig. 6a through Fig. 6d, the reported values have been normalized with respect to their maximum. The $K = 3$ choice is dictated by the number of species, that is known beforehand. Our tests revealed a loss tolerance threshold $N^{'} = 0.08 \cdot N$. As for the banana set, we used the algorithm in Appendix A.7, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2021.3049286, of [29] to generate $N = 1000$ points scattered around a segment of a circle using the following default parameters: radius of the circle of which the banana is an arch equal to $\rho = 5$, starting and ending angles of the arch $\theta_1 = 9 \cdot (\pi/8)$ and $\theta_2 = 19 \cdot (\pi/8)$ respectively, standard deviation that rules the dispersion of the data around the circle equal to $\sigma = 1$. For this set, FCM determined $K = 4$ clusters whose centroids are shown in Fig. 6b, and the computed loss tolerance threshold is $N^{'} = 0.23 \cdot N$. We also examined two real databases that were made available to us: the database of the viewing habits exhibited by the users of a small, multichannel web-TV platform, that were monitored 24 hours a day for 9 months, and the database of daily methane consumption of 200 customers (schools, factories, medium-to-small enterprises) monitored by a local multi-utility company for 100 days. Fig. 6c graphically illustrates the average session time and the average session number for the viewers of the most popular channel extracted from the data set ($N = 113$); the viewers were deliberately grouped in $K = 2$ clusters and the determined FCM loss tolerance threshold is $N^{'} = 0.09 \cdot N$. Finally, Fig. 6d presents the daily gas consumption on the abscissa and the daily peak consumption on the ordinate drawn from the fourth examined data set, where $N = 2 \cdot 10^4$, its $K = 9$ clusters and the corresponding centroids. Here the loss tolerance threshold is $N^{'} = 0.31 \cdot N$.

Such case studies unveil that FCM loss tolerance in determining the centroids vastly varies, depending on the size and the shape of the data set. This is summarized in Table 3, that reports the number of centroids K, the size N and the loss tolerance threshold $N^{'}$ for each of the considered data sets.

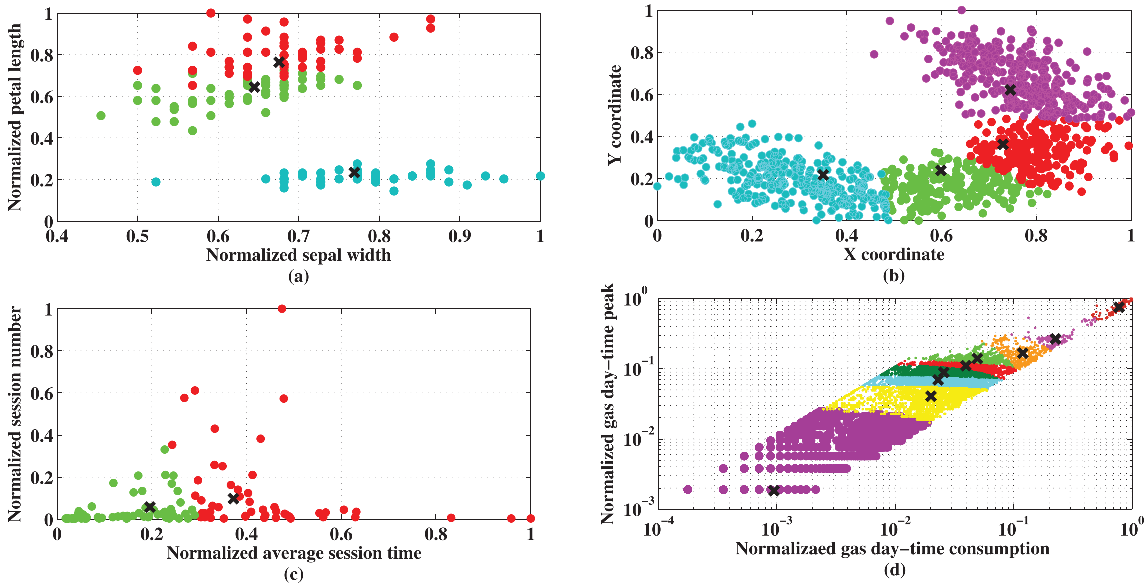


Fig. 6. FCM clustering examples on different data sets.

It is out of the scope of the current work to systematically investigate the previous point: rather, and more pragmatically, we conclude that in our numerical scouting $N^{'}$ fell in the range $[0.08N, 0.31N]$. This indicates that the approximated centroids – modestly – differ from the exact ones, only if a non negligible fraction of the original data is missing. In next Section, the failure probability of the proposed schemes will therefore be evaluated for $N^{'} = 0.2N$, a value that lies in the middle of the above interval.

8.3 Estimating the Node Off Probability p

In order to draw a numerical estimate of the node off probability p for a real use case, several data traces were analyzed, referring to the smart water metering service of a local company in a town close to Turin, Italy. In the examined scenario, metering devices are equipped with a class A Long Range (LoRa) interface [30] for long distance and low power transmissions. They wirelessly upload data to the gateway deployed by the water company. We were allowed to inspect the anonymized packets received from the LoRa gateway, that is run and maintained by an Italian LoRaWAN operator. As LoRa end-devices never voluntarily abandon the network they belong to, the probability p that a node is off coincides with the packet error probability that the transmissions from the metering device experience, which we estimated ex-post through the overall packet error rate. We inspected traffic data over 15 weeks and extracted packet error statistics. The end-devices were configured to send 2 packets per day, that were randomly transmitted choosing one out of three frequency channels in the 868 MHz ISM band. During the 15 weeks of observation, the number of active end-devices was 308, generating a total of 207040 packets.[1] However, only 181257 packets were received correctly. This translates in an estimated off probability $p = 0.125$. Considering that retransmissions are not adopted in the LoRa network under investigation, such value can be interpreted as referring to the worst-case scenario.

TABLE 3
Loss Tolerance Thresholds

Data set	K	N	$N^{'}$
Iris	3	150	$0.08N$
Banana	4	10^3	$0.23N$
WebTV views	2	113	$0.31N$
Methane consumption	9	$2 \cdot 10^4$	$0.09N$

Note that the above setting is meaningful for two distinct reasons: (i) it is a valid example of a scenario where the proposed schemes would consent the extraction of valuable customers' features, without violating the users' privacy; (ii) it allows to estimate p for an IoT architecture based on LoRaWAN, that in recent years has emerged as one of the most widespread solutions for wireless data collection.

9 Numerical Results

The following numerical results illustrate the performance attained by the proposed privacy preserving schemes. We used a custom simulator implemented in MATLAB and we compare the results with the analytical models developed throughout the paper.

The first important aspect to assess is the privacy level the schemes guarantee: in Section 5, their privacy was quantified through the probability P_v of disclosing v secrets in the presence of a given number ϕ of colluding nodes. For the Base Scheme, we recall that k colluding nodes disclose the data of all users with probability 1, that is, $P_v = 1$ when $v = N_{node}$ and $\phi = k$. Conversely, for the Enhanced Scheme Fig. 7 reports the behavior of P_v as a function of v, when $\phi = 10$ and $N_{node} = 30$. The curves refer to $k^{'} = 3, 5, 7, 9$ and $z = 10$ sets. Dashed lines have been obtained by simulation, and display the mean value of P_v, as well as its t-Student 95 percent-confidence intervals, determined from 20 repeated trials for each point. More accurately, in every simulation run $\Phi = 10$ colluding nodes are randomly placed within the z sets of the ring. Then, the honest users randomly choose the nodes in every set to

[1] The total number of packets transmitted by the end-devices was extrapolated using the frame counter of the successfully received ones.

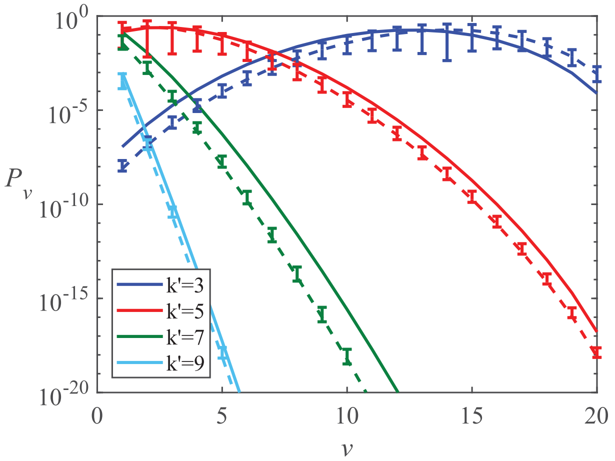


Fig. 7. P_v and $\tilde{P}_v$ as a function of v for the Enhanced Scheme, when $N_{node} = 30$, $k^{'} = 3, 5, 7, 9$, and $\phi = 10$.

send their shares to. If, for every honest node, at least k' shares are delivered to colluding nodes, then the v secrets are revealed and the counter of secret disclosures $m_{disclose}$ is incremented by 1; given a total of m_{config} distinct configurations of colluders are randomly generated, one sample of P_v is numerically determined as the $m_{disclose}/m_{config}$ ratio. The procedure is repeated 20 times, in order to evaluate the mean and the confidence intervals reported in the figure.

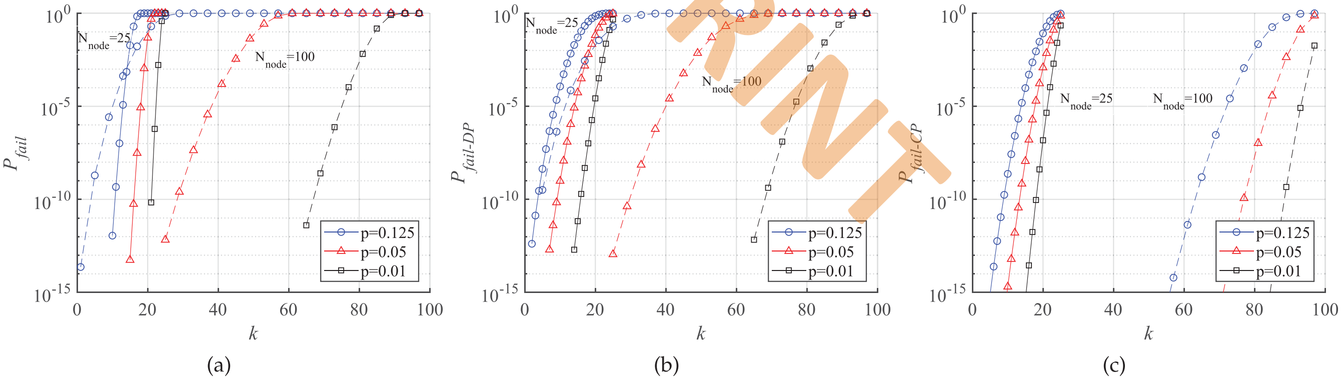
The solid curves show the behavior of $\tilde{P}_v$, as introduced in (7), and offer an approximation to P_v. Provided that $k^{'}$ is high enough, e.g., $k^{'} = 9$, $\tilde{P}_v$ values lie really close to their simulated counterparts, but $\tilde{P}_v$ approximation is satisfying for $k^{'} < 9$ too. We also observe that for increasing values of $k^{'}$, P_v takes on really modest values. On the other hand, increasing $k^{'}$ threshold also increases the failure probability, as it will be shown next.

The second relevant aspect to consider is the reliability of the schemes in aggregating a sufficiently high number of data or, in a specular manner, their failure in doing so. We quantify this through the failure probability P_{fail} defined in (9) and, as discussed in Section 8.2, in the next experiments we assume that the maximum number of tolerated losses is equal to $N^{'} = 0.2 \cdot N$.

For the Base Scheme, Figs. 8a, 8b and 8c show P_{fail}, $P_{fail-DP}$ and $P_{fail-CP}$ as a function of k, respectively. The curves in the figures refer to two scenarios, where $N = 500$ nodes have been organized into either 20 rings with $N_{node} = 25$ each (solid lines) or into 5 rings with $N_{node} = 100$ (dashed lines). Different values for the probability p that a generic node is off are considered, namely, 0.01, 0.05 and 0.125. The last value has been selected having in mind the worst-case scenario discussed in Section 8.3, whereas the remaining values refer to more benevolent settings.

First, note that the DP has a higher failure probability than the CP, and that sensitivity to p is much more pronounced for $N_{node} = 25$ than for $N_{node} = 100$. These remarks are not surprising, as the DP requires all N_{node} nodes to be active while distributing their shares and accepting the shares of all other nodes. In turn, the higher DP vulnerability reflects in P_{fail} and $P_{fail-DP}$ curves being at close range. We also observe that when $N_{node} = 25$, a careful selection of k confines P_{fail} to very low values, even for $p = 0.125$. On the other hand, if $N_{node} = 100$ and $p = 0.125$, P_{fail} is non-negligible even with low values of k.

For the Enhanced Scheme, using the same N and N' values as before, Figs. 9a, 9b, and 9c report P_{fail}, $P_{fail-DP}$ and $P_{fail-CP}$

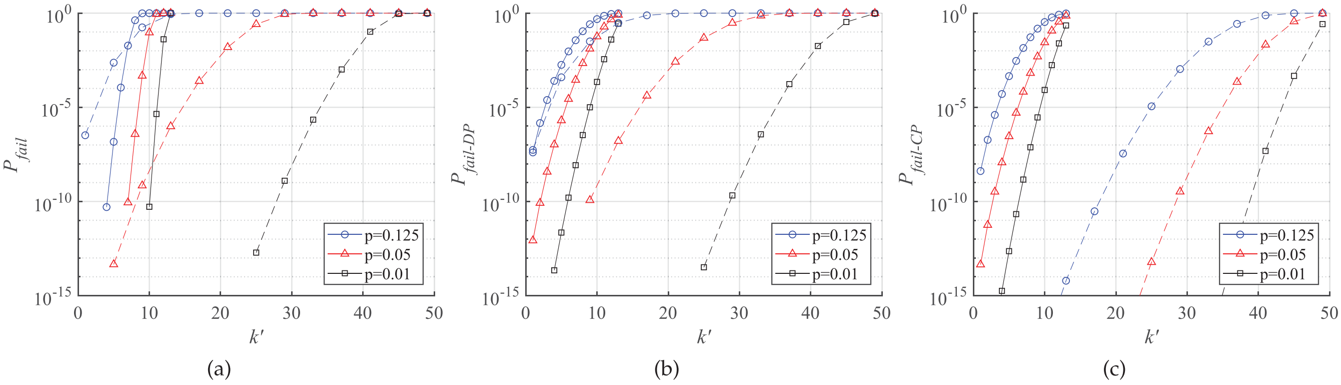


(a)

(b)

(c)

Fig. 8. P_{fail}, $P_{fail-DP}$ and $P_{fail-CP}$ as a function of k for the Base Scheme.

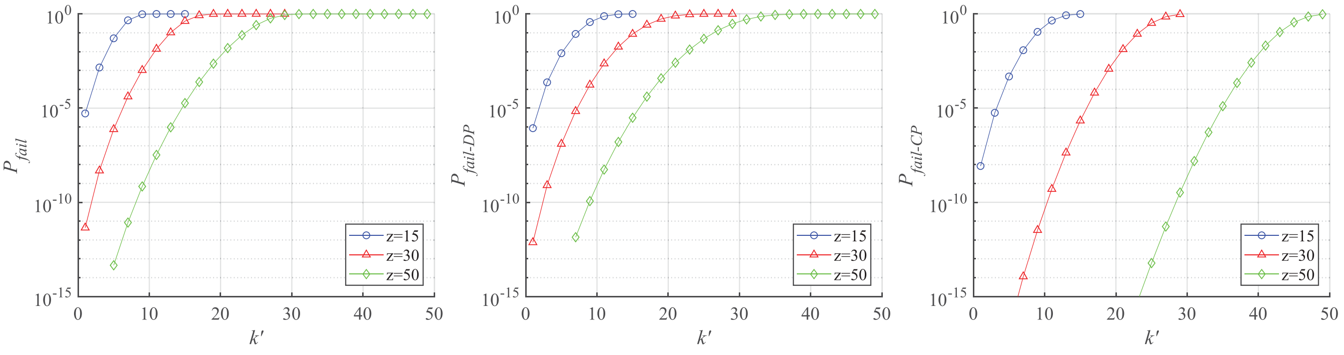


(a)

(b)

(c)

Fig. 9. P_{fail}, $P_{fail-DP}$, and $P_{fail-CP}$ as a function of $k^{'}$ for the Enhanced Scheme.



(a) (b) (c)

Fig. 10. P_{fail}, $P_{fail-DP}$, and $P_{fail-CP}$ as a function of $k^{'}$ for the Enhanced Scheme, $N_{node} = 100$, $p = 0.05$, and $z = 15, 30, 50$.

as a function of $k^{'}$, respectively. Solid lines refer to $N_{node} = 25$ and $z = 10$, whereas dashed lines refer to $N_{node} = 100$ and $z = 50$. These parameters are chosen to have the same number of nodes, $(N_{node}/z) = 2$, in every set. As before, $P_{fail-DP}$ is predominant in P_{fail} evaluation. The comparison between Figs. 8 and 9 reveals that, when $k = k^{'}$, so that the two protocols use the same number of minimum required shares, the Base Scheme exhibits lower failure probability than the Enhanced Scheme. However, as it will be shown later, this comes at the cost of a much higher communication overhead. Additionally, for a given N_{node} value, the Enhanced Scheme failure probability can be confined by decreasing $k^{'}$ and/or increasing z. This is shown in Figs. 10a, 10b, and 10c, where the effects of three distinct values of z on P_{fail}, $P_{fail-DP}$ and $P_{fail-CP}$ are considered, for $N_{node} = 100$ and $p = 0.05$.

Another important figure of merit for the proposed privacy preserving solutions is their communication cost, measured by the number of messages required for completing the single aggregation round. Fig. 11 reports the communication cost as a function of $k = k'$ for the Base Scheme, as well as the maximum cost for the Enhanced Scheme. The reported values refer to $N = 500$, $N_{node} = 50$ (so that $N_{ring} = (N/N_{node})\rceil = 10$) and $z = 20$, that was selected to verify condition (25). Note that, within the examined range of $k = k'$ values, the highest value of the maximum cost of the Enhanced Scheme is about 0.8 times the lowest cost value of the Base Scheme. For $k = k^{'}$, it is therefore possible to conclude that the Enhanced Scheme outperforms the Base Scheme in terms of privacy degree and communication cost; yet, it exhibits a higher failure probability. Nevertheless, for the Enhanced Scheme, the latter figure of merit can be improved suitably picking z and $k^{'}$.

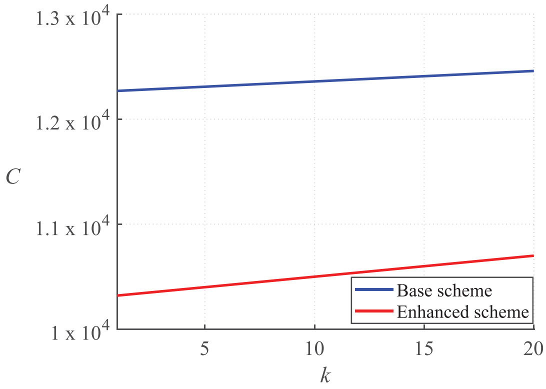


Fig. 11. Communication cost of the two schemes as a function of $k = k'$ when $p = 0.05$.

10 Conclusion

This paper has explored the domain of large scale, privacy-preserving data mining. It has explicitly taken into consideration the possibility that during the mining process, data become inaccessible due to intermittent network connectivity or sudden user departures. A distributed multi-ring architecture has been proposed, where a base scheme and an enhanced secret sharing scheme are put forth. A tight approximation for the privacy degree that the enhanced scheme warrants to users has been established by analysis. Furthermore, the impact of data unavailability on the performance of the two proposed protocols has been theoretically quantified through a newly introduced figure of merit, the failure probability. Their communication cost and computational complexity have also been assessed. The framework has been applied to some specific use cases, using real data traces, including a smart metering scenario based on the emerging LoRa technology. The presented results reveal that the performance figures of both aggregation schemes are attractive, demonstrate that MPC is feasible, and pave the way to new large-scale distributed implementations.

Acknowledgments

The authors would like to thank Cettina Barcellona for her initial work on the Matlab implementation and Fabrizio Giuliano for data analysis of the LoRaWAN network.

References

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016, and repealing Directive 95/46/EC (General Data Protection Regulation). Accessed: Jan. 12, 2021. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj

[2] P. Bogetoft *et al.*, "Secure multiparty computation goes live," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2009, pp. 325–343.

[3] D. Bogdanov, M. Joemts, S. Siim, and M. Vaht, "How the Estonian tax and customs board evaluated a tax fraud detection system based on secure multi-party computation," in *Proc. Int. Conf. Financial Cryptography Data Secur.*, 2015, pp. 227–234.

[4] E. Soria Vazquez, "Towards secure multi-party computation on the internet: Few rounds and many parties," Ph.D. thesis, Dept. Comput. Sci., Univ. Bristol, Bristol, U.K., Feb. 2019.

[5] M. M. D. Burkhart, M. Strasser, and X. Dimitropoulos, "SEPIA: Privacy-preserving aggregation of multi-domain network events and statistics," in *Proc. 19th USENIX Secur. Symp.*, 2010, Art. no. 15.

[6] Y. Duan, J. Canny, and J. Zhan, "P4P: Practical large-scale privacy-preserving distributed computation robust against malicious users," in *Proc. 19th USENIX Secur. Symp.*, 2010, Art. no. 14.

[7] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis, "State-of-the-art in privacy preserving data mining," *SIGMOD Rec.*, vol. 33, no. 1, pp. 50–57, Mar. 2004.

[8] P. Kairouz *et al.*, "Advances and open problems in federated learning," Accessed: Oct. 2020, Dec. 10, 2019. [Online]. Available: https://arxiv.org/abs/1912.04977

[9] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 439–450.

[10] A. C. C. Yao, "How to generate and exchange secrets," in *Proc. 27th Annu. Symp. Found. Comput. Sci.*, 1986, pp. 162–167.

[11] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game: A completeness theorem for protocols with honest majority," in *Proc. 19th ACM Symp. Theory Comput.*, 1987, pp. 218–229.

[12] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proc. 20th ACM Symp. Theory Computing.*, 1988, pp. 1–10.

[13] R. Cramer, I. Damgard, and U. Maurer, "General secure multi-party computations from any linear secret sharing scheme," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2000, pp. 316–334.

[14] A. Ben-David, N. Nisanm, and B. Pinkas, "FairplayMP: A system for secure multi-party computation," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, 2008, pp. 257–266.

[15] I. Damgard, M. Geisler, M. Kroigaard, and J. Nielsen, "Asynchronous multiparty computation: Theory and implementation," in *Proc. 12th Int. Conf. Theory Pract. Public Key Cryptography*, 2006, pp. 160–179.

[16] S. Bogdanov, D. Laur, and J. Willemson, "Sharemind: A framework for fast privacy-preserving computations," in *Proc. 13th Eur. Symp. Res. Comput. Secur.*, 2008, vol. 5283, pp. 192–206.

[17] V. Attasena, J. Darmont, and N. Harbi, "Secret sharing for cloud data security," *Int. J. Very Large Databases*, vol. 26, no. 5, pp. 657–681, 2017.

[18] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk, "Privacy-preserving user clustering in a social network," in *Proc. 1st IEEE Int. Workshop Inf. Forensics Secur.*. 2009, pp. 96–100.

[19] J. Vaidya and C. Clifton, "Privacy-preserving K-means clustering over vertically partitioned data, in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2003, pp. 206–215.

[20] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[21] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *ACM SIGKDD Explorations Newslett.*, vol. 4, pp. 28–24, 2002.

[22] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[23] G. Blakley, "Safeguarding cryptographic keys," in *Proc. Int. Workshop Manag. Requirements Knowl.*, 1979, pp. 313–318.

[24] V. Attasena, N. Harbi, and J. Darmont, "A novel multi-secret sharing approach for secure data warehousing and online analysis processing in the cloud," *Int. J. Data Warehousing Mining*, vol. 11, no. 2, pp. 21–42, 2015.

[25] F. Randazzo, D. Croce, I. Tinnirello, C. Barcellona, and M. L. Merani, "Experimental evaluation of privacy-preserving aggregation schemes on PlanetLab," in *Proc. Int. Wireless Commun. Mobile Comput. Conf.*, 2015, pp. 379–384.

[26] D. Croce, F. Giuliano, I. Tinnirello, and L. Giarré, "Privacy-preserving overgrid: Secure data collection for the smart grid," *Sensors*, vol. 20, 2020, Art. no. 2249.

[27] J. Dun, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, pp. 32–57, 1973.

[28] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[29] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method*, Berlin, Germany: Springer, 2004. [Online]. Available: https://doi.org/10.1007/978-1-4757-4321-0

[30] A technical overview of LoRa® and LoRaWAN®, 2015. Accessed: Jan. 12, 2021. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf

Maria Luisa Merani (Senior Member, IEEE) is currently associate professor at the Department of Engineering, University of Modena and Reggio Emilia, Italy. Her research interest includes the area of wireless networking. She served as technical program co-chair for the 2nd IEEE International Symposium on Wireless Communication Systems 2005 (ISWCS'05) and for IEEE Globecom 2007 and 2009, in 2009 Cambridge University Press, she published her textbook "Hands on networking: from theory to practice" and in the same year she was one of the authors of the Springer book "Handbook of P2P networking." In 2010, she was the general chair of the IEEE International Symposium on Wireless Pervasive Computing and the guest editor of a special issue of the Springer journal "P2P networking and applications" focused on peer-to-peer for video delivery. She is served as an editor of the *IEEE Transactions on Wireless Communications*.

Daniele Croce received the double MSc degree in networking engineering from the Politecnico di Torino and EURECOM Institute, Sophia Antipolis, France, in 2006, and the PhD degree jointly from Politecnico di Torino, Turin, Italy, and Université de Nice-Sophia Antipolis, UNSA, Nice, France, in 2010. Currently, he is an assistant professor at the University of Palermo, Italy. He has worked in several research projects, on Wireless networks, Internet of Things, underwater communications and smart cities. He also worked on assistive technologies for visually impaired people founding the start-up company In.sight, spin-off of Palermo University, Italy.

Ilenia Tinnirello received the PhD degree in telecommunications engineering from the University of Palermo, Italy, in 2004. She is currently an associate professor at the University of Palermo, Italy. She has also been visiting researcher at the Seoul National University, Korea, in 2004, and Nanyang Technological University, Singapore, in 2006. Her research interest include wireless networks, and in particular on protocols and architectures for emerging reconfigurable wireless networks. She has been involved in several European research projects, among which the FP7 FLAVIA project, with the role of technical coordinator, and the H2020 WiSHFUL and Flex5Gware projects.