

Enabling Privacy-Preserving Data Aggregation in Networks of Personal Data Management Systems (Extended Abstract)

Julien Mirval^{1,2,3}, Iulian Sandu-Popa^{3,2}, Luc Bouganim^{2,3} and Paul Tran-van¹

¹Cozy Cloud, "Le Surena" face au 5 Quai Marcel Dassault, 92150 Suresnes, France

²Inria de Saclay, Campus de l'École polytechnique, 1 rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France

³Université de Versailles Saint-Quentin, 45 Avenue des Etats-Unis, 78000 Versailles, France

Abstract

The development and adoption of Personal Data Management Systems (PDMS) have been fueled by technical means, privacy regulations, and smart disclosure initiatives. A PDMS makes it easier for users to collect, process, and share personal data. However, functionalities based on collective computations within a network of PDMSs are still lacking at least in commercial products. This demonstration bridges this gap by leveraging the open-source Cozy Cloud product and recent research results in the area of privacy-preserving decentralized data aggregation. Our demonstration scenario highlights both the utility aspect of collective computations and the main features of the aggregation protocol.

Keywords

Personal data management systems, Secure aggregation, Peer-to-peer, Federated learning


1. Introduction

New privacy-protection regulations (e.g., GDPR) and smart disclosure initiatives in the last decade have boosted the development and adoption of Personal Data Management Systems (PDMSs) [1]. A PDMS (e.g., Cozy [2], Nextcloud, Solid) is a data platform that allows users to easily collect, store, and manage data into a single place, directly generated by the user's devices (e.g., quantified-self data, smart home data, photos) and data resulting from the user's interactions (e.g., social interaction data, health, bank, telecom). Users can then leverage the power of their PDMS to benefit from their personal data for their own good and for the benefit of the community [3]. The ambition of the existing PDMSs is to offer functionalities covering all the major steps in the data life-cycle [1]: (i) data backup and storage; (ii) data collection via connectors to the typical online services holding user data (e.g., bank, telecom, shopping, social networks, email); (iii) data sharing between user's devices or between different users' PDMSs; and (iv) advanced personal computations allowing a user to cross her data from different data silos (e.g., health records and physical activity data).

Proceedings of the Demonstration Track at International Conference on Cooperative Information Systems 2023, CoopIS 2023, Groningen, The Netherlands, October 30 - November 3, 2023

✉ julien.mirval@cozycloud.cc (J. Mirval); iulian.sandu-popa@uvsq.fr (I. Sandu-Popa); luc.bouganim@inria.fr (L. Bouganim); paul@cozycloud.cc (P. Tran-van)

ORCID 0009-0004-4580-5651 (J. Mirval); 0000-0002-9937-4242 (I. Sandu-Popa); 0000-0002-2273-9987 (L. Bouganim)

 © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

However, the PDMS paradigm leads to a shift in the personal data ecosystem since data becomes massively distributed, on the user side. To unlock innovative usages, individuals can leverage their PDMSs by forming large communities of users sharing their data. This allows, for example, to compute statistics for epidemiological studies or to train a Machine Learning (ML) model for recommendation or classification systems. These usages however introduce new security and performance issues, as evidenced by the large body of recent works in this area [4]. To enable such new usages in the PDMS context, we need new solutions adapted to its specificity. These protocols need to protect user privacy and adapt to varying selectivity (i.e., the consent of relevant participants). Ideally, the proposed protocol should provide an accurate result that takes advantage of the high-quality data available in PDMSs. Efficiency (i.e., protocol latency and total load of the system) is of prime importance given the potentially limited communication speed or computation power of PDMSs. Finally, given the scale of such decentralized aggregation, protocols must also be robust to node dropouts.

Ensuring these properties altogether is challenging which might explain the lack of functionalities implementing collective computations by the existing commercial PDMS solutions [1], the privacy-preserving data aggregation in a network of PDMSs being mostly the focus of research works and prototypes. This demonstration is a first step towards bringing closer commercial PDMS solutions and recent academic results for privacy-preserving data aggregation in a network of PDMSs. Specifically, the main contribution of this demonstration is to integrate into an existing, classical, open-source PDMS solution, i.e., Cozy (see Section 2), a privacy-preserving, scalable, and adaptive aggregation protocol (see Section 3) leveraging our recent research results [5, 6]. Thus, our demonstration shows that privacy-preserving collective computations can be enabled in the PDMS context allowing users to benefit, through collective data sharing, from the diverse, abundant and high-quality data stored in their PDMSs (e.g., by collectively training an ML classifier). In addition to a walk around the classical functionalities of the Cozy Cloud PDMS, our demonstration scenario (see Section 4) highlights both the utility aspect of collective computations and the main features of the aggregation protocol.

2. The Cozy Platform

Cozy Cloud [2] has been developing a PDMS platform for over a decade with the objective of providing a "smart digital home" that combines the comfort of having all your data stored and processed in a single place, with the virtues of a reproducible open-source environment. Below, we describe the main features of Cozy.

Data collection. Cozy enables effortless automatic data collection through connectors which fetch or scrap data from external service providers. Connectors are open-source and easy for independent developers to create (e.g., there are more than 150 existing connectors, most of them developed from the community).

Data sharing. Data can be shared selectively with other users thanks to fine-grained control over access groups and permissions. It can also be shared across the user's devices to enable accessing data locally, even during periods of no or low connectivity.

Cross-domain local computations. Users can install apps and services that use their data locally. Upon installation, users are presented with a detailed summary of the data the app

requires and the related purposes so that the users can give their informed consent. These apps enable cross-domain computations to benefit from the variety of user data and provide useful services and interesting analytics. An example is to compute your carbon footprint based on mobility traces as well as home energy consumption data.

Collective computations. In addition, these apps and services can also be used to coordinate calculations between users. This can be used, for example, to anonymously compare previously calculated carbon footprints with those of users around you. However, distributed computations introduce a whole range of new security and privacy risks that are not, or poorly, addressed by current PDMS solutions.

3. Privacy-Preserving Decentralized Data Aggregation

This section summarizes the main design principles proposed in [5] to fulfill the privacy, accuracy, and adaptability properties. Before going into the aggregation protocol details, we briefly introduce the considered computation and threat models.

Computation model. This demo focuses on aggregation primitives which are essential to compute basic statistics on user data and are also a fundamental building block for ML algorithms. A model computation can be triggered by any node, i.e., *querier*, in a PDMS network. The querier broadcasts the computation and each node consents or not to contribute, and in the positive case is called *contributor*. Each node (contributor or not) may be a data processor and is then called *aggregator*. Each contributor trains the model locally for several epochs as described in [7] and sends it to parent aggregators. Achieving a scalable aggregation process requires multiple aggregators, naturally arranged in a tree structure (see Fig. 1.a) wherein the intermediary nodes are aggregators and the leaves are contributors. The querier obtains the result from the tree root.

Threat model. As in the majority of secure aggregation (SA) works [4], we consider the classical *honest-but-curious* threat model, i.e., an attacker can access, but cannot alter, the data manipulated by the attacked nodes (called *leaking nodes*). A PDMS can hold the entire digital life of its owner and thus needs to be highly protected against privacy threats as indicated by recent works [8]. However, we consider that some PDMS owners have succeeded in tampering with their PDMS since no security measure is unbreakable. Since attackers may collude and thus, de facto, control more than one PDMS, the worst-case attack is represented by the maximum number of colluding nodes controlled by a single “attacker”, i.e., C leaking nodes.

Privacy and accuracy: We use a secret sharing scheme without threshold for data confidentiality. Each contributor splits its private value into s shares, which leads to building s separate (parallel) aggregation trees with exactly the same structure. This makes it impossible to reconstruct the secret unless someone collects all s shares. This precludes inferences from an attacker on any of the intermediate results (see Fig. 1.b). Each i^{th} share has the value $x_i = x/s + \epsilon_i$ such that $\sum_{i=1}^n \epsilon_i = 0$, where x is the private value. Thus, shares from different contributors are aggregated separately and if no share is missing (reliability is discussed in [6]), the final result equals the exact sum of all private values, which is computed by the querier. Hence, our protocol provides, by construction, accurate results. The number of shares, s , is computed such that the probability to obtain s shares for an attacker, controlling C nodes, is inferior to α , a

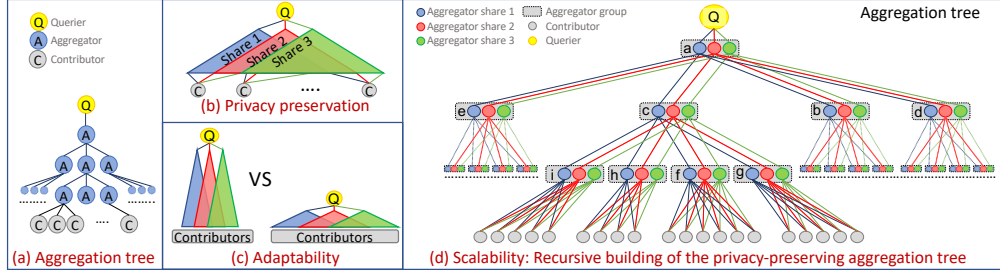


Figure 1: Privacy-preserving, adaptable and scalable data aggregation using multiple aggregation trees

security threshold (e.g., $\alpha = 10^{-6}$). Consequently, considering a uniform node distribution [5], the probability that an attacker controls an entire group is given by $(C/N)^s < \alpha$. Then s is minimal when $s = \lceil \log(\alpha) / \log(\frac{C}{N}) \rceil$.

Adaptability: The number of aggregators and their arrangement (i.e., the tree fan-out and its height) is tuned as a function of the number of contributors, the communication costs and the processing costs as discussed in [5]. This allows the protocol to always offer near-optimal performance (i.e., aggregation latency) and achieve adaptability w.r.t. the computation selectivity and PDMSs characteristics. Furthermore, our protocol can be configured to offer the desired trade-off between the latency and the total cost of the aggregation, which are conflicting objectives : At one extreme, a binary aggregation tree maximally distributes the load but increases the latency, at the other extreme, all contributors concentrate the load on a single aggregator group (see Fig. 1.c).

For the sake of brevity, we omit the details on the scalability property (see Fig. 1.d and details in [5]) and also the thorny issue of reliability (see [6]).

4. Demonstration

Our scenario will give a quick tour of Cozy’s functionalities and will mainly show the benefits and feasibility of collective computations for ML applications in the PDMS context. Our demonstration software is built as an app on Cozy’s platform and uses a set of connected PDMSs that store banking operations to classify. A detailed explanation of the installation and usage of the demo software can be found on a public repository¹ or in video². We eventually aim to provide this app to real end-users of Cozy.

For the purpose of the demonstration, local PDMS instances are created and populated with samples of test data (i.e., banking operations), one of them being the querier. A web interface allows to view all of the instance’s banking operations. Each PDMS instance has some operations that are already classified but it does not have enough classified data to train an accurate ML model. In fact, the locally trained model may introduce too many classification errors that could confuse the user, and considering those errors as unclassified is safer (see Fig. 2 left).

From the demonstration platform’s interface, we can trigger the computation of a collectively trained model leveraging a specific aggregation tree (see Section 3). The platform assigns

¹https://github.com/cozy/dissec_cozy/blob/master/DEMONSTRATION.md

²<https://clipchamp.com/watch/7vsaYZyPEDa>

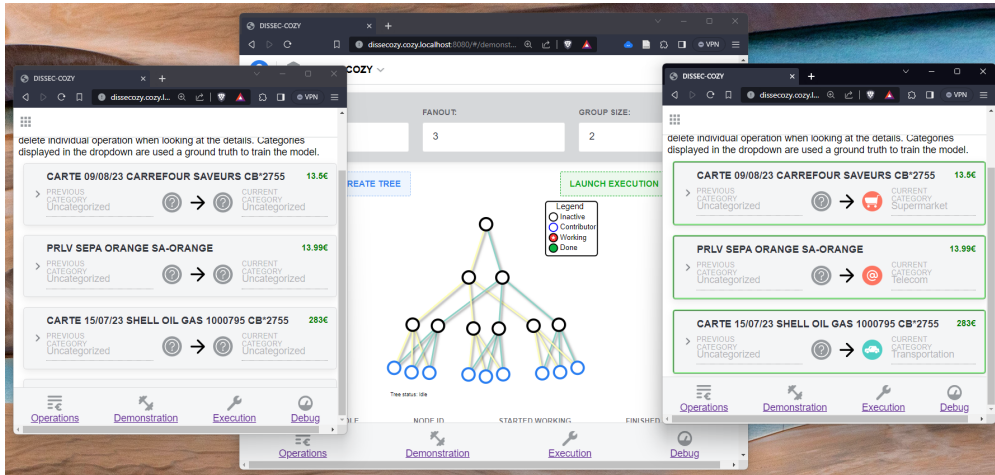


Figure 2: Demonstration platform classifying using the local (left) and distributed training (right)

instances to the tree to facilitate tuning the tree structure. The aggregation process is visible in real-time (see Fig. 2 center) as the assigned nodes start processing data and transmitting intermediate results, starting from contributors at the bottom and up to the querier at the top, which recomposes the global model.

Once the global model is available, we can rerun the classification of the banking operations. We observe that this model can effectively classify all the banking operations (see Fig. 2 right). Also, the demonstration interface allows the audience to comprehend the properties of the aggregation protocol and in particular the computation security, i.e., an attacker controlling a subset of the PDMS instances cannot gain any knowledge on the private data from other nodes.

References

- [1] N. Ancaux, P. Bonnet, L. Bouganim, B. Nguyen, et al., Personal Data Management Systems: The Security and Functionality Standpoint, Information Systems (2019).
- [2] Cozy Cloud, Cozy Cloud (see <https://cozy.io/fr/>), 2023.
- [3] E. Commission, Proposal for a Regulation on European Data Governance (Data Governance Act), COM/2020/767. [eur-lex], 25 October 2020.
- [4] M. Mansouri, M. Önen, W. B. Jaballah, M. Conti, SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning, PETS (2023).
- [5] J. Mirval, L. Bouganim, I. Sandu-Popa, Practical Fully-Decentralized Secure Aggregation for Personal Data Management Systems, in: SSDBM, 2021.
- [6] J. Mirval, L. Bouganim, I. Sandu-Popa, Federated learning on personal data management systems: Decentralized and reliable secure aggregation protocols, in: SSDBM, 2023.
- [7] B. McMahan, E. Moore, D. Ramage, S. Hampson, et al., Communication-Efficient Learning of Deep Networks from Decentralized Data, PMLR, 2017.
- [8] N. Ancaux, L. Bouganim, P. Pucheral, I. Sandu-Popa, et al., Personal Database Security and Trusted Execution Environments: A Tutorial at the Crossroads, PVLDB (2019).