Enhanced Security and Privacy via Fragmented Federated Learning

Najeeb Jebreel
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
najeeb.jebreel@urv.cat

Josep Domingo-Ferrer
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
josep.domingo@urv.cat

Alberto Blanco-Justicia
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
alberto.blanco@urv.cat

David Sánchez
Universitat Rovira i Virgili
Departament d'Enginyeria Informàtica i Matemàtiques
david.sanchez@urv.cat

***Abstract*—In federated learning (FL), a set of participants share updates computed on their local data with an aggregator server that combines updates into a global model. However, reconciling accuracy with privacy and security is a challenge to FL. On the one hand, updates sent by honest participants may reveal their private local information, whereas malicious updates may compromise the model's integrity. On the other hand, enhancing privacy via update distortion damages accuracy, whereas doing so with secure aggregation damages security because it prevents the server from analyzing individual updates. To tackle the accuracy-privacy-security conflict, we propose *fragmented federated learning* (FFL), in which participants randomly exchange and mix fragments of their updates before submitting them. To achieve privacy, we design a lightweight protocol that allows participants to privately exchange and mix encrypted fragments of their updates so that the server can neither obtain individual updates nor link them to their originators. To achieve security, we design a reputation-based defense tailored for FFL that builds trust in participants based on the quality of the fragments they exchange and the mixed updates they send. Since no noise is introduced and attackers can be neutralized, the server can correctly reconstruct a global model from the received updates without accuracy loss. Experiments on four real data sets show that FFL can prevent semi-honest servers from mounting privacy attacks, can effectively counter attacks and can keep the accuracy of the global model.**

***Index Terms*—Federated learning, Privacy, Security, Accuracy, Fragmentation.**

I. Introduction

Federated learning (FL, [1]) enables multiple participants to jointly train a machine learning (ML) model without sending their data to a central server. In FL, participants compute model updates based on their local data and a global model received from a coordination server, to whom they send the resulting updates. The server aggregates the received updates to obtain a new global model, which is redistributed among the participants. This collaboration entails mutual benefits for all parties: i) the participants and the server obtain more accurate models due to learning from joint training data, ii) participants keep their private local data in their devices, and iii) the server distributes the computational load of training across the participants' devices [2]. Since training data do not leave the participants' devices, FL is a suitable option for scenarios where data collection and processing are restricted due to privacy protection laws such as the GDPR [3]. Despite these advantages, FL is still vulnerable to some privacy and security attacks [4].

Regarding privacy, several works [5], [6], [7] have shown that a *semi-honest* server can infer sensitive information from individual updates. Recent privacy attacks show that it is possible to reconstruct training data by inverting the gradients of updates [8], [9].

Regarding security, FL is vulnerable to poisoning attacks [10]. Since the server has no control over the behavior of participants, any of them may deviate from the prescribed protocol to attack the model by conducting untargeted poisoning (*i.e.* Byzantine) attacks [11], [12] and targeted poisoning attacks [13], [14]. Byzantine attacks aim to degrade the model's overall performance, whereas targeted poisoning attacks aim to cause the global model to misclassify attacker-chosen inputs.

Some solutions have been proposed to prevent the server from analyzing individual updates or linking them to their originators. These involve well-known privacy-enabling methods: differential privacy (DP) [15], homomorphic encryption (HE) [16], and secure multiparty computation (SMC) [17]. DP-based methods protect the participants' data by injecting random noise into the parameters of updates at the cost of sacrificing the accuracy of the global model [18]. This conflicts with a key motivation of FL, namely improving the accuracy of the trained models. HE- and SMC-based methods securely aggregate the updates of participants before sending them to the server. This brings privacy to the participants and provides perfectly accurate aggregated updates to the server. Yet, HE and SMC have a high computational cost and they prevent the server from inspecting individual updates, which makes these approaches vulnerable to security attacks. Note that countermeasures against security attacks require direct access by the server to the individual updates to detect and/or filter out those that are poisoned [19], [11], [20].

Simultaneously achieving privacy, security and accuracy is a tough challenge for FL [21], [4], [10]. In fact, this is one of several challenges in ML due to contradicting requirements; see [22] about other conflicts.

Our goal in this paper is to prevent semi-honest servers from performing privacy attacks on individual updates while learning an accurate global model and ensuring protection against security attacks. To this end, we propose FFL, a framework in which participants randomly exchange fragments of updates among them before sending them to the server. See [23] for the full version of this paper.

II. Fragmented Federated Learning

The key idea of FFL is to have the participants randomly fragment and mix their updates before sending them to the server. Since the exchange of fragments does not change the original coordinate positions nor values, the server can accurately aggregate the mixed fragments to obtain updated global models. Mixed updates break the link between the updates and their originators, and do not give the server direct access to individual updates. Thus, FFL prevents a semi-honest server from mounting powerful privacy attacks.

Yet, exchanging fragments between participants should neither impose significant communication or computation overheads nor facilitate participant-side privacy attacks. Thus, we design a fragment exchanging protocol based on lightweight cryptography to: i) allow the participants to exchange and mix random fragments of their updates while incurring very minor communication and computation overheads, ii) prevent participants from seeing each other's original fragments, and iii) allow the server to correctly compute the updated global model from the mixed updates without being able to recover the individual original updates or link them to their originators.

Additionally, we design a reputation-based defense for FFL that builds trust in participants based on the quality of updates. Specifically, the server holds a global reputation vector and uses it to select participants for training and adaptively aggregate their mixed updates. Participants also hold local reputation vectors and use them to decide whether to exchange fragments with the other participants. An attacker who exchanges poisoned fragments with honest participants will eventually find no honest participant to exchange their fragments with.

A. FFL design

Our design includes K participants and an aggregator server A. We assume that at most $K/5$ may be malicious and attack the global model's integrity by sending poisoned updates to A. Also, we assume that A is semi-honest and may use the participants' updates to infer information about their local private data. Before starting any training task, A generates a public/private key pair (pk_A, sk_A) and broadcasts pk_A to the K participants. Then, the server uses the metadata of the participants' devices to register them into the system and assigns them a unique pseudonym. All parties are assumed to possess pairwise secure communication channels and communicate with each other using pseudonyms. At the first global training round, A starts an FL task by initializing the global model W^0 and the global reputation vector $\gamma^0 \in \mathbb{R}^K$. This vector is used to select participants for training and re-weight their mixed updates in the global model aggregation. Also, each participant k initializes a local reputation vector $\zeta_k^0 \in \mathbb{R}^{K-1}$ that is used to store a local reputation value for each other participant.

Adaptive selection of participants. At each round t, A selects a set S of potential honest participants. S is a set of at least 2 peers chosen among the participants with global reputations in the first quartile of γ^t. After that, A sends the global model W^t to the participants in S. The server also adds the pseudonyms of the selected participants to a public billboard seen by all participants.

Local training. Each participant $k \in S$ trains the received model on her private local data to obtain her local update W_k. Then, k scales her computed local update by the number of data points she holds, d_k. The next step is for k to randomly select another participant j from the public board to exchange a fragment of her encrypted update W_k with her.

Participant selection for exchange. k uses her local reputation vector ζ_k^t to decide whether to exchange fragments with any other participant j. To do so, k chooses and contacts a random participant from the public billboard whose local reputation is in the first quartile of ζ_k^t. j accepts the request from k if k's local reputation is in the first quartile of j's local reputation vector ζ_j^t.

Fragments exchanging and mixing. Let $PRNG(\cdot)$ be a public pseudo-random number generator that takes an integer as a seed, and λ the bit-length of parameters. Let $\mathbb{G}_p$ be the multiplicative group of integers modulo a large prime p, and g a generator of the group. All subsequent operations with vectors are performed coordinate-wise.

1) Participant k randomly generates integers a and s_{r_k}, and sends $g^a \mod p$ and $Enc_{pk_A}(s_{r_k})$ to j.
2) Participant j generates integers b and s_{r_j}, and computes $g^b \mod p$ and $r_j = PRNG(s_{r_j})$, of bitlength $|W|\lambda$. j generates a mask $m = PRNG(g^{ab} \mod p)$ of 0's and 1's of bitlength $|W|$ and $\neg m$ such that $m \oplus \neg m = \mathbf{1}_{|W|}$. Then, j sends $g^b \mod p$, $Enc_{pk_A}(s_{r_j})$, and $X_j = (W_j \odot m \oplus r_j)$ to k, where $\odot$ is an operator between an update and a mask that preserves the i-th update parameter if the i-th mask bit is 1, and clears to 0 the i-th update parameter if the i-th mask bit is 0.
3) k computes $r_k = PRNG(s_{r_k})$ of length $|W|\lambda$. k generates $m = PRNG(g^{ab} \mod p)$ and its inverse $\neg m$. Then, k computes her encrypted mixed update as follows:

$$\begin{aligned}(W_k)'_{mix} &= (W_k \odot \neg m) \oplus X_j \\ &= W_k \odot \neg m \oplus W_j \odot m \oplus r_j \\ &= (W_k)_{mix} \oplus r_j\end{aligned} \tag{1}$$

   Then, k sends $X_k = (W_k \odot m \oplus r_k)$ to j. She also sends $(W_k)'_{mix}, Enc_{pk_A}(s_{r_j})$ to the server.
4) j computes her encrypted mixed update as in Expression (1) (swapping subindices j and k).
   Finally, j sends $(W_j)'_{mix}, Enc_{pk_A}(s_{r_k})$ to the server.

In Expression (1), the result of subexpression $W_k \odot \neg m$ is to clear to 0 all parameters of W_k at coordinates where the mask m has a 1. On the other hand, subexpression $W_j \odot m$ clears to 0 all parameters of W_j where the mask m has a 0. Bitwise adding both subexpressions yields the mixed update $(W_k)_{mix}$. Note that r_j encrypts the mixed update into $(W_k)'_{mix}$. Since the mask m is random, the mixed update can be expected to contain the same number of 1s and 0s. Hence, $(W_k)_{mix}$ can be expected to contain half of the coordinates from W_j and the other half from W_k. Another important remark is that the mixed update is encrypted using r_j, whereas k has only $Enc_{pk_A}(s_{r_j})$, so k cannot obtain the full cleartext mixed update.

Decryption of encrypted mixed updates. A receives the pair $Enc_{pk_A}(s_{r_j})$ and $(W_k^{t+1})'_{mix}$ from each participant. A generates the OTP $r_j = PRNG(Dec_{sk_A}(Enc_{pk_A}(s_{r_j})))$ and bitwise adds it to $(W_k^{t+1})'_{mix}$ to obtain $(W_k^{t+1})_{mix}$.

Receiving $Enc_{pk_A}(s_{r_j})$ allows the server to decrypt the encrypted mixed update but does not allow it to extract any original fragment separately or link it to the fragment's originator. Moreover, the use of the DHKE [24] to generate seed g^{ab} ensures that no one but k and j knows the generated mask m.

Computation of reputation values and aggregation. We explain how reputations are computed and used to neutralize potential attackers in the system.

First, A computes the similarity between the received mixed updates and their centroid. We use the coordinate-wise median as a centroid value, given its resistance to outliers. Previous studies show that the impact of untargeted attacks is more evident in the magnitudes of whole updates, whereas the impact of the targeted attacks is more evident in the last-layer gradients [13]. Thus, we use the ℓ_1-distance over the whole updates to detect untargeted attacks and the cosine similarity of the last-layer gradients to detect targeted attacks. We the normalize both measures to the interval $[0, 1]$ and sum them, weighted by parameter α, to obtain a quality value.

Then, A computes the global reputation γ_k^{t+1} according to the quality measure and sends each participant k the change in their global reputation. Participants use this value to update the local reputation of the peer they exchanged their update with $\zeta_{k,j}^{t+1}$. Note that if the global reputation of a honest peer decreases, she can assume that the participant she exchanged her update with is an attacker. Thus, she will be less willing to exchange fragments with this user in future epochs.

Finally, the global reputations are used to compute the trust values ν^t for each of the received mixed updates. These these trust values are used to reweight the mixed updates when aggregating them to obtain the new global model.

III. Experimental Analysis

We tested the proposed method on three ML tasks: tabular data classification, image classification and sentiment analysis. Table I summarizes the data sets and models we used. To evaluate the effectiveness of FFL at defeating reconstruction attacks, we used the code provided by the authors of [9]. Our code and data are available for reproducibility purposes[1].

TABLE I: Data sets and models used in the experiments

Task	Data set	# Examples	Model	# Parameters
Table classif.	Adult	48K	MLP	∼5K
Image classif.	CIFAR10	60K	VGG16	∼15M
Sent. analysis	IMDB	50K	BiLSTM	∼13M

A. Robustness against security attacks

We next report results on the robustness of FFL against two attack strategies: in *Strategy 1*, the attacker generates two poisoned fragments, one of which he exchanges with another participant k, and he sends to the server a mixed update consisting of his other poisoned fragment and the good fragment of k; in *Strategy 2*, the attacker exchanges a poisoned fragment with k and sends a fully poisoned update to the server.

Tables II and III show that FFL can effectively defend against untargeted and targeted poisoning attacks, respectively, while preserving the benign model performance. Moreover, FFL outperforms the state-of-the-art defenses in achieving good model performance while preventing the attackers from mounting successful security attacks and hindering the semi-honest server from mounting privacy attacks.

[1]https://github.com/najeebjebreel/FFL

B. Protection against the reconstruction privacy attack

FFL offers protection against the most powerful privacy attack, namely the reconstruction attack proposed in [9]. This attack does not require auxiliary data and can estimate the private training data by inverting their corresponding gradients. Fig. 1 shows an example of the results of a reconstruction attack carried out against mixed updates.

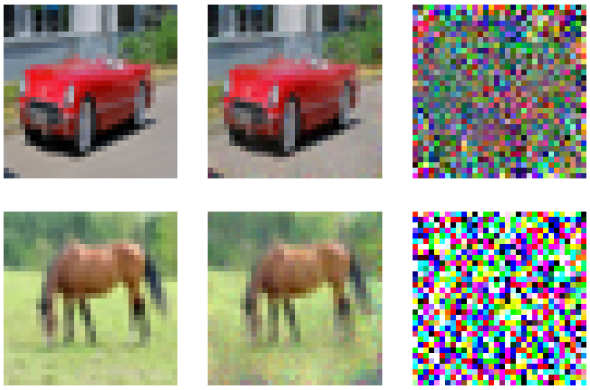


Fig. 1: Reconstruction of two input images from the gradients of two participants. Left: Two input images. Middle: Reconstruction from original gradients (FL). Right: Reconstruction from mixed gradients (FFL).

IV. Conclusions and Future Work

FFL leverages cooperation between participants in FL systems to preserve their privacy without renouncing robust and accurate global models. FFL offers a practical solution where participants privately and efficiently exchange random fragments of their updates before sending them to the server. We have also proposed a reputation-based defense tailored for FFL that builds trust in the participants based on the quality of updates they send and the fragments they exchange. We have demonstrated that the proposed framework can effectively counter privacy and security attacks. All the above is achieved while obtaining a global model's accuracy similar to that of standard FL and imposing affordable communication cost and computation overhead on the participating parties.

Acknowledgments

This work has received partial support from the European Commission (project H2020-871042 "SoBigData++"), the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and D. Sánchez, and grant 2021 SGR 00115), MCIN/AEI/ 10.13039/501100011033 and "ERDF A way of making Europe" (grant PID2021-123637NB-I00 "CURLING"), and INCIBE and European Union NextGenerationEU/PRTR (project "HERMES" and INCIBE-URV Cybersecurity Chair).

TABLE II: Robustness against untargeted attacks [12]. The percentage of attackers was 20% for all benchmarks. Best scores are in bold. TE is the test error or loss, All-Acc is the overall accuracy.

Benchmark/ Method		Adult-MLP		CIFAR10-VGG16		IMDB-BiLSTM	
		TE	All-Acc%	TE	All-Acc%	TE	All-Acc%
FL	FedAvg (no attacks) [1]	0.349	82.56	0.881	80.77	0.475	88.63
	FedAvg [1]	1.198	75.08	10.324	10.0	0.618	67.7
	Median [19]	0.349	**82.87**	1.020	78.91	0.524	88.33
	Trimmed mean [19]	0.350	82.61	1.046	**80.23**	**0.457**	**88.79**
	Multi-Krum [11]	0.350	82.69	0.998	78.86	0.565	87.72
	FoolsGold [13]	44.051	69.47	12.813	9.02	2.694	51.61
FFL	Strategy 1	**0.349**	82.84	**0.931**	79.74	0.536	88.15
	Strategy 2	0.350	82.86	0.938	79.26	0.542	87.48

TABLE III: Robustness against targeted label-flipping attacks [20]. The percentage of attackers was 20% for four benchmarks. Best scores are in bold. Src-Acc is the source class accuracy, which is the accuracy of the test samples of the source class. ASR is the attack success rate, which is the proportion of targeted examples which are incorrectly classified into the attacker-chosen class.

Benchmark/ Method		Adult-MLP			CIFAR10-VGG16			IMDB-BiLSTM		
		TE	Src-Acc%	ASR%	TE	Src-Acc%	ASR%	TE	Src-Acc%	ASR%
FL	FedAvg (no attacks) [1]	0.349	38.5	61.46	0.881	70.7	14.5	0.519	87.16	12.84
	FedAvg [1]	0.371	11.7	88.28	0.913	53.8	29.3	0.562	63.61	36.39
	Median [19]	0.352	33.6	66.4	0.943	58.0	24.8	0.643	59.92	40.08
	Trimmed mean [19]	0.350	34.73	65.27	0.943	58.4	25.5	0.649	58.2	41.8
	Multi-Krum [11]	**0.349**	**39.98**	**60.02**	0.896	42.6	39.3	0.886	39.05	60.95
	FoolsGold [13]	0.350	39.37	60.63	0.989	67.4	16.6	0.549	**86.86**	**13.14**
FFL	Strategy 1	0.350	39.59	60.41	**0.849**	**68.90**	**13.20**	**0.544**	86.64	13.36
	Strategy 2	**0.349**	39.90	60.10	0.892	68.80	15.00	0.524	86.16	13.84

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[3] European Commission, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)," 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[5] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[6] 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 *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[7] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.

[8] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[9] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?" *arXiv preprint arXiv:2003.14053*, 2020.

[10] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, D. Sánchez, A. Flanagan, and K. E. Tan, "Achieving security and privacy in federated learning systems: Survey, research challenges and future directions," *Engineering Applications of Artificial Intelligence*, vol. 106, p. 104468, 2021.

[11] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 118–128.

[12] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4583–4596, 2020.

[13] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)*, 2020, pp. 301–316.

[14] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[15] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[16] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, 2009, pp. 169–178.

[17] A. C. Yao, "Protocols for secure computations," in *23rd annual symposium on foundations of computer science (sfcs 1982)*. IEEE, 1982, pp. 160–164.

[18] J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "The limits of differential privacy (and its misuse in data release and machine learning)," *Communications of the ACM*, vol. 64, no. 7, pp. 34–36, 2021.

[19] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[20] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[21] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[22] Y. Liu, B. Guo, X. Zou, Y. Li, and S. Shi, "Machine learning assisted materials design and discovery for rechargeable batteries," *Energy Storage Materials*, vol. 31, pp. 434–450, 2020.

[23] N. M. Jebreel, J. Domingo-Ferrer, A. Blanco-Justicia, and D. Sánchez, "Enhanced security and privacy via fragmented federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 5, pp. 6703–6717, 2024.

[24] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.