Federated Learning on Personal Data Management Systems: Decentralized and Reliable Secure Aggregation Protocols

Julien Mirval, Luc Bouganim, Iulian Sandu Popa

Federated Learning on Personal Data Management Systems: Decentralized and Reliable Secure Aggregation Protocols

Julien Mirval
julien.mirval@cozycloud.cc
Cozy Cloud, Inria-Saclay
UVSQ, Université Paris-Saclay
France

Luc Bouganim
luc.bouganim@inria.fr
Inria-Saclay
UVSQ, Université Paris-Saclay
France

Iulian Sandu-Popa
iulian.sandu-popa@uvsq.fr
UVSQ, Université Paris-Saclay
Inria-Saclay
France

ABSTRACT

The development and adoption of personal data management systems (PDMS) has been fueled by legal and technical means such as smart disclosure, data portability and data altruism. By using a PDMS, individuals can effortlessly gather and share data, generated directly by their devices or as a result of their interactions with companies or institutions. In this context, federated learning appears to be a very promising technology, but it requires secure, reliable, and scalable aggregation protocols to preserve user privacy and account for potential PDMS dropouts. Despite recent significant progress in secure aggregation for federated learning, we still lack a solution suitable for the fully decentralized PDMS context. This paper proposes a family of fully decentralized protocols that are scalable and reliable with respect to dropouts. We focus in particular on the reliability property which is key in a peer-to-peer system wherein aggregators are system nodes and are subject to dropouts in the same way as contributor nodes. We show that in a decentralized setting, reliability raises a tension between the potential completeness of the result and the aggregation cost. We then propose a set of strategies that deal with dropouts and offer different trade-offs between completeness and cost. We extensively evaluate the proposed protocols and show that they cover the design space allowing to favor completeness or cost in all settings.

CCS CONCEPTS

• **Computer systems organization → Peer-to-peer architectures**.

KEYWORDS

Secure aggregation, peer-to-peer, reliability, 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 Cloud [8], Nextcloud, Solid) is a data platform that allows users to easily collect, store, and manage into a single place data 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 [6].

As a result, the PDMS paradigm leads to a shift in the personal data ecosystem since data becomes massively distributed, on the user side. It also holds the promise of unlocking innovative usages. An individual can now cross her data from different data silos, e.g., health records and physical activity data. In addition, 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 systems. In this context, it is natural to rely on a fully decentralized PDMS architecture (as opposed to central servers that raise several important issues such as cost, availability and scalability with the number of users), but this also poses new challenges.

Aggregation primitives are essential to compute basic statistics on user data and are also a fundamental building block for ML algorithms. In particular, Secure Aggregation (SA) is a central component of Federated Learning (FL), introduced in [12], as evidenced by the large body of recent work in this area [11]. However, to enable such new usages in the PDMS context, we need new solutions adapted to its specificity. First, PDMS users rely on large peer-to-peer systems for data sharing and computations [1, 5] thus requiring fully decentralized and scalable aggregation protocols, discarding data centralization on servers. Also, 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. To summarize, our goal is to design protocols that fulfill the following properties: **fully decentralized and highly scalable**, with the number of participants; **privacy-preserving**, i.e., protecting the confidentiality of the contributed user data; **accurate**, i.e., no trade-off between accuracy and privacy (e.g., like in the data anonymization or differential privacy approaches); **adaptable**, i.e., adapting to a large spectrum of computation selectivity values (reflecting the subset of contributor nodes) and system configurations (network and cryptographic latency); and **reliable**, i.e., handling node dropouts (e.g., failures, voluntary disconnections or unexpected communication delays).

Ensuring these properties altogether is challenging and to the best of our knowledge, the existing distributed Secure Aggregation (SA) protocols fail to achieve this objective. On one hand,

approaches such as local differential privacy are based on adding noise to protect privacy. This affects accuracy [3] or reliability to dropouts [15] and requires a very large number of participants to reduce the impact of noise which contradicts an adaptive node selectivity (see Section **??**). On the other hand, despite leveraging different cryptographic schemes in SA for FL [11] (e.g., encryption-based [2, 9] or secret sharing-based [4, 7, 10]), existing solutions employ a similar hybrid architecture wherein one or several highly available and powerful servers aggregate the data supplied by many user devices. Although some solutions consider the case of node dropouts, this applies to client devices and never to aggregation servers [4, 7]. In a Peer-to-Peer (P2P) PDMS system, all computations are performed by internal PDMS nodes (i.e., user devices). Hence, the data aggregators and data contributor nodes have the same constraints, i.e., limited computing power and availability. Such nodes cannot be expected to carry out heavy cryptographic operations [4] and can drop out during the computation. Fortunately, the P2P approach allows involving many nodes to perform a computation thus reducing the load on individual aggregators.

A first effort towards SA adapted to P2P systems was made in [13], where we designed a protocol that fulfill the above properties in an ideal setting, i.e., without considering the reliability issue. This work brings two major novelties. First, we focus on the reliability property, which is difficult to guarantee in a fully-decentralized setting and deserves a detailed study. Second, although our protocols apply to SA in general, we chose to study the more general case of FL, given its particular interest in the PDMS paradigm. The study of FL is also more challenging due to the potentially large size of the model, which increases the scalability problem. In our experiments, we consider model sizes from very small to very large, thus covering a wide range of use cases (including classical SA).

Our contributions are as follows. We analyze the impact of dropouts, be it contributor or aggregator nodes, on the other properties of an SA protocol designed for a P2P PDMS system. Node dropouts have a direct impact on accuracy (i.e., a single failure can make the final computation result useless) and on efficiency (i.e., it can introduce large latency). From this analysis, we derive the precise requirements of a reliable protocol and show that in a fully-decentralized context, reliability also introduces a tension between result completeness (i.e., the percentage of initial contribution in the final result, despite dropouts) and computation cost. We introduce the necessary building blocks to deal with these requirements. Then, we propose a variety of execution strategies offering different trade-offs between completeness and cost and allowing to cover a wide spectrum of dropout rates, contributor selectivity or trained model sizes. Our extensive experimental evaluation shows that the proposed strategies cover well the design space allowing to favor completeness or cost in all settings.

The full version of the paper [14] is structured as follows. We first discuss the related work w.r.t. the required properties. We then introduce the considered architecture and threat model. The next section reminds the main design principles proposed in [13] and then introduces, as a starting point, a straw-man SA protocol which efficiently computes the required aggregation assuming an ideal world (i.e., there are no node dropouts). This allows to highlight the challenges induced by reliability issues. We then present the necessary building blocks to addresses the reliability related challenges, before proposing four SA strategies that leverage those building blocks and allow for different trade-off between result completeness and aggregation cost. Finally, we extensively evaluate the proposed strategies and conclude.

ACKNOWLEDGMENTS

This work has been supported by the ANR 22-PECY-0002 IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR.

REFERENCES

[1] Nicolas Anciaux, Philippe Bonnet, Luc Bouganim, Benjamin Nguyen, et al. 2019. Personal Data Management Systems: The Security and Functionality Standpoint. *Information Systems* (2019).
[2] Johes Bater, Gregory Elliott, Craig Eggen, Satyender Goel, et al. 2017. SMCQL: Secure Query Processing for Private Data Networks. *PVLDB* (2017).
[3] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. 2018. Personalized and Private Peer-to-Peer Machine Learning. In *AIStat.*
[4] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, et al. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *ACM CCS.*
[5] Luc Bouganim, Julien Loudet, and Iulian Sandu Popa. 2023. Highly Distributed and Privacy-Preserving Queries on Personal Data Management Systems. *The VLDB Journal* (2023).
[6] EU Commission. 25 October 2020. Proposal for a Regulation on European Data Governance (Data Governance Act), COM/2020/767. [eur-lex].
[7] Henry Corrigan-Gibbs and Dan Boneh. 2017. Prio: Private, Robust, and Scalable Computation of Aggregate Statistics. In *NSDI.*
[8] Cozy Cloud. 2023. *Cozy Cloud (See https://cozy.io/fr/).*
[9] Ye Dong, Xiaojun Chen, Kaiyun Li, Dakui Wang, et al. 2021. FLOD: Oblivious Defender for Private Byzantine-Robust Federated Learning with Dishonest-Majority. In *ESORICS.*
[10] Peeyush Gupta, Yin Li, Sharad Mehrotra, Nisha Panwar, et al. 2019. Obscure: Information-Theoretic Oblivious and Verifiable Aggregation Queries. *PVLDB* (2019).
[11] Mohamad Mansouri, Melek Önen, Wafa Ben Jaballah, and Mauro Conti. 2023. SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning. *PETS* (2023).
[12] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. PMLR.
[13] Julien Mirval, Luc Bouganim, and Iulian Sandu-Popa. 2021. Practical Fully-Decentralized Secure Aggregation for Personal Data Management Systems. In *SSDBM.*
[14] Julien Mirval, Luc Bouganim, and Iulian Sandu Popa. 2023. Federated Learning on Personal Data Management Systems: Decentralized and Reliable Secure Aggregation Protocols. In *Proceedings of the 35th International Conference on Scientific and Statistical Database Management.* 1–12.
[15] Amaury Bouchra Pilet, Davide Frey, and François Taïani. 2019. Robust Privacy-Preserving Gossip Averaging. In *SSS.*