SECURE COMMUNICATION MODEL FOR QUANTUM FEDERATED LEARNING: A PROOF OF CONCEPT

Anonymous authors

ABSTRACT

We design a model of Post Quantum Cryptography (PQC) Quantum Federated Learning (QFL). We develop a proof of concept with a dynamic server selection and study convergence and security conditions.

1 INTRODUCTION

Quantum Federated Learning (QFL) is an emerging area with several studies in recent years; see (Larasati et al.) and references therein. Most QFL works focus on optimization (Kaewpuang et al.; Yang et al.; Yun et al.; Xia & Li), security (Xia et al.; Zhang et al.) and implementation (Qi; Yamany et al.; Abbas et al.; Huang et al.; Chehimi & Saad). However, post-quantum secure QFL is poorly studied in the literature.

We develop a preliminary study with a proof of concept model of post-quantum secure QFL. The two *main contributions* are as follows: i) we develop a novel PQC-QFL model, ii) we implement the proof of concept and evaluate the proposed model theoretically and experimentally.
The uploading of learned model parameters, m_i, from i clients to the server can be poisoned by an adversary. To protect against unauthorized access during uploads/downloads, post-quantum secure schemes are essential. We consider a client c_i with signature scheme $S_i(sk_i, pk_i)$ which is used to sign the parameters m_i as, $\delta_i \leftarrow S_i^{sign}(m_i, sk_i)$, and are verified as $\{m_i, pk_i, \delta_i\}$ by the server.

Algorithm 1 Post Quantum Secure Communication in QFL

Initialization: n devices in $device_list = \{d_1, d_2, ..., d_n\}$.
Generate keys (pk, sk) for each device.
Randomly select a device $d_s = d_i$ for server role.
procedure DEVICETASK(pk, sk)
  for $device$ in $device_list$ **do**
    Train local params w_d.
    Sign params w_d with PQC-Scheme.
    $\sigma_d \leftarrow Sign(p_d)$.
    Send $\{\sigma_d, w_d, pk_d\}$ to server.
  end for
end procedure
procedure SERVERTASK(pk, sk)
  Initialization: $param_list = []$
  Server d_s receives $\{w, \sigma, pk\}$ from all devices.
  for device d_i in $devices_list$ **do**
    d_s verifies each σ_i.
    if Verification is true **then**
      $param_list.append(w_i)$
    else
      w_i is excluded.
    end if
  end for
  Perform fedAveraging.
  Send back global params w_g to devices.
end procedure

2 PROPOSED PQC-QFL MODEL

We present details of the proposed PQC-QFL model in Algorithm 1. For n devices, a server device d_s is selected dynamically (e.g., random server selection). First, all devices train the models locally. Once the local models are learned, each device d_i signs with its private key sk_i to generate the signature σ_i. Each device sends this information to the server d_s and its public key pk_i. Upon receiving this information, the server verifies and filters the compromised models. Eventually, it performs the model aggregation to generate a global model and sends it back to the devices. The PQC-QFL design is grounded in the convergence and security conditions as follows: i) each device trains local Stochastic Gradient Descent (SGD) with T iterations, and PQC scheme with security level λ, the Algorithm 1 converges at a rate of $n * \{O(1/\sqrt{T}) + O(1/\lambda^2)\}$; ii) with Algorithm 1, the security level is SUF-CMA secure (provided by the Dilithium (Ducas et al.) signature scheme). See the appendix for details. More importantly, with the proposed model, we have demonstrated that the resilience of the proposed dynamic server is always higher.

All the fundamental assumptions, such as convexity, unbiasedness, bounded variance, etc., made during the design of PQC-QFL are discussed in detail in the appendix. With the proposed PQC-QFL, we have the following two theorems.

Theorem 1 (Resilience) *The occurrence of single-point failure with Algorithm 1 (e.g., random-server architecture) is always less than in the current QFL.*

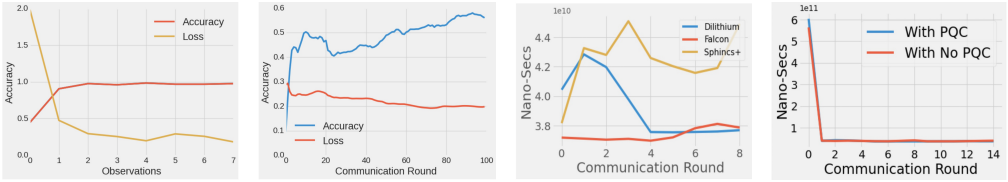
Theorem 2 (Convergence) *Given n devices, each device trains local stochastic gradient descent (SGD) with T iterations and employs PQC at security level λ. Then Algorithm 1 converges with a rate of $n * \{O(1/\sqrt{T}) + O(1/\lambda^2)\}$.*

The proofs are deferred to the appendix.

2.1 Implementation and Evaluation

We developed our implementation and experimental framework by extending the framework presented by Zhao et al. The post-quantum cryptography, liboqs [1] and liboqs-python [2] libraries are integrated with the codebase. For most experiments, we employ the Dilithium signature scheme. MNIST dataset is shared between clients by following a cycle-m structure where each client will only have a certain number of label classes.

As shown in Figure 1(a), a random server selection performs far better, which is visible after 2 epochs in the training. The observations are recorded for every 4 times per epoch. This implies that with PQC-QFL, quantum machine learning performs very well in terms of training. Figure 1(b), shows the performance on the test set done by the server with the new global model. After the 100 communication round, the test accuracy is around 55%, which is poor in comparison to classical machine learning. This demonstrates that QFL suffers heavily over non-IID datasets. Figure 1(d)



(a) Training Accuracy (b) Test Accuracy (c) PQC Schemes (d) Comm-Time

Figure 1: Accuracy, Performance, and Communication time of PQC-QFL.

shows the difference in the impact of using PQC schemes in signing and verification. Some added tasks of the signature scheme would add some delay to the system's overall performance. However, as shown in Figure 1(d), we cannot see much noticeable adverse impact by employing the PQC schemes in terms of nanoseconds. Figure 1(c) shows the comparison of this effect on Dilithium with other schemes such as Falcon (Fouque et al., 2018) and SPINCHS+ (Bernstein et al., 2019). Observe that Dilithium and Falcon perform in a similar fashion. Whereas SPHINCS+ exhibits increased delay compared to Dilithium and Falcon, which requires further investigations and in-depth understanding and is left for future work.

3 Conclusion

We have developed a novel preliminary model and implementation of PQC-QFL. We performed extensive analysis and presented results that assert the practicality and suitability of PQC schemes in the QFL. The proof of concept experiment also involved the removal of a fixed central server approach where any participating device can be selected to perform as a central server in contrast to a dedicated server in the traditional QFL.

[1] https://github.com/open-quantum-safe/liboqs.git
[2] https://github.com/open-quantum-safe/liboqs-python.git

REFERENCES

Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. 1(6):403–409. ISSN 2662-8457. doi: 10.1038/s43588-021-00084-1. URL http://arxiv.org/abs/2011.00027. Comment: 25 pages, 10 figures.

Daniel J Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The sphincs+ signature framework. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 2129–2146, 2019.

Mahdi Chehimi and Walid Saad. Quantum Federated Learning with Quantum Data. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8617–8621. doi: 10.1109/ICASSP43922.2022.9746622.

Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. pp. 238–268. ISSN 2569-2925. doi: 10.46586/tches.v2018.i1.238-268. URL https://tches.iacr.org/index.php/TCHES/article/view/839.

Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. Falcon: Fast-fourier lattice-based compact signatures over ntru. *Submission to the NIST's post-quantum cryptography standardization process*, 36(5), 2018.

Rui Huang, Xiaoqing Tan, and Qingshan Xu. Quantum Federated Learning With Decentralized Data. 28:1–10. ISSN 1558-4542. doi: 10.1109/JSTQE.2022.3170150.

Rakpong Kaewpuang, Minrui Xu, Dusit Niyato, Han Yu, Zehui Xiong, and Xuemin Sherman Shen. Adaptive Resource Allocation in Quantum Key Distribution (QKD) for Federated Learning. URL http://arxiv.org/abs/2208.11270. Comment: 6 pages, 6 figures, and a conference. arXiv admin note: text overlap with arXiv:2208.08009.

Harashta Tatimma Larasati, Muhammad Firdaus, and Howon Kim. Quantum Federated Learning: Remarks and Challenges. In *2022 IEEE 9th International Conference on Cyber Security and Cloud Computing (CSCloud)/2022 IEEE 8th International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 1–5. doi: 10.1109/CSCloud-EdgeCom54986.2022.00010.

Jun Qi. Federated Quantum Natural Gradient Descent for Quantum Federated Learning. URL http://arxiv.org/abs/2209.00564. Comment: Published parts of book in Federated Learning.

Qi Xia and Qun Li. QuantumFed: A Federated Learning Framework for Collaborative Quantum Training. In *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. doi: 10.1109/GLOBECOM46510.2021.9685012.

Qi Xia, Zeyi Tao, and Qun Li. Defending Against Byzantine Attacks in Quantum Federated Learning. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 145–152. doi: 10.1109/MSN53354.2021.00035.

Waleed Yamany, Nour Moustafa, and Benjamin Turnbull. OQFL: An Optimized Quantum-Based Federated Learning Framework for Defending Against Adversarial Attacks in Intelligent Transportation Systems. pp. 1–11. ISSN 1558-0016. doi: 10.1109/TITS.2021.3130906.

Chao-Han Huck Yang, Jun Qi, Samuel Yen-Chi Chen, Pin-Yu Chen, Sabato Marco Siniscalchi, Xiaoli Ma, and Chin-Hui Lee. Decentralizing Feature Extraction with Quantum Convolutional Neural Network for Automatic Speech Recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6523–6527. doi: 10.1109/ICASSP39728.2021.9413453.

Won Joon Yun, Jae Pyoung Kim, Soyi Jung, Jihong Park, Mehdi Bennis, and Joongheon Kim. Slimmable Quantum Federated Learning.

Yichi Zhang, Chao Zhang, Cai Zhang, Lixin Fan, Bei Zeng, and Qiang Yang. Federated Learning with Quantum Secure Aggregation. URL http://arxiv.org/abs/2207.07444.

Haimeng Zhao. Exact Decomposition of Quantum Channels for Non-IID Quantum Federated Learning. URL http://arxiv.org/abs/2209.00768. Comment: 6 pages excluding appendices and references. Code available at https://github.com/JasonZHM/quantum-fed-infer.

APPENDIX: THEORETICAL ANALYSIS

Theorem 1. *The occurrence of single-point failure with Algorithm 1 (e.g., random-server architecture) is always less than in current QFL.*

PROOF. With algorithm 1, the server role is selected randomly or based on the reputation as required. Thus, the server keeps changing with time. For an adversary A, to attack fixed-server $fixedServer$, as identity is the same every time, the probability of adversary accuracy attacking the server is 1. However, with a random or server selection approach, the probability of an adversary A finding the right device $randomServer$ to attack is almost always < 1. Thus, we can write,

$$P(A \rightarrow fixedServer) > P(A \rightarrow randomServer).$$

□

CONVERGENCE ANALYSIS

To analyze the convergence of the proposed QFL algorithm 1, let's follow standard convergence analysis for SGD with the following assumptions.

Assumption 1 (Smoothness) *Loss function for each worker $L_i(x)$ is L-smooth i.e. $\forall(x, y) \in \mathbb{R}^d$,*

$$\mid L_i(x) - L_i(y) - \langle \Delta L_i(y), x - y \rangle \mid \leq \frac{L}{2} \mid\mid x - y \mid\mid^2 .$$

where, $\Delta L_i(y)$ is the gradient of L at point y and $\mid\mid x - y \mid\mid$ is the Euclidean distance between x and y. Also, L is the Lipschitz constant which is a mathematical concept to measure how fast a function can change. L-smooth loss function means its slope doesn't change too rapidly.

Assumption 2 (Convex) *The loss function for each worker $L_i(x)$ is $Convex$.*

In optimization, it is essential to find the global minimum and guarantee the convergence and efficiency of the algorithms.

Assumption 3 (Unbiasedness and Bounded Variance) *Gradients hold properties of unbiasedness and bounded variance.*

To analyze the convergence properties of the QFL algorithm, let us consider the following optimization problem:

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} L_i(\boldsymbol{w}), \tag{1}$$

where $\boldsymbol{w}$ is the model parameter, $L_i(\boldsymbol{w})$ is the loss function for the i-th device, and n is the total number of devices.

The stochastic gradient descent algorithm for QFL involves the following update rule:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}t - \eta_t \sum i = 1^n \nabla L_i(\boldsymbol{w}_t), \tag{2}$$

where η_t is the learning rate at iteration t.

Remark 1 *Standard SGD converges to a stationary point of the objective function with a rate of $O(1/\sqrt{t})$, where t is the total number of iterations.*

Under the assumptions, we can show that SGD converges to a stationary point of the objective function with a rate of $O(1/\sqrt{T})$, where T is the total number of iterations.

We show that the iterates $\boldsymbol{w}_t$ converge to a limit point, which is a stationary point of the objective function. We can use the following inequality to bound the expected distance between the iterates and the limit point:

$$\mathbb{E}[|\boldsymbol{w}_t - \boldsymbol{w}^*|^2] \leq \frac{2B^2}{\eta_t t} \tag{3}$$

where $\boldsymbol{w}^*$ is the limit point, B is an upper bound on the norm of the gradients, and η_t is the learning rate at iteration t.

Using this inequality, we can show that the expected objective function value converges to the optimal value at a rate of $O(1/\sqrt{T})$. Specifically, we have:

$$\mathbb{E}[f(\boldsymbol{w}_t)] - f(\boldsymbol{w}^*) \leq \frac{2B^2 L}{\sqrt{T}}, \tag{4}$$

where $f(\boldsymbol{w})$ is the objective function, and L is the Lipschitz constant of the gradients.

Therefore, we have shown that SGD converges to a stationary point of the objective function with a rate of $O(1/\sqrt{T})$, under suitable assumptions on the loss functions, gradients, and learning rate.

CONVERGENCE CONDITIONS OF ALGORITHM 1

Remark 2 *PQC scheme Dilithium has a time complexity of $O(1/\lambda^2)$ where λ is the security level of the scheme.*

With security level λ, the key generation happens only once which takes time as $keyTime()$. For signing, the time it takes is $signTime()$ whereas, for verification, the time it takes will be $verificationTime()$. The performance of Dilithium depends on many factors like hardware implementation, message size, etc. Here,

$$\{keyTime() + signTime() + verificationTime()\} \propto \lambda$$

Theorem 2 *Consider n devices, each device trains local SGD with T iterations, and PQC scheme with security level λ. Under this setting, the algorithm 1 converges with a rate of $n * \{O(1/\sqrt{T}) + O(1/\lambda^2)\}$.*

PROOF. From remark 1, we get the SGD convergence rate as: $O(1/\sqrt{T})$. Whereas with the added time complexity of the PQC scheme for signing, verifying, and key generating, the proposed scheme converges at the rate of $O(1/\sqrt{T}) + O(1/\lambda^2)$. Where PQC time complexity is $O(1/\lambda^2)$. Thus, the total time delay for convergence rate would be at least $\leq n * \{O(1/\sqrt{T}) + O(1/\lambda^2)\}$ for n devices. □

SECURITY CONDITIONS

For each communication between the trainer device and server device by the following algorithm 1, the security level is SUF-CMA secure which is provided by the Dilithium signature scheme.

For any digital signature scheme, the standard notion is UF-CMA security which refers to security under chosen message attacks. The security model involves an adversary with an accessible public key that can be used to access the signing oracle to sign other messages. The adversary tries to get a valid signature for any new message. Dilithium signature scheme is SUF-CMA secure i.e. Strong Unforgeability under Chosen Message Attacks. It is based on the hardness of standard lattice problems.

With Dilithium implementation, the communication between server and client becomes SUF-CMA secure i.e. for quantum random oracle H, adversary $\mathcal{A}$ has the advantage of breaking the communication which can be represented as (Ducas et al.):

$$Adv_{Dilithium}^{SUF-CMA}(\mathcal{A}) \leq Adv_{k,l,D}^{MLWE}(\mathcal{B}) + Adv_{H,k,l+1,\zeta}^{SelfTargetMSIS}(\mathcal{C}) + Adv_{k,l,\zeta'}^{MSIS}(\mathcal{D}) + 2^{-254} \quad (5)$$

for uniform distribution D over $\mathcal{S}_n$. Also,

$$\zeta = max\{\gamma_1 - \beta, 2\gamma_2 + 1 + 2^{d-1} * 60\} \leq 4\gamma_2$$
$$\zeta' = max\{2(\gamma_1 - \beta), 4\gamma_2 + 2\} \leq 4\gamma_2 + 2$$

In Eqn. 5, the assumptions used are:

1. $Adv_{m,k,D}^{MLWE}(\mathcal{B})$ is an advantage of algorithm B in solving MLWE (Module variant of Learning with Error) problem for integers m, k with probability distribution $D : R_q \rightarrow [0, 1]$. This assumption is required to protect against key recovery.
2. $Adv_{H,k,l+1,\zeta}^{SelfTargetMSIS}(\mathcal{C})$ refers to advantage of algorithm C in solving SelfTargetMSIS problem.It is based on combined hardness of MSIS and the hash function H. SelfTargetMSIS assumption provides the basis for new message forgery.
3. MSIS assumption is needed for strong unforgeability.

The notations used are m, k are integers, D is a probability distribution, $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ are algorithms, H is a cryptographic hash function, 2^{-254} is a small constant representing negligible probability of occurrence of some rare event.

Dilithium is simple to implement securely, stateless, and acceptable combined size of public key and signature (Ducas et al.).