A New Privacy-Preserving Solution for Clustering Massively Distributed Personal Times-Series

Tristan Allard, Georges Hébrail, Florent Masseglia, Esther Pacitti

A New Privacy-Preserving Solution for Clustering Massively Distributed Personal Times-Series

Tristan Allard[‡] ✉ Georges Hébrail[*] Florent Masseglia[#] Esther Pacitti[+]
[‡] *IRISA & Univ. Rennes 1, 263 av. Général de Gaulle, 35042 Rennes cedex, France*
[*] *EDF R&D, 1 av. Général de Gaulle, BP 408, 92141 Clamart cedex, France*
[#,+] *Inria & Lirmm, Univ. Montpellier, Campus St Priest, 860 rue de St Priest, 34095 Montpellier cedex 5, France*
first.last@{[‡]irisa.fr, [*]edf.fr, [#]inria.fr, [+]lirmm.fr}

Abstract*—New personal data fields are currently emerging due to the proliferation of on-body/at-home sensors connected to personal devices. However, strong privacy concerns prevent individuals to benefit from large-scale analytics that could be performed on this fine-grain highly sensitive wealth of data. We propose a demonstration of Chiaroscuro, a complete solution for clustering massively-distributed sensitive personal data while guaranteeing their privacy. The demonstration scenario highlights the affordability of the *privacy vs. quality* and *privacy vs. performance* tradeoffs by dissecting the inner working of Chiaroscuro - launched over energy consumption times-series -, by exposing the results obtained by the individuals participating in the clustering process, and by illustrating possible uses. *Keywords—differential privacy, gossip, k-means, time-series;

I. Introduction

The ongoing wave of personal sensors is leading to a massive generation of personal time-series related to individuals' health (*e.g.,* through smart wristbands or smart bed scales) or to inner-home activities (*e.g.,* through electrical smart plugs or smart meters). These time-series are typically sent to their owner's personal devices, such as his smartphone, his tablet, or his laptop.

The availability of this wealth of data massively distributed over personal devices is an unprecedented opportunity for individuals to learn valuable knowledge. Cluster analysis, also called *clustering*, aims at forming groups of data (or *clusters*) such that similar data appear in the same group and dissimilar data appear in different groups. Clustering is widely used in various application domains, *e.g.*, medicine, genetics, marketing, energy, or social sciences. We believe that clustering personal time-series can directly benefit individuals. Assume for example that time-series contain variations of weight. An individual suffering from obesity could benefit from the clustering results by identifying *interesting groups* of weight time-series - *e.g.,* groups within which weight time-series are similar to his own time-series on a subsequence but exhibit finally a steady decrease – in order to further discover and investigate the associated diets. Similar uses can benefit individuals in many situations (*e.g.,* clustering electrical consumption time-series for identifying the low-consumption groups and discover the equipments that could be replaced to improve the electrical consumption).

A naive approach for clustering these masses of personal time-series could simply consist in copying them from the set of personal devices to a central server which could then apply the usual cluster analysis techniques. However, fine grain personal time-series are sensitive data. A personal time-series indeed conveys a large quantity of information about the individual(s) it concerns, and may lead to arbitrary disclosures [1]. Although entrusting a single entity with large amounts of data may be practical, recurring large scale data breaches show that this is actually hazardous in the real world (see, *e.g.*, http://datalossdb.org). Moreover, the frequent occurrence of data breaches fuels the growing concern of individuals about the systematic centralization of their personal data. As a result, the centralized approach not only introduces an additional risk to data privacy but also an additional obstacle to large-scale personal data analysis.

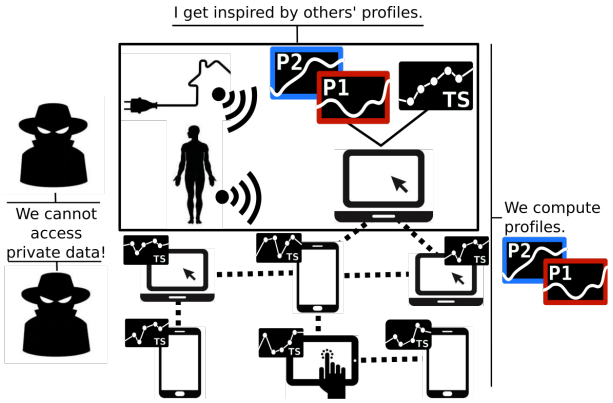

Fig. 1. Collaborative Clustering of Massive Personal Time-Series with Privacy Guarantees

In this demonstration, we present Chiaroscuro [2], a complete solution for clustering personal time-series that are massively distributed on a large population of honest-but-curious personal devices without jeopardizing their privacy. The challenge addressed by Chiaroscuro arises from the conjunction of the massive distribution of the execution over possibly faulty computing nodes on one side, with the strong privacy guarantees that personal time-series need on the other side. To the best of our knowledge, related works fail in addressing one side of the challenge or the other. This is essentially due either to the use of non fault-tolerant cryptographic techniques or to the presence of out-of-control data disclosures (see [2] for a deeper discussion of related works).

In order to address together the two sides of the challenge, Chiaroscuro relaxes the traditional security desideratum by allowing *differentially private* intermediate results to be dis-

closed during the execution, any other information remaining encrypted. Demonstrating Chiaroscuro essentially aims at showing that using encryption together with differentially private perturbation can be a key enabler in the design of massively distributed privacy-preserving analytical algorithms. The demonstration highlights the main arguments that support this claim: (1) a high level of privacy can be reached (*i.e.,* probabilistic variant of differential privacy), (2) a high level of quality can be reached (similar to the quality of centralized clustering results), and (3) costs remain affordable given the resources of today's personal devices.

II. CHIAROSCURO

A. Preliminaries In a Nutshell

Clustering. k-means [3] aims at proposing k clusters that optimize an objective quality function[1] (*e.g.,* the intra-cluster inertia which measures the homogeneity of the set of time-series within clusters). In general, a cluster can be described by extension by enumerating its content, or by intension with *e.g.,* the average of its time-series, called *centroid* or *profile* below. k-means is an iterative algorithm that progressively "fits" to data a set of k proposed centroids. It starts the first iteration by chosing the k initial centroids, *e.g.,* at random, and terminates when a termination criterion is satisfied (*e.g.,* the centroids converge, or a given number of iterations is reached). Each iteration follows three steps:

1) *Assignment step:* For each time-series, get its closest centroid and assign it to the corresponding cluster;
2) *Computation step:* For each cluster, compute its average time-series, which is its candidate centroid for the next iteration if any (also called its *mean* below);
3) *Convergence step:* If the distance between the set of centroids and the set of means is greater than a given threshold, then another iteration starts, taking as input centroids the set of means (go to Step 1), otherwise return the set of means;

Distribution. Gossip aggregation algorithms are lightweight fully decentralized approximate algorithms executed within large sets of participants. They simply consist of periodical point-to-point exchanges between participants, called *gossip exchanges* in the rest of the document. Each participant holds its own approximation of the global aggregate and updates it, at each exchange, with the one of the communicating participant. The approximation error depends on the number of gossip exchanges per participant and is guaranteed to converge to zero exponentially fast [4].

Privacy. Chiaroscuro's privacy guarantees rely on two building blocks: an additively-homomorphic encryption scheme on one side and the differentially-private Laplace perturbation scheme on the other side. First, Chiaroscuro is independent of any specific encryption scheme provided that (1) it satisfies the strong semantic-security level, (2) it is additively-homomorphic, and (3) the decryption is performed collaboratively by any subset of participants provided it is sufficiently large. The Damgard-Jurik encryption scheme [5] that we use in our implementation is an instance of such schemes. Second, differential privacy is the current *de facto* standard for disclosing to untrusted parties aggregated data, such as a sum of a set of time-series for example, while guaranteeing a strong privacy level to individuals having participated in the aggregate. The Laplace perturbation mechanism satisfies the ϵ-differential privacy model [6] by adding random noise to the aggregate to be disclosed. The noise is sampled in a Laplace distribution parameterized according to ϵ and to the aggregate disclosed. A Laplace random variable can be computed by summing up n terms independently generated based on the gamma distribution, n being fixed beforehand. These terms are called *noise-shares*. When several aggregates related to the same individuals are perturbed and disclosed, differential privacy is still satisfied (self-composition property) and the global privacy level, seen as a *privacy budget*, must be divided among the perturbations because it is is the addition of all the privacy levels set for perturbing the various aggregates. Note that because of the inherent approximations of gossip algorithms, Chiaroscuro satisfies a probabilistic variant of ϵ-differential privacy.

[1] Characterizing precisely the algorithms that Chiaroscuro can support is future work.

B. Overview of Chiaroscuro

Diptych Data Structure. Parallelizability is a crucial property in this massively distributed context, making a clustering algorithm such as k-means especially relevant. k-means is essentially based on a twofold data structure made of the centroids on one side and of the means on the other side. The centroids are used during Steps 1 and 3 (*i.e.,* assignment and convergence steps) for performing distance comparisons, and the means result from the algebraic computations performed in Step 2 (*i.e.,* computation step). Chiaroscuro brings together additively-homomorphic encryption and differentially-private perturbation in order to shield the two sides of this data structure while enabling the gossip-based distribution of the execution sequence. In Chiaroscuro, Step 2 is performed over additively-homomorphic encrypted means, which support the algebraic operations required by this step, while Step 1 and Step 3, harder to perform over encrypted data, are performed over cleartext centroids perturbed to satisfy differential privacy. The resulting data structure consists thus of the perturbed centroids on one side and of the encrypted means on the other side; it is called *Diptych* and is key to the execution sequence.

Execution Sequence. Chiaroscuro's execution sequence revisits the assignment, computation, and convergence steps by distributing and articulating them together based on the diptych data structure. It is iterative, identical for all participants, and proceeds without any global synchronization (the late participants simply synchronize on the latest iteration during their gossip exchanges). It consists in the following:

1) *Assignment step (local):* pull in a set of perturbed centroids, assign the local time-series to the closest perturbed centroid and initialize (1) the corresponding encrypted mean with the encryption of the local time-series and (2) all the other means with the encryptions of zero-valued time-series;
2) *Computation step (distributed):* Compute the set of perturbed means:
    a) Gossip computation of the encrypted means;
    b) Gossip computation of the encrypted noises;

c) Local addition of the encrypted noises to the encrypted means;
d) Collaborative decryption of the perturbed encrypted means;

3) *Convergence step (local):* if the distance between the perturbed centroids and the perturbed means is greater than a given threshold, then another iteration starts, taking as input the set of perturbed means[2] (go to Step 1), otherwise return the set of perturbed means;

Steps 1 and 3 are performed locally on cleartext data, they do not present strong challenges. It is the computation step that concentrates the hardest points of the execution sequence. Chiaroscuro solves it by proposing a gossip sum algorithm working on additively-homomorphic encrypted data and by using it as a building block for computing the encrypted means and the Laplace noises. The collaborative decryption is performed by getting from a sufficient number of distinct participants their partial decryptions.

Quality-Enhancing Heuristics. Chiaroscuro also embeds quality-enhancing heuristics for reducing the impact of the perturbation on the quality of the clustering. They act on (1) the quality of the sequence of centroids through smart privacy budget distribution strategies and on (2) the quality of each centroid by smoothing the perturbed means.

III. Demonstration

This demonstration allows to observe thoroughly the inner working of Chiaroscuro, and to illustrate the use of the resulting centroids (or profiles) by an individual. We present below the software platform that supports this objective, the mutable and fixed initial parameters, and the demonstration scenario.

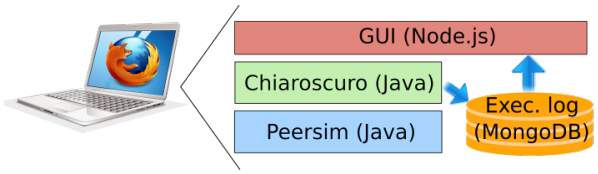

Fig. 2. Demonstration Platform: The demonstration runs on a single laptop (no connection to the network needed). It executes both Chiaroscuro (within Peersim) and the GUI system. The execution log is stored in a local MongoDB database and displayed by the GUI through a web browser.

A. Platform

Figure 2 depicts the demonstration platform. Chiaroscuro's engine is implemented within the well-known distributed computing simulator *Peersim* [7]. Chiaroscuro is written in Java and implements Peersim's `nextCycle` method by the core of its execution sequence. This method is the same for all participants and is the entry point for Peersim each time it calls a participant. Our graphical user interface is based on *Node.js*. The client-side part allows the client to choose the values for a clearly-defined set of initial parameters (see below the parameters allowed to change, called *mutable parameters*), to visualize a variety of information about the execution of Chiaroscuro parameterized with the chosen values (see below the visualization details), and to interact with the clustering result. This information is captured in the execution logs, stored in a MongoDB instance, and is then interpreted by the GUI system. Both Chiaroscuro's engine and the GUI system run locally on the same machine.

B. Parameters

In this demonstration, we allow some initial parameters to be set up by the audience, the others being fixed to default values. The mutable parameters are carefully selected so that a change in their values brings insights into the quality reached by Chiaroscuro (*e.g.,* the differential privacy level, the quality-enhancing heuristics enabled, the use-case - electrical consumption time-series or tumor-size growth) and into its costs (*e.g.,* the number of participants required for decryption). The fixed parameters are related to the k-means algorithm (*e.g.,* number of initial centroids, convergence threshold), to the encryption scheme (*e.g.,* size of the encryption key), and to the gossip algorithm (*e.g.,* number of participants, number of exchanges per participant). In order to keep reasonnable the execution time of Chiaroscuro, we simulate a tiny population (*e.g.,* on the order of 10^3 participants rather than 10^6 as targeted by Chiaroscuro) and we disable the homomorphic operations (a single machine can hardly cope with the encryption load of a thousand participants). We stress that this has no consequence on the objectives of the demonstration: (1) the distributed algorithms are not changed whether homomorphic operations are enabled or not, (2) the performance overhead that would be due to homomorphic operations and to a larger population size are clearly displayed in the GUI based on actual average measures performed beforehand (*e.g.,* of encryption/decryption/addition times), (3) the approximation error of gossip algorithms is kept similar to a context with a larger population by decreasing the number of messages per participant, and (4) the impact of the perturbation is also kept similar by scaling the differential privacy level to obtain the same "noise magnitude / population size" ratio.

Chiaroscuro is demonstrated over a real dataset and a synthetic one, each related to a targeted application domain. The CER dataset [8] contains the electricity consumption time-series of thousands of real Irish homes and businesses. The NUMED dataset contains time-series representing the tumor growth of cancer suffering patients synthetically generated based on mathematical models [9]. The demonstration uses a subset of each dataset for keeping reasonable the execution times.

C. Scenario

The demonstration scenario showcases Chiaroscuro by allowing the audience to follow thoroughly the evolution over a complete run of the perturbed centroids obtained by participants, of their quality (compared to a centralized k-means), and of the network and encryption costs. We also put a special emphasis on the use of the result by *Bob*, a pre-defined fictional participant that wants to benefit from the resulting centroids by participating with his time-series but without giving away his privacy. Figure 3 illustrates the typical progress of the demonstration. A strong focus has been put on the pedagogical aspects of the GUI since it includes the supports for giving the explainations necessary for understanding intuitively the

[2]Chiaroscuro supports the addition of other termination criteria for coping with the impact of the differentially-private perturbation on the convergence of centroids (*e.g.,* monitoring centroids quality).

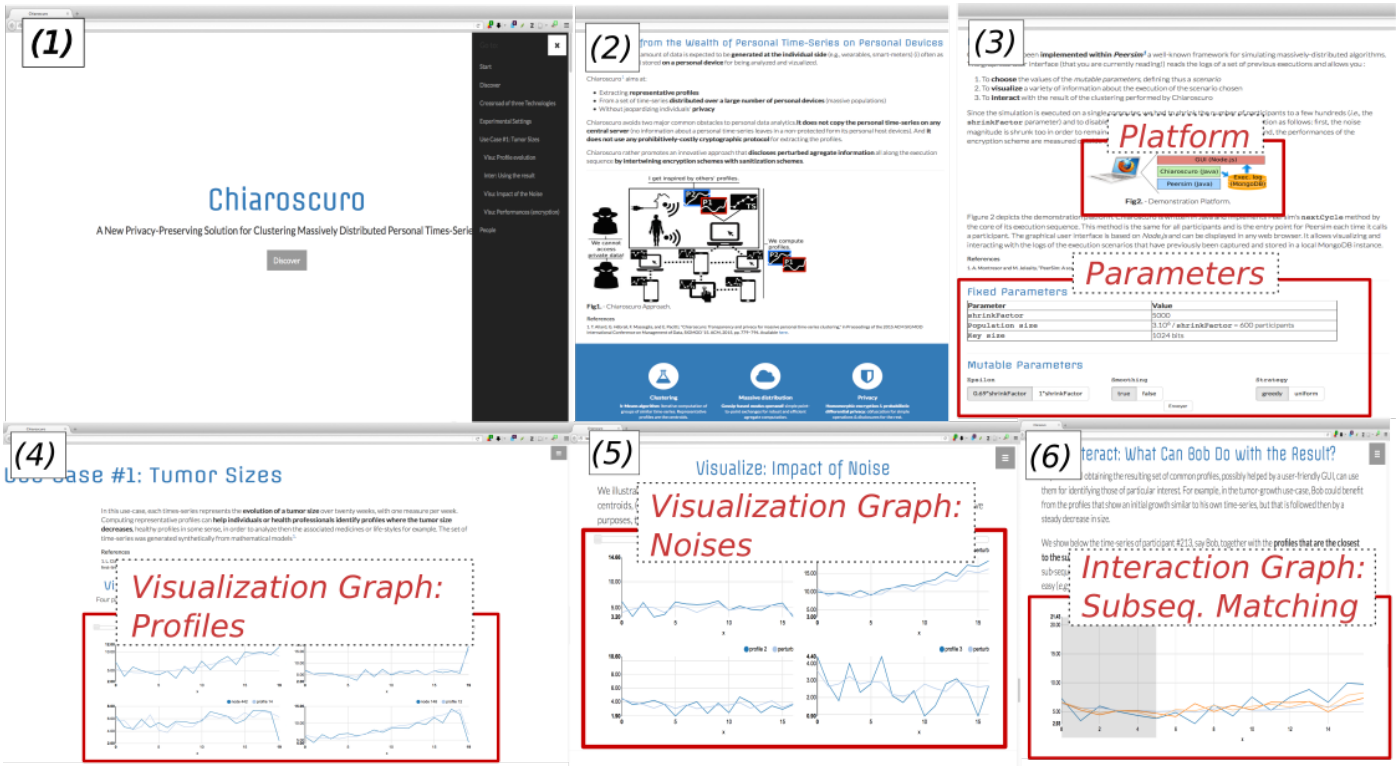

Fig. 3. A Glimpse over the Graphical User Interface (annotated in red): Sequence of screenshots showing from the top-left to the bottom-right: (1) the welcome screen, (2) the intuitions of the approach, (3) the experimental platform, a few fixed parameters (in the table), and a few mutable parameters (the buttons), (4) for the first use-case (tumor-growth time-series over twenty weeks), the graphs showing for a random subset of four participants the evolution of their closest centroid along the iterations (a slide bar allows navigating along the iterations), (5) an illustration of the impact of the noise on four random centroids along the iterations (a slide bar allows navigating along the iterations), and (6) an illustration of the use of the clustering results by an individual (finding the closest profiles given a sub-sequence of his own time-series).

technical aspects of Chiaroscuro. The GUI is divided into a sequence of screens. After the introductory screens explaining Chiaroscuro and the demonstration platform, and allowing to set the mutable parameters, the GUI displays a set of graphs. The first type of graphs is passive, dedicated to vizualize, *e.g.*, the evolutions of the centroids, the noise values, and the quality and cost measures, based on slide bars over the iterations. The second type of graph is interactive, it illustrates the use of the resulting centroids by Bob, for example by displaying Bob's time-series and allowing the audience to interact with it by selecting a sub-sequence and finding the centroids the closest to the sub-sequence chosen.

IV. Acknowledgment

The authors warmly thank C. Maupetit and M. Simonin from Inria Rennes for having designed and developed the GUI system.

References

[1] M. Newborough and P. Augood, "Demand-side management opportunities for the uk domestic sector," *Gen., Trans. and Dist.*, vol. 146, no. 3, pp. 283–293, 1999.

[2] T. Allard, G. Hébrail, F. Masseglia, and E. Pacitti, "Chiaroscuro: Transparency and privacy for massive personal time-series clustering," in *SIGMOD*, 2015.

[3] S. Lloyd, "Least squares quantization in PCM," *Trans. Inf. Theor.*, vol. 28, no. 2, pp. 129–137, 1982.

[4] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *FOCS*, 2003, pp. 482–491.

[5] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of paillier's probabilistic public-key system," in *PKC*, 2001.

[6] C. Dwork, "Differential privacy," in *ICALP*, 2006, pp. 1–12.

[7] A. Montresor and M. Jelasity, "PeerSim: A scalable P2P simulator," in *P2P*, 2009, pp. 99–100.

[8] ISSDA, *The Commission for Energy Regulation, Electricity Customer Behaviour Trial*, 2012, http://www.ucd.ie/issda.

[9] L. Claret, M. Gupta, K. Han, A. Joshi, N. Sarapa, J. He, B. Powell, and R. Bruno, "Evaluation of tumor-size response metrics to predict overall survival in western and chinese patients with first-line metastatic colorectal cancer." *J. Clin. Onc.*, vol. 31, no. 17, pp. 2110–2114, 2013.