

Building cross-border federated infrastructures for secure and private AI: an overview of privacy enhancing technologies and their challenges

Ines Ortega-Fernandez¹, Jaime Loureiro Acuña¹, Alberto Pedrouzo-Ulloa².

¹Galician Research and Development Center in Advanced Telecommunications (GRADIANT) Vigo, Spain, Email: {iortega, jloureiro}@gradient.org

²atlanTTic Research Center, Universidade de Vigo, Vigo, Spain, Email: {apedrouzo}@gts.uvigo.es

Abstract

In recent years, Artificial Intelligence (AI) has seen a remarkable surge in adoption in many everyday applications, primarily fueled by Machine Learning (ML) algorithms that rely on extensive data for model training. However, privacy constraints and the decentralization of data across various repositories, often constrained by data sharing limitations, present a significant challenge. To address this issue, Federated Learning (FL) techniques have emerged with the promise of facilitating collaborative model training across disparate devices or entities while offering better data privacy guarantees. While it is true that FL enhances data privacy, security concerns still remain, including privacy attacks that compromise the confidentiality of training data like attribute inference and even data reconstruction attacks. To strengthen the baseline privacy provided by FL, it is essential to research and develop novel privacy enhancement methods for Federated Learning. Our goal is to deliver a highly scalable, armored Federated AI service platform for researchers, enabling AI-powered studies of multi-site, siloed, cross-domain, cross-border European datasets with high privacy guarantees which comply with data privacy regulations such as the General Data Protection Regulation (GDPR). This paper explores how FL can be a useful tool for implementing collaborative training of ML models with privacy guarantees, focusing on the main challenges to be addressed to achieve an armored FL framework.

A brief introduction to Federated Learning

In recent years, the use of Artificial Intelligence (AI) has significantly increased in many everyday applications and its popularity can be largely attributed to the abundance of extensive training datasets. However, datasets must be often gathered from repositories controlled by different organizations that have strict data sharing limitations.

To address this, Federated Learning (FL) was originally positioned as a major privacy-preserving innovation, apparently able to address the problem of collaborative learning by enabling jointly training of models across different entities while keeping the training data private [1]. In particular, in cross-silo Federated Learning (see Figure 1), different data owners

(each with their own data silos) collaborate to train a common ML model without sharing their data with each other. Instead, they send their local model updates to a central server, which aggregates the updates from each participant and sends the aggregation back to the different data owners [6].

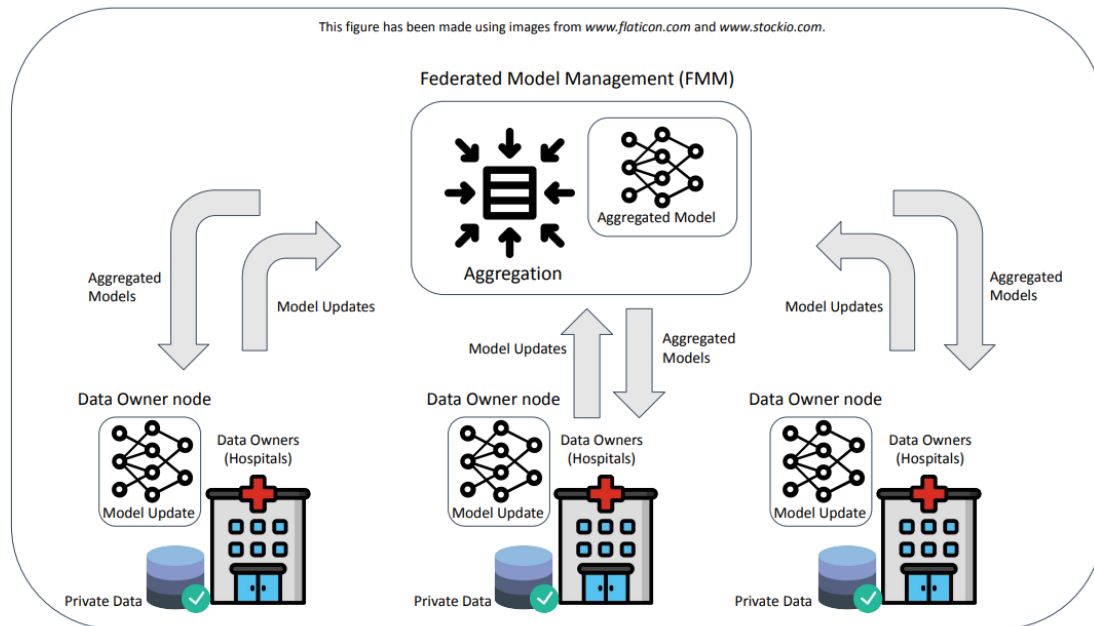


Figure 1: High-level description of a cross-silo FL flow chain.

However, further research has cast a shadow of doubt on the strength of privacy protection provided by FL [2]. Potential vulnerabilities and threats pointed out by researchers reveal that FL approaches are susceptible to curious aggregator threat, to man-in-the-middle and insider attacks that disrupt the convergence of the models; and, most importantly, to inference attacks that aim to re-identify data subjects from FL's AI model parameter updates [3] [4] [5]. Different attacks have been developed that compromise data privacy, allowing an adversary to determine if a specific individual's record is part of the training data, infer attributes about the individual, or even reconstruct the training data

In view of the strong privacy protection stance of the European Union, expressed through the CyberSecurity Strategy, the CyberSecurity Act and the GDPR, these new findings regarding the privacy guarantees provided by FL are harmful to the proliferation of FL as a technology with privacy and security guarantees.

Thread models for Federated Learning

FL is characterized by a collaborative training pipeline in which Data Owners (DOs) keep private their respective datasets while iteratively exchanging local model updates with another party denoted as Aggregator (Agg). Although data is not explicitly shared, we can still find security and privacy risks related to data training leakage, data subject re-identification or even global model corruption.

Threat models in FL inherit their features from conventional ML training, but they can be considerably aggravated by the iterative and interactive exchange of local model updates between DOs and the aggregator. In this section, we will briefly explore the most important privacy threats applicable to a FL setting [6] [7].

First, **inference attacks** may attempt to inspect the AI model updates (see *Figure 1*) and reconstruct from them the original data (DRA, *Data Reconstruction Attack*), infer certain attributes of individual data records (PIA, *Attribute Inference Attack*) or even infer membership of a certain subject in a dataset (MIA, *Membership Inference Attack*). For example, if one of the Data Owners is hospital X, an inference/re-identification attack may disclose that Ms. Y is a patient of X, and an attribute leakage attack may reveal that she has cancer. The inference attacks may be performed by both aggregator or any data owner. The aggregator may be able to inspect many local model updates from the same data owner, giving it white-box access to different local ML training models over the same training data. On the other side, the data owners may be able to inspect many aggregated model updates, granting them white-box access to different ML trained models over the same collective training data.

The presence of more malicious participants implies stronger attacks. For instance, in **poisoning attacks** a malicious participant manipulates the training process by providing arbitrary model updates to poison the global model for their own benefit. Even if this type of attack seems to focus on reducing the performance, by disrupting the convergence of global and local models or causing convergence to fake minima, it can also be used to make the training algorithm converge to an ML model more vulnerable to model inference attacks. On the one hand, the aggregator can adversarially design ad-hoc aggregated updates as a means to infer in the next round more information from one or several of the DOs' training data. On the other hand, other DOs can adversarially perturb their respective local model updates to maximize the leakage of other DOs' training data in subsequent aggregation rounds.

It is important to point out that several malicious parties (both DOs and/or aggregator) can collude to maximize the training data leakage produced by the model updates of a particular DOs, increasing the potential damage of their attacks.

Privacy enhancing technologies for FL

In order to mitigate and prevent the attacks exposed above, we can resort to the field of Privacy Enhancing Technologies (PETs), which has evolved rapidly in recent years, providing increasingly efficient and feasible solutions to the problem of securely processing and sharing sensitive private data. This has resulted in a diverse set of PET flavors [8], where the choice of a specific technique for a particular use case depends on the available resources and the privacy problem to be addressed. Unfortunately, significant obstacles still exist that limit the adoption of PETs in general applications. The obstacles vary from degradation of utility of data, to high computational and communication costs. This section provides a short overview of the most relevant PETs that can be used to strengthen the baseline privacy provided by FL. We classify them here into two groups: encryption-based and non-encryption based.

First of all, **Homomorphic Encryption (HE)** allows to perform computations directly on encrypted data [9]. While it provides input privacy for the model updates, it also presents a higher computational cost than other PET techniques. It requires the use of other primitives

(e.g., Zero-Knowledge Proofs or ZKPs) to upgrade the security model against stronger adversaries. On the other hand, **Secure Multi-Party Computation (SMPC)** allows a group of parties to jointly compute a function while keeping the parties' inputs secret. In general, many SMPC solutions [10] (for example those based on the use of Secret Sharing [11]) present a lower computational cost than HE, but usually require a higher number of communication rounds.

Focusing on the non-encryption-based PETs, **Differential Privacy (DP)** offers statistical privacy guarantees by adding controlled statistical noise (Gaussian, Laplacian, or using other distributions) to attributes of individual data records before sharing them. While it provides the lowest computational overhead among all the PETs mentioned here, DP [12] entails a tradeoff between privacy level and utility. Initially developed to protect databases, these techniques have been successfully applied to enhance privacy in FL schemes by applying them to the weights or gradients exchanged during the federated training process. By employing DP mechanisms, the influence of a specific data point on the trained algorithm is diluted. The main parameter of these mechanisms is the **privacy budget**, which defines the amount of noise inserted into the algorithm. When choosing the privacy budget, it is important to consider that the noise accumulates with each iteration of the FL training, as more iterations reveal more information about the model's learning. A smaller privacy budget provides greater privacy but also results in a higher degradation of the model utility. The problem of adjusting the privacy budget can be approached from three main perspectives: creating different mechanisms, studying how the privacy budget accumulates in each iteration (by discovering properties of the mechanisms), or applying the mechanism at different stages of the FL training.

A combined approach for an armored FL

The techniques presented in the previous section cannot simultaneously protect against all the privacy leaks and attacks that can be present in an FL setting. So, composing an end-to-end FL system will typically involve combining several PETs in order to attain all the desired privacy properties.

For example, the main difference between HE and SMPC relies on the used computational resources: HE requires less interaction between the parties, but has a higher computational cost, while SMPC has a lower computational cost but the communication becomes the main bottleneck. A possible way of combining HE and SMPC to benefit from both approaches is Multi-Key Homomorphic Encryption (MKHE): most existing HE-based solutions for secure computation work under a single-key approach in which ciphertexts are encrypted under the same key. However, this single-key approach is clearly impractical for general multi-party computation with more than two parties. Consequently, while it is true that HE allows us to make general computations over encrypted data, its use must be adapted to the specific needs of FL. Still, many of the current HE-based solutions for secure aggregation work under this **single-key approach**, where the ciphertexts provided by all data owners are encrypted under the same key. This introduces a relevant drawback in FL which consists in the possibility of a **collusion between the aggregator and some of the data owners** [13] [14]. If the aggregator sends the ciphertexts received from one data owner to another one, who also owns the secret key, all inputs provided by the former could be decrypted by the latter.

With a HE whose ciphertexts are encrypted under different keys or their combination, the aggregator could perfectly compute the required aggregation function and return the aggregated models to the final user. Now, by means of a collaborative protocol, parties could re-encrypt the output under the final user's key, or directly decrypt it. As input ciphertexts are not encrypted under the secret key of the final user, **the risk of a collusion with the cloud is removed.**

Next, we briefly detail the existing possibilities to have several keys and HE at the same time. With this aim, the "ideal" concept of a multi-key HE is divided into the two main families: Threshold HE and Multi-Key HE. First, **Threshold HE** requires a setup phase involving the joint participation of all the parties for the key generation process. In general, each party generates his/her own secret key and public key and, afterwards, all of them gather their keys to generate a collective public key. In contrast to Threshold HE, the setup phase of a **multi-key HE** scheme does not require the joint collaboration of all the parties, and each user can independently generate its own secret and public keys. Afterwards, ciphertexts under different public keys can be evaluated "on-the-fly", resulting in an output ciphertext whose decryption will depend on the secret keys of all the input ciphertexts used in the circuit evaluation.

It is worth mentioning that threshold and multi-key variants present different advantages and shortcomings depending on the desired key setup complexity, evaluation efficiency, or system scalability, to name just a few practical performance indicators. Actually, in practice, hybrid systems can be designed which combine the best features of threshold and multi-key HE schemes. A general review of the problem of managing several keys can be found in [15], but only for data sharing. A review of the case for several keys in combination with HE, currently can be found in the works [16] [17] [18].

One of the main problems with the use of DP in an FL setting is that we must trust the aggregator. Focusing now on possible combinations of DP with other PET methods, Pentyala et al. [19] propose the use of Secure Multi-Party Computation (SMPC) in combination with DP to address this issue. The combination of SMPC with DP offers other benefits, such as the optimization of the use of the privacy budget by each of the computing parties: each party only needs to know its own desired level of privacy, its own function to be computed, and its measure of accuracy. If we focus now on HE instead of general SMPC, the same considerations as in SMPC follow. We can actually combine DP mechanisms with HE in a similar way [20], by adding noise to the inputs before encryption, also during the homomorphic computation, or even directly to the outputs before decryption. More recently, some works are exploring more sophisticated combinations of both PET technologies.

Conclusions

In this paper we have explored the concept of Federated Learning (FL) and its significance in addressing privacy concerns in collaborative machine learning scenarios. While FL initially appeared as a promising approach for preserving data privacy while training machine learning models across multiple entities, recent research has raised questions about its practical effectiveness in protecting sensitive information. Various threats, including inference attacks, poisoning attacks, and collusion among malicious parties, have exposed vulnerabilities in FL, undermining its role as a privacy-preserving technology.

To reinforce the privacy guarantees of FL, we discussed the role of Privacy Enhancing Technologies (PETs) as a critical component. PETs offer a range of tools and techniques for safeguarding data and model privacy in FL systems. These technologies can be broadly categorized into encryption-based and non-encryption-based methods, each with its strengths and trade-offs.

In the search for comprehensive privacy protection in FL, we explored different examples of PET combinations. In particular, we examined the synergy between Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC) to balance computational efficiency and communication overhead, and the potential of combining Differential Privacy (DP) with SMPC or HE to enhance the protection of FL systems.

While each PET has its benefits and drawbacks, achieving a holistic and robust privacy-preserving FL system requires thoughtful integration of these technologies. The choice of PETs should align with the specific privacy challenges and computational requirements of each use case. Moreover, ongoing research and development efforts are exploring innovative ways to combine multiple PETs effectively to mitigate various privacy threats.

The adoption of FL as a secure and privacy-preserving technology remains an open research problem, especially in light of evolving privacy regulations and increasing data collaboration needs.

Acknowledgements

This publication was supported by the TRUMPET project, funded by the European Union under Grant Agreement No. 101070038

References

- [1] M. Khan, F. G. Glavin, and M. Nickles, "Federated learning as a privacy solution - an overview," *Procedia Computer Science*, vol. 217, pp. 316–325, 2023, 4th International Conference on Industry 4.0 and Smart Manufacturing
- [2] Alberto Pedrouzo-Ulloa, Jan Ramon, Fernando Pérez-González, Siyanna Lilova, Patrick Dufloy, Zakaria Chihani, Nicola Gentili, Paola Ulivi, Mohammad Ashadul Hoque, Twaha Mukammel, Zeev Pritzker, Augustin Lemesle, Jaime Loureiro-Acuña, Xavier Martínez, Gonzalo Jiménez-Balsa. "Introducing the TRUMPET project: TRUStworthy Multi-site Privacy Enhancing Technologies." *CSR 2023*: 604-611.
- [3] M. Mansouri, M. Onen, W. B. Jaballah, and M. Conti, "Sok: Secure " aggregation based on cryptographic schemes for federated learning," *Proc. Priv. Enhancing Technol.*, vol. 2023, no. 1, pp. 140–157, 2023.
- [4] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NIPS*, 2017, pp. 119–129.
- [5] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *IEEE Symposium on Security and Privacy*. IEEE, 2019, pp. 739–753.

[6] Peter Kairouz; H. Brendan McMahan; Brendan Avent; Aurélien Bellet; Mehdi Bennis; Arjun Nitin Bhagoji; Kallista Bonawit; Zachary Charles; Graham Cormode; Rachel Cummings; Rafael G. L. D'Oliveira; Hubert Eichner; Salim El Rouayheb; David Evans; Josh Gardner; Zachary Garrett; Adrià Gascón; Badih Ghazi; Phillip B. Gibbons; Marco Gruteser; Zaid Harchaoui; Chaoyang He; Lie He; Zhouyuan Huo; Ben Hutchinson; Justin Hsu; Martin Jaggi; Tara Javidi; Gauri Joshi; Mikhail Khodak; Jakub Konečný; Aleksandra Korolova; Farinaz Koushanfar; Sanmi Koyejo; Tancrede Lepoint; Yang Liu; Prateek Mittal; Mehryar Mohri; Richard Nock; Ayfer Özgür; Rasmus Pagh; Hang Qi; Daniel Ramage; Ramesh Raskar; Mariana Raykova; Dawn Song; Weikang Song; Sebastian U. Stich; Ziteng Sun; Ananda Theertha Suresh; Florian Tramèr; Praneeth Vepakomma; Jianyu Wang; Li Xiong; Zheng Xu; Qiang Yang; Felix X. Yu; Han Yu; Sen Zhao, *Advances and Open Problems in Federated Learning*, now, 2021.

[7] Lyu, Lingjuan & Yu, Han & Yang, Qiang. (2020). Threats to Federated Learning: A Survey.

[8] Big Data UN Global Working Group, UN Handbook on Privacy-Preserving Computation Techniques. <https://unstats.un.org/bigdata/task-teams/privacy/index.cshtml>, 2022. [Online]. Available: <https://unstats.un.org/bigdata/task-teams/privacy/index.cshtml>

[9] M. Chase et al., "Security of Homomorphic Encryption," HomomorphicEncryption.org, Redmond, WA, techreport, Jul. 2017.

[10] D. Evans, V. Kolesnikov, and M. Rosulek, "A Pragmatic Introduction to Secure Multi-Party Computation," *Foundations and Trends® in Privacy and Security*, vol. 2, pp. 70–246, 2018.

[11] Damgard, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart, "Practical Covertly Secure MPC for Dishonest Majority - Or: Breaking the SPDZ Limits," in *ESORICS*, 2013, vol. 8134, pp. 1–18.

[12] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*, 2006, pp. 1–12.

[13] Alberto Pedrouzo-Ulloa, Fernando Pérez-González, David Vázquez-Padín: Secure Collaborative Camera Attribution. *EICC 2022*: 97-98.

[14] Alberto Pedrouzo-Ulloa, Fernando Pérez-González, David Vázquez-Padín: Multi-Key Homomorphic Encryption for Collaborative Camera Attribution. Poster presentation. 5th HomomorphicEncryption.org Standards Meeting 2022.

[15] Jun Tang, Yong Cui, Qi Li, Kui Ren, Jiangchuan Liu, Rajkumar Buyya: Ensuring Security and Privacy Preservation for Cloud Data Services. *ACM Comput. Surv.* 49(1): 13:1-13:39 (2016).

[16] Elena Fuentes Bongenaar: Multi-key fully homomorphic encryption report. Retrieved from <https://www.cse.chalmers.se/~elenap/papers/Multi-key%20fully%20homomorphic%20encryption%20report.pdf>.

[17] Asma Aloufi, Peizhao Hu: Collaborative Homomorphic Computation on Data Encrypted under Multiple Keys. *CoRR abs/1911.04101* (2019).

[18] Asma Aloufi, Peizhao Hu, Yongsoo Song, Kristin E. Lauter: Computing Blindfolded on Data Homomorphically Encrypted under Multiple Keys: A Survey. *ACM Comput. Surv.* 54(9): 195:1-195:37 (2022).

[19] Sikha Pentylala, Davis Railsback, Ricardo Maia, Rafael Dowsley, David Melanson, Anderson C. A. Nascimento, Martine De Cock. "Training Differentially Private Models with Secure Multiparty Computation." *CoRR abs/2202.02625* (2022).

[20] Arnaud Grivet Sébert, Renaud Sirdey, Oana Stan, Cédric Gouy-Pailler: Protecting Data from all Parties: Combining FHE and DP in Federated Learning. CoRR abs/2205.04330 (2022).