Byzantine-Resilient Federated Learning: Evaluating MPC Approaches

Yasin Abdullah
Department of Computer Algebra
ELTE Eotvos Lorand University
Budapest, Hungary
p7hnjx@inf.elte.hu

Mohammed B. Alshawki
Department of Computer Algebra
ELTE Eotvos Lorand University
Budapest, Hungary
alshawki@inf.elte.hu

Peter Ligeti
Department of Computer Algebra
ELTE Eotvos Lorand University
Budapest, Hungary
ligetipeter@inf.elte.hu

Wissem Soussi
Communication Systems Group, Department of Informatics
University of Zurich
Zurich, Switzerland
sous@zhaw.ch

Burkhard Stiller
Communication Systems Group, Department of Informatics
University of Zurich
Zurich, Switzerland
stiller@ifi.uzh.ch

***Abstract*—Federated learning (FL) has emerged as a paradigm shift for collaborative machine learning to preserve data privacy. However, without considering the security measures through relevant cryptographic mechanisms, the collaborative process is vulnerable to various attacks. This paper evaluates the strength and scalability of Semi2k protocol for secure Multi Party Computation (MPC) under two major attacks, namely label-flipping and min-max attacks. We established a controlled simulations involving various numbers of malicious clients and MPC nodes. Our result showed that Semi2k offers limited protection against min-max attacks, showing no advantage over non-MPC setups in short training runs. However, it significantly improves accuracy under label-flipping attacks at 500 iterations, though overall accuracy declines with more malicious clients. Longer training improves resilience to label-flipping but increases communication overhead. Communication costs grow linearly with participants, highlighting a trade-off between scalability and efficiency.**

***Index Terms*—Federated Learning, MPC Protocols, Byzantine Attacks.**

I. Introduction

Federated Learning (FL) is a distributed machine learning approach that allows several clients to simultaneously train a model with a central server without revealing their private local data [1]. It has recently emerged as a paradigm shift in collaborative machine learning, where decentralized entities train a shared model together without the necessity to share the local data [2]. Each client, trained on its local dataset, only sends the model changes to the central server, which combines them to improve the global model. However, FL faces significant challenges in adversarial environments, where malicious participants, known as Byzantine users, seek to compromise the learning process. Poisoning attacks, such as label-flipping and min-max attacks, inject model updates during aggregation to degrade model performance [3]. These vulnerabilities are further worsened by the inefficiency of traditional aggregation mechanisms like Federated Averaging in adversarial cases, which lacks robustness against malicious behaviour [4]. Furthermore, the potential for privacy leakage through shared model updates requires stronger defenses in privacy-sensitive applications [5].

The critical challenges are evident—particularly in high-stake domains such as health care, finance, and Internet of Things (IoT) systems, where FL has gained increasing adoption—which needs to be addressed. In such scenarios, the integrity and privacy of federated models become crucial since adversarial manipulations can cause fatal results such as misdiagnosis, financial fraud, or security breaches of critical infrastructures. The need for the reliability of FL systems requires a robust aggregation approach capable of neutralizing adversarial influence without overlooking privacy preservation.

Secure Multi-Party Computation (MPC) is a cryptographic technique that allows multiple parties or nodes to jointly compute a function on their inputs while keeping those inputs private. When integrated into FL, MPC enables secure aggregation of model updates by having each client share encrypted fragments of its update with several computation parties, which prevents any single node from accessing complete data and mitigates adversarial attacks. This paper utilizes the SAFEFL [6] framework to evaluate MPC-enabled FL under adversarial conditions. We have introduced SAFEFL framework in more details in Appendix A. In this paper, we focus on assessing the robustness of the FL framework that relies on Federated Averaging (FedAvg) aggregation technique against label-flipping and min-max attacks with varying numbers of Byzantine participants. These two attacks have been briefly defined in Appendix C. FedAvg is chosen as the baseline aggregation method due to its simplicity and efficiency in diverse FL environments. It is a widely used aggregation technique that computes the weighted average of client updates. However, its vulnerability to adversarial behaviours, such as label-flipping and min-max attacks, makes

This research was supported by the European Union's Horizon 2020 research and innovation programs through the NATWORK project (Grant agreement ID: 101139285)

it ideal for evaluating the effectiveness of secure aggregation mechanisms like MPC [3].

The choice of FedAvg facilitates analyzing trade-offs between computational efficiency, privacy, and robustness, making FedAvg a critical foundation for assessing and improving secure and privacy-enhanced FL systems. We assessed the computational and communication overhead introduced by Semi2k MPC protocol. Appendix B describes the protocol. Through this analysis, our objective is to evaluate the effectiveness of MPC-enabled FL in mitigating adversarial impacts while analyzing the computational and communication trade-offs, thereby providing practical insights into deploying secure and robust FL systems. Simulations in this paper were conducted with varying ratio of adversarial influence ($10\%, 20\%, 30\%, 40\%$ Byzantine clients) to reflect real-world heterogeneity. These scenarios evaluate the scalability and robustness of MPC-enabled FedAvg against increasingly adversarial environments, ensuring the feasibility of deploying such systems in dynamic, real-world settings.

II. Related Works

Many have studied the security of MPC-enabled FL, analyzing its strengths and limitations under various adversarial settings [5], [7]–[10]. Li et al. [7] conducted a comprehensive assessment to mitigate Byzantine attacks in FL, emphasizing the vulnerabilities of commonly used aggregation methods such as FedAvg in adversarial environments. Their review highlights that FedAvg, while computationally efficient, lacks mechanisms to distinguish between honest and malicious updates, making it susceptible to Byzantine attacks. Such attacks, like label-flipping and min-max attacks, can significantly degrade model accuracy by introducing adversarial gradients into the aggregation process. To address this, they explore robust aggregation methods, including Trim-Mean and Krum, among the earliest techniques developed to filter out adversarial updates. Trim-Mean eliminates outlier gradients by finding the mean after removing the highest and smallest values, while Krum chooses gradients with the fewest deviations based on pairwise distances. While effective in limited adversarial scenarios, these techniques face scalability challenges in larger, non-iid environments. Their limitations underscore the need for advanced solutions like MPC-based aggregation to handle sophisticated adversarial attacks and diverse data distributions.

Kaminaga et al. [5] worked on improving FL security with secret sharing and MPC. Their work provides two important contributions to improving safe and reliable FL systems. First, it uses several datasets to examine the integration of secret-sharing-based MPC affecting model performance across various learning tasks, including image and activity classification. Second, it compares this privacy-enhancing strategy with both centralized machine learning and traditional FL architectures that do not employ privacy-enhancing technologies (PETs) to perform a thorough performance and resource efficiency analysis. The result emphasized that incorporating MPC introduces a manageable computational overhead (e.g., CIFAR10 average aggregation time of 16.43 ± 0.32s and upload time of 10.10 ± 0.54s) while delivering substantial improvements in data privacy by eliminating exposure of individual model updates. Interestingly, MPC also maintains model performance close to traditional FL—for example, on the MNIST dataset, FL with MPC achieved 99.2% accuracy, nearly matching centralized training. These results highlight the critical trade-offs between the added security benefits and the marginal computational overhead incurred, offering practical insights into optimizing FL systems with integrated privacy mechanisms. Built on this, our work evaluates Semi2k protocol under label-flipping and min–max attacks to further clarify the balance between robustness and communication cost in real-world scenarios.

III. Implementation

This section describes the practical implementation and provides a clear understanding of how the simulations in this paper were conducted.

A. Hardware and Virtual Machine Specifications

The simulations were conducted on a virtual machine hosted on Azure, configured to provide sufficient computational resources for running FL simulations and the cryptographic computations required by the MP-SPDZ framework. The virtual machine specifications are as follows:

- **Capacity:** Standard E4as v4 (4 vCPUs, 32 GiB memory).
- **Processing:** AMD EPYC 7763 64-Core Processor x64 Architecture.
- **Operating System:** Ubuntu 24.04 LTS (Server Edition).
- **Storage:** Premium SSD LRS.

B. Software Dependencies

The SAFEFL framework was installed with all dependencies specified in the requirements file. However, a modification was made to resolve compatibility issues with the `hdbscan` library, updating the dependency to `hdbscan==0.8.31`. The key dependencies are Python 3.8, PyTorch 1.11.0, NumPy 1.21.5, MP-SPDZ Framework 0.3.2, Matplotlib 3.5.1, and Hdbscan 0.8.31 (modified version).

The MP-SPDZ framework was configured to support the Semi2k protocol for secure MPC, which is central to providing privacy-preserving aggregation in SAFEFL.

C. Scenario Descriptions

The following controlled scenarios were designed to systematically evaluate the performance and robustness of the FL framework under various adversarial conditions. The simulation setups varied based on the number of Byzantine clients, attack types, MPC nodes, and aggregation rounds.

The HAR (Human Activity Recognition) dataset was chosen for the experiment and partitioned among 30 federated clients, each receiving data from different subjects with distinct class distributions to simulate a decentralized, heterogeneous, non-IID environment. More details are provided in Appendix D

Many deep learning studies adjust training durations based on convergence, often using early stopping criteria; however, we intentionally fixed the iteration counts at 50 and 500. The 50-iteration setting provides a measure of short-run efficiency and lower computational overhead, while the 500-iteration setting captures long-run robustness under various adversarial conditions. This controlled approach promotes reproducible comparisons across different configurations (e.g., varying numbers of Byzantine clients and attack types) by eliminating the variability inherent in dynamic training durations. In detail, we implemented two main scenarios as follows:

1) **Baseline Simulations**
   In this setting, we evaluated FedAvg under both benign and adversarial conditions. First, we conducted a control experiment without any attacks, and then introduced label-flipping attacks, followed by min-max attacks. We aimed to explore how normal and increasing malicious influences impact performance and accuracy by varying the number of Byzantine clients (3, 6, 9, and 12).
2) **MPC-Enabled Simulations** Similarly to the baseline, we evaluated FedAvg integrated with MPC (using the Semi2k protocol) by varying the number of Byzantine clients (3, 6, 9, and 12) and performing tests with 5 and 9 MPC nodes. We assessed the impact of label-flipping and min-max attacks to gain a complete understanding of how benign and increasing malicious influences affect the performance and accuracy of MPC-enabled FL.

D. Communication Diagram

The diagram illustrates a secure aggregation framework for FL, integrated with MPC for privacy-preserving model aggregation.

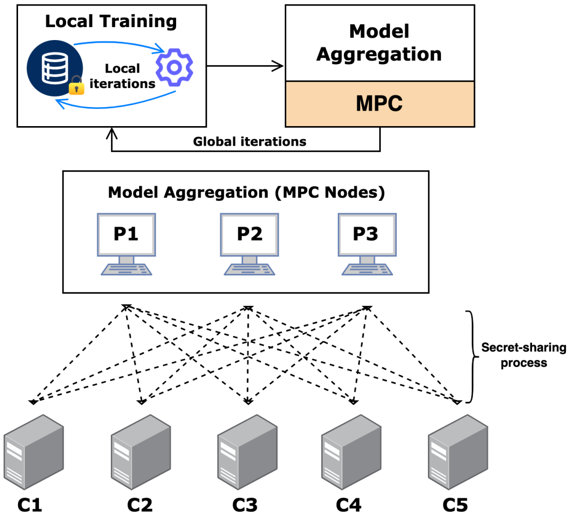

Fig. 1. MPC-Enabled FL Aggregation Process

First, a global model is initialized with a common architecture shared among all clients (C1–C5). Then, in the local training phase, the clients independently train their models on their local data without external sharing to preserve data confidentiality. After that, these model updates undergo a secret-sharing process, dividing each update into secure fragments to enhance privacy before being distributed among MPC nodes (P1, P2, and P3). Subsequently, during the aggregation phase through the nodes, these fragmented updates are collaboratively and securely aggregated without a central aggregator. This mechanism prevents individual reconstruction of client-specific model contributions by using secret-shared updates. This secure aggregation produces a global model that is then redistributed again to the clients for further training rounds.

The MPC nodes collectively serve as a decentralized aggregation committee, individually performing secure computations without full access to complete model updates. This decentralized approach allows robust privacy protection since no single node can access all information, thus providing resilience against potential malicious actions as long as a sufficient number of nodes remain honest. In parallel, FL clients maintain autonomy by only interacting with MPC nodes via encrypted secret shares, which prevents direct client-to-client communication and further provides data privacy. The nodes aggregate these encrypted updates securely, reconstructing and distributing the global model update back to the clients through an iterative process.

E. Parameter Settings

The simulations were executed using the following parameters, which reflects certain scenario descriptions: Number of Workers: 30, Batch Size: 128, Iterations (NITER): 50 and 500, MPC Protocol: Semi2k, MPC Parties: 5 and 9 computation nodes, Byzantine Setups: Varying malicious clients (3, 6, 9, 12), Thread and Parallelism: 4 threads and 2 parallels, Chunk Size: 500 (for MPC aggregation).

It is noteworthy to mention that the clients and MPC nodes operate independently. The clients train local models and compute model updates, while the MPC nodes handle the secure aggregation. MPC nodes receive encrypted updates and execute secure aggregation on their own. The final aggregated update is reconstructed and returned to the FL framework for model updates. Without MPC, clients would send raw updates directly to a central server, aggregating the updates using FedAvg.

F. Evaluation Metrics

The performance and overhead of the FL framework are evaluated using three key metrics. **Model Accuracy** is measured on a test set $D = \{(x_i, y_i)\}_{i=1}^{N}$ and defined as $\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\hat{y}_i = y_i\}$ (with $\hat{y}_i = \arg\max f(x_i)$), representing the proportion of correctly classified samples; **Runtime Overhead** is the additional computational time resulted by the MPC protocol, calculated as $T_{\text{runtime}} = \sum_{i=1}^{N_{\text{iter}}} t_i$, where t_i is the time for the ith iteration, capturing the extra cost incurred during secure aggregation; and **Communication Costs** reflect the total data exchanged during the aggregation, defined as $C_{\text{comm}} = \sum_{i=1}^{N_{\text{rounds}}} d_i$, with d_i being the data (in MB) transmitted in the ith round, summed over all rounds.

IV. Analysis

The findings in this paper obtained from experiments conducted over a specified number of iterations (NITER) or rounds, state the vital role of MPC during the data aggregation process and highlight the trade-offs between computational efficiency, communication overhead, and enhanced privacy and robustness in the experiment.

A. MPC in FL

In the following, we demonstrate the resilience of the MPC-enabled FL framework against adversarial attacks. Note that the green line in the figures denotes the aggregated model accuracy from the control experiment (no MPC, no attacks), serving as a benchmark for our comparisons.

We have studied the effect of label-flipping and min-max attacks. Generally, label-flipping attacks were significantly more detrimental to the global model's accuracy than min-max attacks. For instance, with 12 malicious clients and 5 MPC nodes in 50 iterations, accuracy dropped to 17.30% (Figure 4), and this degradation worsened with longer iterations, consistent with the findings of Bhagoji et al. [3]. This behavior aligns with the attack mechanism, which distorts the trustworthiness of training data, severely disrupting the model's decision boundaries. Whereas min-max attacks caused a more gradual decline in accuracy, as seen in Figure 5, where accuracy for 12 malicious clients with 9 MPC nodes remained relatively high at 72.54%. This highlights its resilience, as previously observed by Blanchard et al. [4].

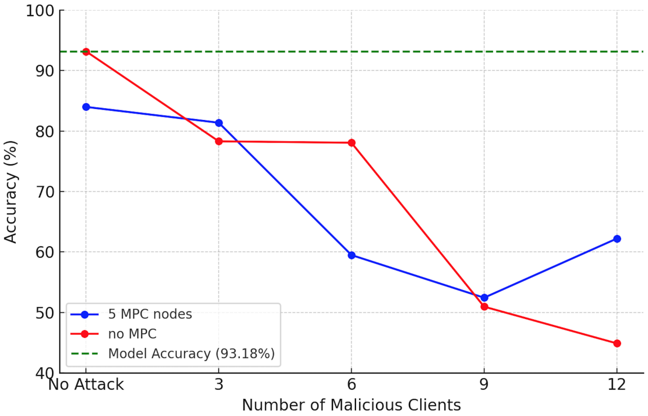

Fig. 2. Accuracy Trends: Label-flipping Attack (NITER 500)

Under label-flipping attacks, integration of MPC in FL still showed an accuracy decrease as the number of malicious clients increased, but it showed improvements compared to non-MPC setups. As shown in Figure 4, with 3 MPC nodes (NITER 50), the accuracy dropped from 67.67% (no attack) to 13.99% (12 malicious clients), while for 5 MPC nodes (NITER 500) in Figure 2, MPC improved accuracy from 44.88% (non-MPC) to 62.21% with 12 malicious clients. This showed that MPC mitigates the impact of label-flipping attacks more effectively than non-MPC configurations with longer iterations setup. In contrast, during min-max attacks, accuracy followed a more gradual decline, but Semi2k seemed to offer no significant improvement over non-MPC setups, even occasionally reduced accuracy, as shown in Figure 3. These results highlight that while Semi2k is more effective in mitigating label-flipping attacks under longer iterations, its benefits during min-max attacks remain limited.

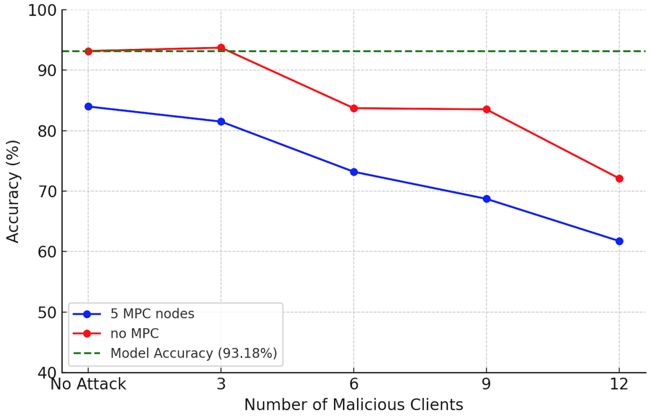

Fig. 3. Accuracy Trends: Min-max Attack (NITER 500)

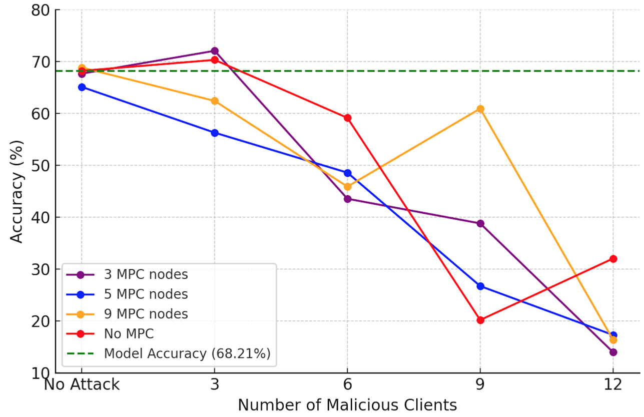

Fig. 4. Accuracy Trends: Label-flipping Attack (NITER 50)

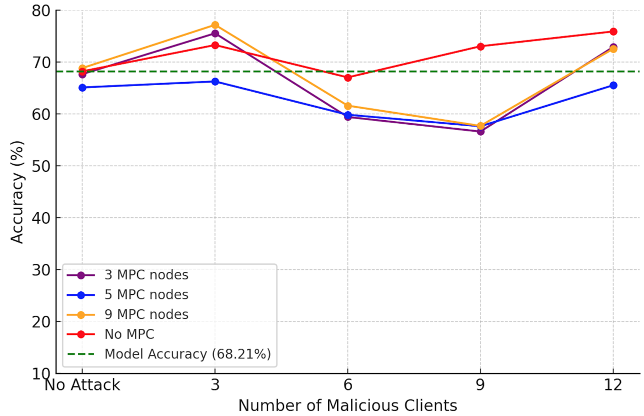

Fig. 5. Accuracy Trends: Min-max Attack (NITER 50)

B. Accuracy

Our evaluation under different attack scenarios highlights the accuracy performance of the FL with MPC compared to non-MPC configurations. As discussed in Section IV-A, we noted that the FL framework showed lower accuracy degradation in the case of min-max attacks when compared to label-flipping attacks, which had a more significant impact on the final accuracy. Furthermore, as the number of malicious clients increased, label-flipping attacks led to an even greater decline in accuracy. However, MPC-enabled FL showed notable improvements in accuracy under label-flipping attacks compared to non-MPC setups in 500 iterations. This showed that longer training durations helped mitigate some of the accuracy degradation caused by label-flipping attacks, highlighting Semi2k's resilience in this scenario, while accuracy for min-max attacks remained more stable across both iteration counts.

With NITER 50, Semi2k's handling of both label-flipping and min-max attacks showed no significant improvement over non-MPC setups but only added additional time overhead. For instance, in min-max attacks with 9 malicious clients, accuracy for 5 MPC nodes was 66.26%, increasing slightly to 67.04% for 9 MPC nodes (Figure 5). Similarly, for label-flipping attacks in the same scenario, the accuracy for 5 MPC nodes was 50.95%, dropping to 45.89% for 9 MPC nodes (Figure 4). These results highlight that in shorter iterations, Semi2k fails to provide meaningful resilience against either attack type, while increasing computation and communication costs. This reinforces the need for further optimizations to improve its efficiency and effectiveness in adversarial conditions.

C. Computation and Communication Cost

The computational and communication overhead introduced by the MPC protocol scales with the number of MPC nodes due to the added rounds of secret-sharing and reconstruction required for secure aggregation. For example, in table I, the 9 MPC nodes configuration consistently required more execution time than the 5 MPC nodes setup across all attack scenarios. This result is consistent with prior studies [11] highlighting the linear increase in MPC complexity with the number of computation parties.

TABLE I
Execution Time in NITER 50

MPC nodes	Label-flipping Attack (Execution Time)	Min-max Attack (Execution Time)
3 nodes	963,919 s	963,919 s
5 nodes	2947.45 s	2959.4 s
9 nodes	16291.2 s	16194.7 s

The type of attack, label-flipping or min-max, did not significantly affect the time overhead, indicating that the protocol's complexity is largely influenced by the number of MPC nodes rather than adversarial behaviour. This suggests that Semi2k's design ensures predictable performance regardless of attack type.

In tables II and III, data sent (party 0) represents the communication load on an individual computation party, reflecting the protocol's per-party requirements. As the number of MPC nodes increased, the data sent by party 0 and the global data transmitted grew proportionally. This trend aligns with the fundamental design of MPC protocols, where more participating nodes necessitate additional communication rounds for secure aggregation [6]. Despite the higher communication costs, the data load was evenly distributed across all parties, so no single participant experienced an unequal load. This balanced communication distribution, combined with the enhanced privacy and robustness benefits, underscores practically Semi2k for privacy-sensitive FL deployments.

TABLE II
Data Sent (Party 0) and Global Data Sent (NITER 50)

MPC nodes	Data Sent (Party 0)	Global Data Sent (All Parties)
3 nodes	146472 MB in ∼358076 rounds	438061 MB
5 nodes	551107 MB in ∼1379768 rounds	2.73944e+06 MB
9 nodes	2.13486e+06 MB in ∼5414000 rounds	1.90643e+07 MB

TABLE III
Data Sent (Party 0) and Global Data Sent (NITER 500)

MPC nodes	Data Sent (Party 0)	Global Data Sent (All Parties)
5 nodes	5.511e+06 MB in ∼13797520 rounds	2.73941e+07 MB

It is important to note that certain limitations constrained the scope of the analysis. The hardware constraints restricted the implementation to a maximum of 500 iterations and limited scalability testing to configurations with up to 9 MPC nodes (5 MPC nodes in case of 500 iterations). These constraints prevented a deeper exploration of long-term attack impacts and performance in larger FL setups. In addition, the static attack patterns used in the simulations did not capture real-world adversaries' dynamic and adaptive nature, which could significantly affect the protocol's evaluation.

V. Discussion and Use-cases

This section highlights the practical deployment recommendations of the Semi2k protocol in FL systems.

A. Scalability

As shown in our results, the Semi2k protocol shows strong scalability in certain scenarios. In addition, its ability to maintain privacy with more nodes aligns with the findings of Mohassel et al. [5], underscoring its potential for secure and scalable FL. This makes Semi2k suitable for large-scale systems, such as:

- **Collaborative Healthcare Models**: Hospitals across multiple regions train models collaboratively without compromising patient privacy, even when malicious actors attempt to distort results.

- **Financial Fraud Detection**: Banks securely aggregate transaction data across various parties to detect fraud, even with varying trust levels among participants.
- **Smart City IoT Applications**: Devices in a smart city setting contribute data securely for predictive modelling (e.g., traffic flow or environmental monitoring), ensuring scalability without exposing individual device data.

B. Overhead Consideration

As one critical aspect of deploying Semi2k, which scales with the number of MPC nodes, the experiments show that the communication rounds required for secure aggregation increased proportionally to the number of participants, as shown by the data in Table II.

For example, in a configuration of 9 MPC nodes, the global data sent during NITER 50 exceeded 1.9×10^7 MB, compared to 5.5×10^6 MB for a setup of 5 MPC nodes. These overheads are manageable in systems with robust networking capabilities (e.g., data centers) but may pose challenges in resource-constrained environments such as IoT devices. Such findings are consistent with the established linear growth in MPC protocols [11]. Therefore, Semi2k is best suited for applications where computational and communication resources are not severely limited, such as cross-institutional collaborations or data centers.

C. Security Considerations

The Semi2k protocol enables secure aggregation while mitigating the impact of adversarial participants. However, the type of attack significantly affects its effectiveness:

- **Min-max Attacks**: The secure aggregation mechanism did not demonstrate significant resilience against adversarial impact during min-max attacks in the experiment. However, its potential applicability in scenarios where adversarial influence is minimal or less aggressive, such as certain federated recommendation systems, requires further investigation.
- **Label-flipping Attacks**: These attacks remain a significant challenge, as accuracy drops sharply with the increase in malicious participants. This limitation highlights the need for additional robust aggregation methods or hybrid protocols to enhance Semi2k's performance in adversarial environments, as suggested in Sun et al. [12].

FL security can be enhanced against poisoning attacks while preserving the confidentiality afforded by MPC through the integration of poisoning-specific mitigation techniques. One promising approach is the Moving Target Defense (MTD) strategy [13]–[15]. This method addresses the challenges posed by non-IID data across participants by leveraging the MTD principle of dynamically altering the attack surface, thereby decreasing the success probability of poisoning attacks. For instance, Feng et al. [15] showcase a proactive selection of the aggregation algorithm, an algorithm pool including Krum, FedAvg, and Trim-Mean. Another MTD proactive action is the adjustment of the MPC aggregation topology, combined with a reactive deselection of suspected malicious MPC nodes, which are detected through a model similarity-based scoring system.

D. Deployment in Privacy-Sensitive Environments

The Semi2k protocol's secure aggregation capabilities make it highly applicable in scenarios where privacy is paramount:

- **Healthcare**: Collaborations across hospitals to train predictive models without exposing sensitive patient data.
- **Finance**: Federated fraud detection, where banks aggregate transaction data securely without exposing customer information.
- **Smart Cities**: IoT devices contributing data to centralized models for traffic prediction or environmental monitoring for individual device data to remain secure.
- **Telecommunication networks**: With the virtualization of network functions in telecommunication networks, the advent of virtual network operators (VNOs) using the same Telco Cloud infrastructure is greatly increasing. VNOs' automated management and orchestration of services can be optimised using secure FL without compromising the confidentiality of individual VNOs' private traffic.

These applications align with recent advancements in privacy-preserving machine learning [6], where secure aggregation protocols like Semi2k are increasingly seen as foundational.

VI. Conclusion

This paper highlights the effectiveness of MPC-enabled FL in mitigating adversarial attacks by utilizing Semi2k protocol in the data aggregation phase. Our results revealed that Semi2k demonstrates limited effectiveness in mitigating min-max attacks, with no significant improvement over non-MPC setups in shorter training iterations. In contrast, under label-flipping attacks, Semi2k showed notable improvements in accuracy compared to non-MPC configurations during 500 iterations, despite the overall decline in accuracy as the number of malicious clients increased. Longer training durations amplified the protocol's strengths and weaknesses, with higher iterations enhancing resilience in label-flipping scenarios but increasing communication overhead. Additionally, communication costs were observed to scale linearly with the number of participants, emphasizing the trade-off between scalability and computational efficiency.

Future studies are suggested to analyze other common attacks such as Trim and Scaling attacks, the combination of attacks, as well as to focus on improving the protocol's strength through enhancements such as filtering mechanisms or by adapting aggregation methods. Real-world testing would also be a critical step, as it would validate the protocol's scalability and performance under practical constraints, such as limited bandwidth and resource heterogeneity. These suggestions would address gaps to enhance the applicability of MPC-based aggregation protocols in diverse FL environments.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[2] Y. Yan, M. B. Alshawki, M. Zoltay, M. Gál, R. Hollós, Y. Jin, L. Péter, and Á. Tényi, “Fedlabx: a practical and privacy-preserving framework for federated learning,” *Complex & Intelligent Systems*, vol. 10, no. 1, pp. 677–690, 2024.

[3] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, “Analyzing federated learning through an adversarial lens,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[4] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] H. Kaminaga, F. M. Awaysheh, S. Alawadi, and L. Kamm, “MPCFL: Towards multi-party computation for secure federated learning aggregation,” in *Proceedings of the Conference on Aggregation and Secure Federated Learning*, ACM, 2023.

[6] N. Trieu and et al., “SAFEFL: An mpc-friendly framework for private and robust federated learning,” in *Proceedings of the 2021 IEEE Symposium on Security and Privacy (S&P)*, 2021.

[7] S. Li, E. C.-H. Ngai, and T. Voigt, “An experimental study of byzantine-robust aggregation schemes in federated learning,” *IEEE Transactions on Big Data*, 2023.

[8] C. Zheng, L. Wang, Z. Xu, and H. Li, “Optimizing privacy in federated learning with mpc and differential privacy,” in *Proceedings of the 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning*, pp. 165–169, 2024.

[9] L. Zhong, L. Zhang, L. Xu, and L. Wang, “Mpc-based privacy-preserving serverless federated learning,” in *2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pp. 493–497, IEEE, 2022.

[10] Y. Li, H. Li, G. Xu, T. Xiang, and R. Lu, “Practical privacy-preserving federated learning in vehicular fog computing,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4692–4705, 2022.

[11] D. Evans and et al., “Efficient secure multiparty computation for privacy-preserving federated learning,” *Journal of Cryptography and Information Security*, vol. 32, no. 1, pp. 45–65, 2018.

[12] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, “Can you really backdoor federated learning?,” in *Proceedings of the 2nd International Workshop on Federated Learning*, 2019.

[13] A. Amich and B. Eshete, “Morphence: Moving target defense against adversarial examples,” in *Proceedings of the 37th Annual Computer Security Applications Conference*, ACSAC ’21, (New York, NY, USA), p. 61–75, Association for Computing Machinery, 2021.

[14] A. Roy, A. Chhabra, C. A. Kamhoua, and P. Mohapatra, “A moving target defense against adversarial machine learning,” in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, (New York, NY, USA), p. 383–388, Association for Computing Machinery, 2019.

[15] C. Feng, A. H. Celdrán, Z. Zeng, Z. Ye, J. v. der Assen, G. Bovet, and B. Stiller, “Leveraging mtd to mitigate poisoning attacks in decentralized fl with non-iid data,” in *2024 IEEE International Conference on Big Data (BigData)*, pp. 7745–7754, 2024.

[16] R. Cramer, I. Damgård, D. Escudero, P. Scholl, and C. Xing, “Spdz: Efficient mpc mod 2 for dishonest majority,” in *Annual International Cryptology Conference*, pp. 769–798, 2018.

[17] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, “Secure single-server aggregation with (poly)logarithmic overhead,” in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’20, (New York, NY, USA), p. 1253–1269, Association for Computing Machinery, 2020.

[18] M. Keller, “MP-SPDZ: A versatile framework for multi-party computation,” in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[19] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2013.

Appendix A
SAFEFL Framework

SAFEFL is an MPC-based framework that evaluates FL techniques’ efficacy in privacy inference and poisoning attacks. At its core, SAFEFL is a communicator interface enabling PyTorch-based implementations, and the MP-SPDZ framework supports various MPC protocols. SAFEFL uniquely integrates MPC protocols with robust aggregation techniques, offering a flexible platform to simulate adversarial conditions and evaluate FL systems [6]. Its ability to configure various attack types (e.g., label-flipping, min-max), Byzantine proportions, and aggregation methods makes it a versatile tool for privacy and robustness analysis. This integration enables secure aggregation of local models, ensuring individual updates remain confidential despite adversarial manipulation.

The framework supports the simulation of diverse adversarial scenarios. It allows adjusting parameters such as the number of participants, the proportion of Byzantine (malicious) actors, and the types of attacks, including label-flipping and min-max strategies. In addition, SAFEFL can assess various aggregation methods for comparative analysis of robustness and efficiency. This type of flexibility offers a valuable instrument for developing and evaluating FL systems capable of addressing privacy inference and poison attacks. SAFEFL is utilized in this paper to simulate scenarios involving the FedAvg aggregation technique under label-flipping and min-max attacks. The resilience of FedAvg was evaluated in maintaining model performance in adversarial conditions by including various numbers of Byzantine participants. Furthermore, integrating Semi2k MPC protocol for evaluating computational and communication overheads introduced by secure aggregation can provide insights into the feasibility of deploying such systems in real-world applications. SAFEFL supports:

- **Aggregation Methods:** Multiple aggregation techniques, such as FedAvg, FLTrust, Trim-Mean, and Divide-and-Conquer.
- **Adversarial Scenarios:** Simulations of various attack types, including label-flipping, min-max attacks, scaling attacks, and tailored attacks such as Krum-specific adversaries. Parameters such as the number of participants, the proportion of Byzantine clients, and client data heterogeneity can be customized.
- **Secure Multi-Party Computations:** SAFEFL integrates multiple MPC protocols, including Semi2k, SPDZ2k [16], and Replicated2k, for secure aggregation that prevents adversarial clients from gaining sensitive information about other participants.

Appendix B
Multi-Party Computation

Multi-Party Computation (MPC) in FL enables secure aggregation by keeping the model updates private during the training process [5], [17]. MPC enhances privacy protection by ensuring updates remain encrypted or secret-shared throughout aggregation, addressing vulnerabilities in traditional FL methods where updates delivered to a central server could expose sensitive data [5]. MPC also provides robustness against collusion, where malicious clients or a compromised server could exploit shared updates, and enables secure aggregation during Byzantine attacks.

During the secret-sharing process, each client splits its local model updates into multiple ”shares” and securely distributes these shares among computation parties. This method ensures that no single party can reconstruct the original update without all shares. The computation parties collaboratively perform operations on the secret-shared inputs, such as summing model updates, without exposing individual updates.

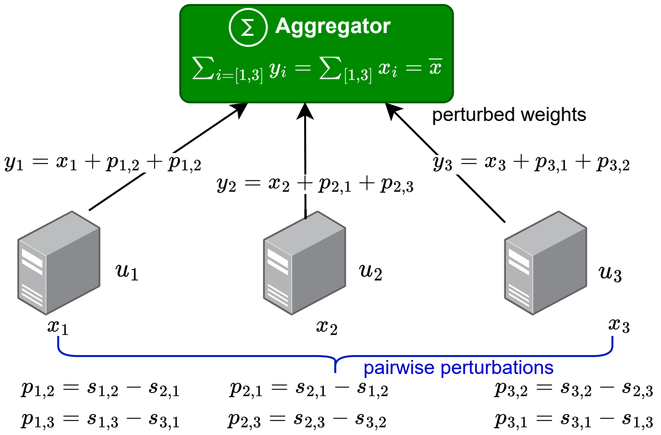
Σ Aggregator
$\sum_{i=[1,3]} y_i = \sum_{[1,3]} x_i = \overline{x}$
perturbed weights
$y_1 = x_1 + p_{1,2} + p_{1,2}$
$y_2 = x_2 + p_{2,1} + p_{2,3}$
$y_3 = x_3 + p_{3,1} + p_{3,2}$
u_1
u_2
u_3
x_1
x_2
x_3
pairwise perturbations
$p_{1,2} = s_{1,2} - s_{2,1}$
$p_{1,3} = s_{1,3} - s_{3,1}$
$p_{2,1} = s_{2,1} - s_{1,2}$
$p_{2,3} = s_{2,3} - s_{3,2}$
$p_{3,2} = s_{3,2} - s_{2,3}$
$p_{3,1} = s_{3,1} - s_{1,3}$

Fig. 6. MPC’s secure aggregation

Formally, each participant $u \in U$ formats its FL updates as a vector x_u of dimension k and composed of integers on the range $[0, R)$ for some known R. The elements of the vector:

$$\overline{x} = \sum_{u \ inU} x_u \tag{1}$$

resulting in the sum of all the participants’ vectors, should also be in the range $[0, R)$. Assuming a pair of participants *u* and *v*, *u* samples a vector $s_{u,v}$ uniformly from the $[0, R)^k$ for each other participant *v*. Participants *u* and *v* exchange $s_{u,v}$ and $s_{v,u}$ over a securely encrypted channel and compute perturbations:

$$p_{u,v} = s_{u,v} - s_{v,u}. \tag{2}$$

This leads to obtained perturbations such that:

$$p_{u,v} = -p_{v,u} \pmod{R} \text{ and for } u = v, p_{u,v} = 0 \tag{3}$$

Each participant then sends to the aggregator $y_u = x_u + \sum_{v \in U} p_{u,v}$ (mod *R*). Finally, the aggregator simply sums the perturbed values as the paired perturbations in y_u cancel each other, and what is obtained is simply the sum of all participants’ vectors, $\overline{x}$. Thus, as depicted in Figure 6, the aggregated global model update is reconstructed preserving the confidentiality of each client’s contributions.

In SAFEFL, MP-SPDZ is implemented as a state-of-the-art framework for executing various MPC protocols. We chose Semi2k protocol, implemented within the MP-SPDZ framework, as an optimized MPC protocol for secure and efficient computation in FL. It operates on a ring of size 2^k and provides semi-honest security, ensuring privacy as long as computation parties adhere to the protocol (i.e., parties are assumed to follow the protocol without deliberately deviating, even if they try to learn additional information from their received data). It supports efficient arithmetic operations such as addition and multiplication, which are integral to the aggregation process. Semi2k is opted for due to its strong scalability, accommodating a large number of participants with minimal computational overhead compared to protocols such as SPDZ2k—thanks to its modulo 2^k arithmetic that keeps all values within a fixed, hardware-friendly range [18].

Appendix C
Byzantine Attacks

Byzantine attacks represent one of the most significant challenges in FL, where malicious participants aim to disrupt the training process by injecting faulty or adversarial updates. This study focuses on two prominent attack types: label-flipping and min-max Attacks.

Label-flipping attacks involve mislabeling training data to misguide the global model, resulting in degraded classification accuracy. This type of attack is stealthy, as it exploits the assumption that the data of each client are inherently trustworthy [3]. On the other hand, min-max attacks manipulate gradient updates from malicious clients to maximize the deviation of the global model from its intended direction. These attacks appear statistically valid but are adversarial in nature. The results by Sun et al. [12] highlight the effectiveness of min-max attacks in bypassing simple aggregation rules like FedAvg, leading to catastrophic failures in the global model.

Appendix D
HAR Dataset

The Human Activity Recognition (HAR) dataset is a widely recognized benchmark for evaluating classification models in FL scenarios. The dataset consists of sensor data collected from smartphone accelerometers and gyroscopes, containing 561 input features derived from raw sensor signals and six activity labels: walking, walking upstairs, walking downstairs, sitting, standing, and lying down [19]. The dataset includes 10,299 samples divided into training and testing sets.

The HAR dataset is chosen for its relevance to FL due to its decentralized nature and diversity in data distributions. Each client’s data represents a unique user’s activity patterns, enabling the simulation of heterogeneous data distributions common in FL settings. Its high dimensionality and classification complexity make it a challenging benchmark for evaluating model robustness under adversarial conditions, such as Byzantine attacks. Additionally, its non-IID characteristics closely mirror real-world scenarios, making it ideal for testing the scalability and resilience of FL algorithms.