Post Quantum Secure Aggregation for Federated Learning Using Lattice-Based Cryptography

1st Rama Lakshmi Chunduri
Computer Science and Engineering (of Aff.)
UC San Diego (of Aff.)
San Diego, USA
rchunduri@ucsd.edu

***Abstract*—Federated learning (FL) is collaborative model training without sharing raw data of individuals, but secure aggregation (SecAgg) is required to prevent leakage of individual updates. Existing SecAgg protocols rely on RSA or elliptic-curve cryptography (ECC), which are vulnerable to quantum attacks. In this project, we propose a post-quantum secure aggregation scheme based on lattice cryptography, using the ML-KEM and ML-DSA primitives. We then integrate our scheme into a standard FL pipeline and evaluate it on the MNIST benchmark. Experimental results show that post-quantum SecAgg achieves nearly identical accuracy, runtime, and communication cost compared to plaintext aggregation, while offering strong security guarantees against quantum adversaries. These findings demonstrate that lattice-based cryptography provides a practical and future-proof solution for secure aggregation in federated learning.**

***Index Terms*—federated learning, secure aggregation, post-quantum cryptography, lattice based cryptography, ML-KEM, ML-DSA**

I. Introduction

Improving the machine learning models in smart devices we use on a daily basis is easier when you have a large dataset. While collecting datasets from users' devices plays a huge role in developing the machine learning model it also raises privacy concerns. This is where the federated machine learning model comes into play. Federated machine learning is a decentralized approach where a model is trained collaboratively using datasets from multiple devices without the data leaving their device. This process involves training the model on individual datasets on devices and then aggregating the results [2].

Security is very crucial in these aggregation protocols as the gradients of the machine learning model on a device being leaked puts the original data at risk. Solutions such as fully homomorphic encryption (FHE) have been studied widely in federated learning as it allows for computations to be done on encrypted data [5]. But as we know, most of these are based on RSA or ECC, they are computationally heavy, and vulnerable to quantum threats. In this paper we conduct a study experimentally on aggregation protocols that implement lattice based cryptography protocols and test their quantum security.

II. Background

In this section we will cover the formal definitions of federated learning, secure aggregation, post quantum cryptography and fully homomorphic encryption.

a) Federated Learning: Federated Learning is a collaborative machine learning approach where the model is trained using datasets from multiple clients by preserving their privacy. Some examples of applications of federated learning are:

- Google's Gboard: The Google Gboard trains a model on each individual device and then sends the results to the server instead of the individual's raw data. These results are masked using cancelling masks before sending to the server and they are aggregated. The masks cancel during the aggregation process. This way, we have a large and diverse dataset to train our models while the individual's privacy is maintained.
- Siri's voice recognition: Similar to Google's Gboard, instead of sending the raw audio/voice data to the server we train a model on the individual's device and send the results to server after encryption. Computations by the server are performed on the encrypted data and the result is decrypted.

b) Secure Aggregation: In the federated learning approach we train a model on each individual's device and then send the masked version of the results to the server. Every pair of clients have a shared secret and the this is used to generate a mask. One of the clients adds this mask to their update and the other client subtracts. This happens for every possible pair of clients. After masking all the updates, the masked updates are aggregated by the server and since the masks cancel the value we get is equivalent to the aggregation of unmasked plaintexts. This process of collecting masked updates from different individual devices to send them to the central server is called secure aggregation. We require our aggregation protocols to be secure as we do not want the individual updates to be exposed during the process of aggregation. The individual updates being exposed puts the individual's data at risk [1].

c) Post Quantum Cryptography: Most of the classical cryptography protocols are based on RSA and ECC which can be broken by quantum computers using Shor's algorithm. However, lattice based cryptography approaches are still very promising to be quantum secure. Some lattice based cryptographic schemes are LWE, Ring-LWE, Module-LWE etc.

- **LWE:** LWE or Learning With Errors is a mathematical problem that serves as a foundation for most post quantum cryptographic schemes. This problem gives us n linear equations in a vector s with n coordinates by adding some error to the equations, where s is the secret key for encryption. While solving for n coordinates when given n equations is an easy job, adding noise makes this a computationally hard problem even for quantum computers.
- **Ring-LWE:** It is a variation of LWE but instead of vectors we use a ring of polynomials which makes homomorphic encryption and key exchange more efficient.
- **Module-LWE:** Module-LWE is a generalization of Ring-LWE that uses modules over polynomial rings. This forms the basis of standardized NIST PQC schemes such as ML-KEM (Kyber) and ML-DSA (Dilithium).

d) Fully Homomorphic Encryption: Fully Homomorphic Encryption or FHE is a cryptographic scheme that allows us to process data and perform computations on encrypted data and decrypt the result. FHE can be used to perform calculations using addition and multiplication on encrypted data, and we can construct more complex operations using addition and multiplication. This is useful with any confidential or sensitive data to first encrypt it and perform computations without the server learning the actual data. Aggregation protocols in federated learning are an application of FHE.

III. Security Goals and Threat Model

The aim of this project is to design an aggregation protocol that remains secure against quantum-capable adversaries. We consider three adversaries:

Server (Malicious) The central server is assumed to be a malicious adversary. It may attempt to learn information about individual client updates even when following the protocol honestly, or it may deviate from the protocol in an effort to manipulate or misuse client data.

Clients (Semi-honest) Clients are assumed to be semi-honest. They execute the protocol as specified in order to protect their own data, but they may attempt to infer more information about the aggregated model or other clients than intended.

Network (Eavesdropping) We assume the communication network is open and subject to passive eavesdropping. Thus, adversaries may intercept transmitted messages, motivating the use of post-quantum secure encryption to ensure confidentiality.

Security Goals. Our protocol aims to achieve:

- *Confidentiality:* Individual client updates must remain hidden from the server, other clients, and eavesdroppers.
- *Correctness:* The final aggregated model must accurately reflect the contributions of all participating clients.
- *Post-quantum security:* All guarantees should hold against adversaries equipped with quantum computing capabilities.

IV. Methodology

We begin by implementing and validating a lattice-based additive encryption scheme on small toy weights to confirm the correctness and efficiency of masking and unmasking operations. We will then integrate this scheme into a federated learning pipeline and apply it to real-world datasets and models such as the MNIST dataset. We evaluate our protocol against existing aggregation approaches by measuring (i) aggregation runtime, (ii) communication cost, and (iii) model accuracy compared to plaintext training.

V. Implementation

I first validated the secure aggregation protocol before applying it to real world datasets such as the MNIST dataset. To validate, I ran a simulation in python using PyTorch for model training and liboqs for post quantum cryptographic primitives. For this phase I used randomly generated data as oy weights. The objective of this phase is to confirm the correctness of masking and aggregation after masking.

A. Simulation Phase

For the simulation phase I used a logistic regression model (TinyLogReg) with a single linear layer mapping a 784-dimensional input to 10 output classes. But instead of using real data for the purpose of simulation I used randomly generated data. After this confirmation, I applied the same protocol on the MNIST dataset.

B. Protocol Structure

Each client has:

- A post quantum key pair for key encapsulation (ML-KEM-768)
- A post quantum signature key pair (Dilithium)
- A local model initialized with server weights and a private data loader.

The server is responsible for coordinating between the training rounds. The protocol has two phases. The first phase or phase 1 involves clients exchanging encapsulated secrets with each other and filling out "inboxes" that contain ciphertexts required for deriving the masks. In the next phase which is phase 2, each of these clients performs a single stochastic gradient descent (SGD) update on its local model and then computes the parameter delta with respect to the global model. This delta is then masked deterministically using integer masks that were derived from the pairwise shared secrets. The reason why we have phase 1 and phase 2 instead of both of them together as a single phase is because all the clients have to finish creating their shared secrets before the aggregation step. So phase one is where we generate the secrets and after that we use phase 2 to train the models and send masked updates to the servers.

C. Masking Mechanism

Symmetric masks for each pair of clients are generated using a pseudorandom generator which gives an expanded version with signed 32-bit integers. Each client adds masks of clients with higher indices than itself and subtracts the masks of clients with indices lower than itself. After all clients finish this the masks exactly cancel out. This guarantees that the sum of the masked updates is the same as the sum of the unmasked updates, meaning the server can aggregate updates from multiple clients without actually looking at each one of them.

D. Aggregation and Verification

The server verifies the signatures on the client payloads and then deserializes masked updates and then aggregates them. To verify the correctness, the code also prints the unmasked average delta. The code also logs the $l-2$ norms of the masked and unmasked aggregates, as well as their difference which remained to be close to zero throughout the simulation. At the end, the global model weights are updated using these averages.

E. Experimental Output

I tried multiple values for number of clients and number of rounds and across all these tests I observed that:

- The $l-2$ norms of masked and unmasked updates were identical
- The global model weights updated consistently for each round
- Final model weights were finite values

We have received the exact same results with close to zero difference norm for the MNIST dataset as well.

VI. Results

A. Baselines

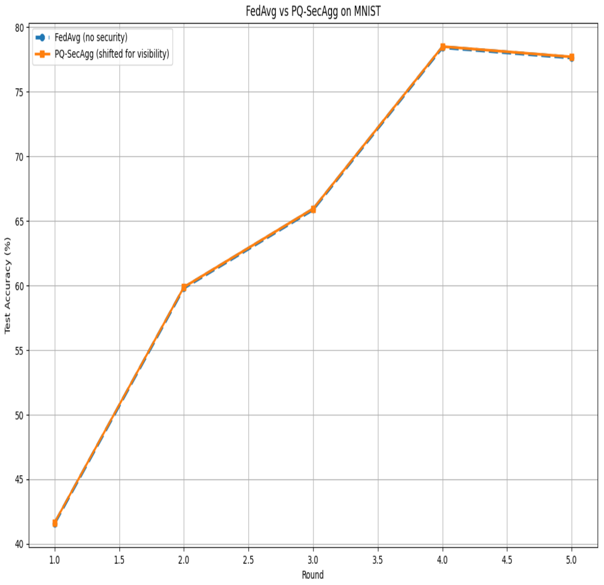
As baselines, I compared this scheme of post quantum secure aggregation against:

- Plaintext aggregation (aggregation without masking or any kind of cryptography)
- The classical SecAgg protocol based on elliptic-curve cryptography (ECC) [Bonawitz et al., CCS 2017], which is cited directly rather than reimplementing it, as it is widely considered standard.

B. Results

Plaintext aggregation versus post quantum secure aggregation showed no difference in cost of communication and model accuracy although the aggregation times differ. As shown in the picture below both graphs of plaintext aggregation and post quantum secure aggregation turned out to be exactly the same for test accuracy vs. number of rounds. For the toy weights and for the MNIST dataset we see in the experiments that both of them have the exact same values for communication cost and model accuracy but different aggregation runtimes.

It is also observed that the accuracy of the model increases with the number of rounds of training, but we don't train too many rounds to avoid overfitting. This can be observed in the recorded plots below. They represent the graphs for test accuracy plaintext aggregation versus post quantum secure aggregation for different values of number of rounds of training.



Number of rounds = 3, model accuracy = 65.9%

C. Analysis

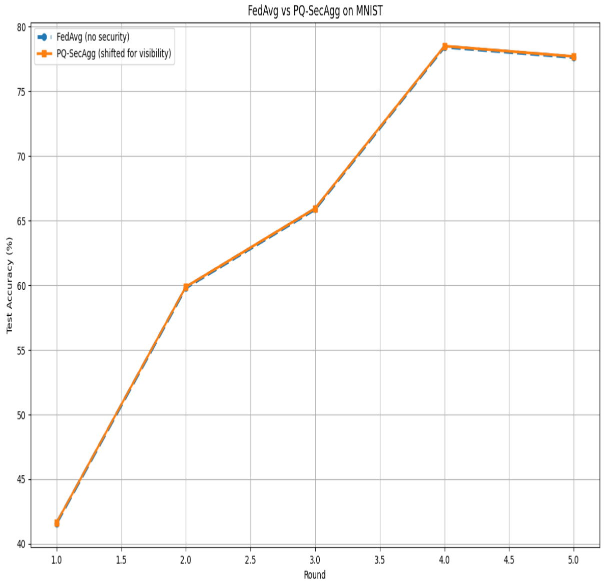
The fact that the delta measurements for both plaintext aggregation and post quantum secure aggregation are exactly the same implies that the masks are cancelling each other and the server aggregates the same value in both cases. The difference in aggregation runtime is negligible even for a larger dataset such as the MNIST, meaning the cost of using post quantum secure aggregation is not much in terms of aggregation runtime.

VII. Related Work

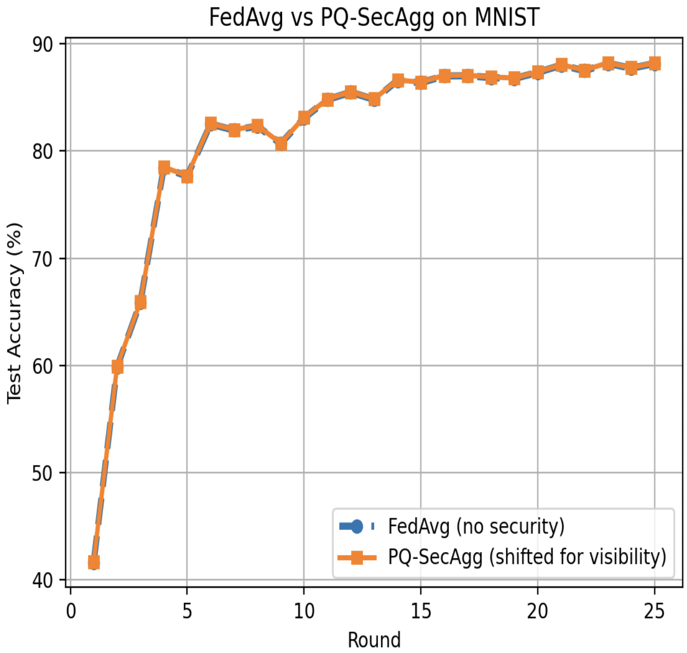
Secure Aggregation for Federated Learning was first introduced by [1]. This paper explained a protocol based on elliptic curve cryptography which was used in Google's Gboard. Fully homomorphic encryption (FHE) can also be studied in the case of federated learning but since it is very computationally heavy secure aggregation can be preferred on large datasets. With the upcoming quantum threats RSA and ECC based protocols no longer guarantee security. In this situation lattice based cryptographic protocols are more promising and that is what this work contributes towards.

VIII. Conclusions

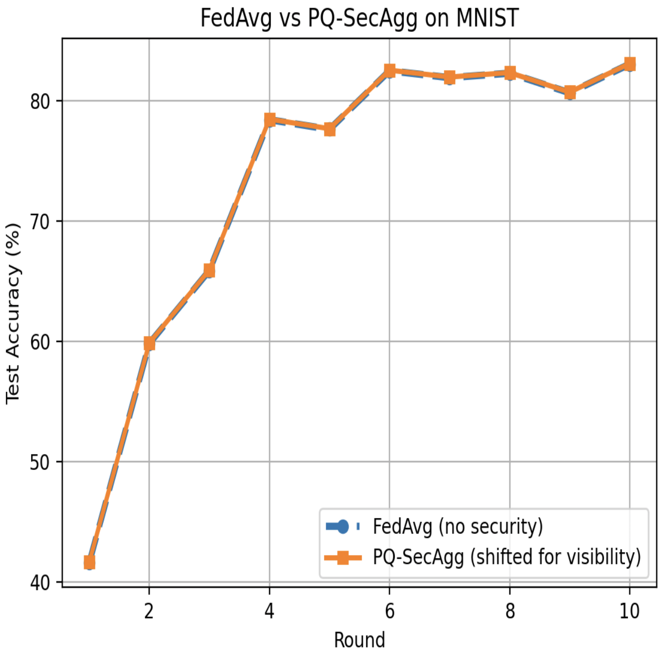
We presented a secure aggregation protocol for federated learning that uses lattice-based cryptography to achieve post-



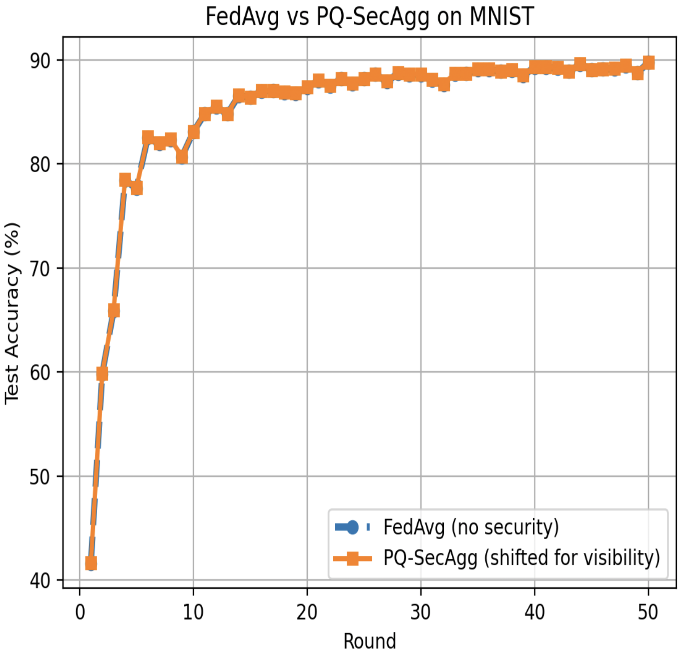
Number of rounds = 5, model accuracy = 77.65%



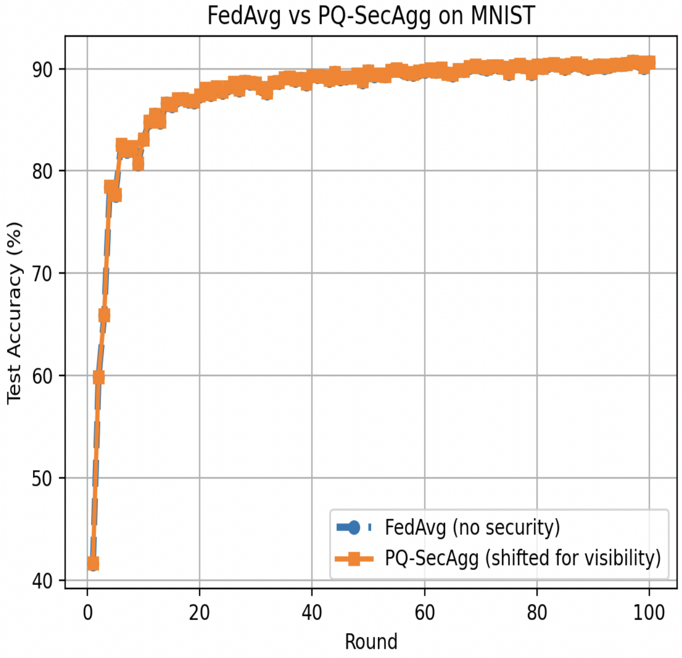
Number of rounds = 25, model accuracy = 88.17%



Number of rounds = 10, model accuracy = 83.08%



Number of rounds = 50, model accuracy = 89.74%



Number of rounds = 100, model accuracy = 90.60%

quantum security. These experiments on MNIST demonstrated that this protocol maintains correctness of aggregation, and maintains its runtime very close to plaintext aggregation, and achieves accuracy comparable to both plaintext and ECC-based secure aggregation. Compared to ECC-SecAgg, our approach provides stronger long-term security guarantees against quantum adversaries, but at the cost of higher communication cost due to larger key sizes.

IX. Discussions and Future Work

Post Quantum Cryptography increases the key sizes and therefore the sizes of ciphertexts, however, the per round cost still remains linear. Some of my future work will be:

- If some of the participants dropout, meaning if they go offline or stop communicating, the system should be able to recover the dropout party's masks and maintain the functions of the aggregation protocol
- Incorporate differential privacy to give stronger privacy guarantees to the individuals.
- Make the protocol more efficient in a way such that all rounds can proceed without waiting for all the clients to finish in the case that some clients are extremely slow.

Acknowledgment

I wish to acknowledge the San Diego Undergraduate Tech Conference (SDUTC) for providing the opportunity to present this research, the open-source community, and the developers of libraries such as PyTorch and liboqs, which provided essential tools for the implementation and experiments in this work.

References

[1] Bonawitz et al., “Practical Secure Aggregation for Federated Learning on User-Held Data,” Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1175–1191, 2017.

[2] Kim Martineau and Cole Stryker, “What is Federated Learning,”, IBM Research Blog, 24-Aug-2022 , Available: https://research.ibm.com/blog/what-is-federated-learning. [Accessed: 10-Aug-2025].

[3] National Institute of Standards and Technology, “NIST releases first 3 finalized post-quantum encryption standards,” NIST News, 13-Aug-2024. [Online]. Available: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards. [Accessed: 30-Aug-2025].

[4] K. Burlachenko, A. Alrowithi, F. Albalawi and P. Richtárik, “Federated Learning is better with Non-Homomorphic Encryption,” arXiv preprint arXiv:2312.02074, 2023. doi: 10.1145/3630048.3630182.

[5] N. Baracaldo, H. Shaul “Federated Learning Meets Homomorphic Encryption,”, IBM Research Blog, 16-Dec-2022 , Available: https://research.ibm.com/blog/federated-learning-homomorphic-encryption. [Accessed: 15-Aug-2025]..

[6] S. Datta et al., “Federated Learning With Quantum Computing and Fully Homomorphic Encryption: A Novel Computing Paradigm Shift in Privacy-Preserving ML,” arXiv preprint arXiv:2409.11430v3, 2024