Blockchain-Enabled Verifiable Collaborative Learning for Industrial IoT

Jayasree Sengupta
Indian Institute of Engineering Science and Technology, Shibpur, India
Email:jayasree202@gmail.com

Sushmita Ruj
University of New South Wales, Sydney, Australia
Email: sushmita.ruj@unsw.edu.au

Sipra Das Bit
Indian Institute of Engineering Science and Technology, Shibpur, India
Email: sdasbit@yahoo.co.in

I. INTRODUCTION

With the commencement of the Industrial Internet of Things (IIoT), followed by Industry 4.0, the amount of data generated by the connected components has grown drastically. This opens up new possibilities for effectively utilizing such data to improve the operational services of the industries as well as provide intelligent customer support. To gain knowledge, industries are inclined toward developing novel machine learning (ML) approaches for processing or modeling such data. Hence, it is extremely important to collect training data distributed across enterprises or industries to develop a highly trained model [1]. However, training ML models in such a distributed setting introduces additional security concerns as industries are unwilling to share their private and sensitive information with third-party servers. As a result, collaborative learning [2] has recently been introduced, which allows data owners to collectively train a globally shared model by leveraging their private inputs without exporting them to anyexternal server.

Collaborative learning is a promising new paradigm where each device contributes to the global model update by sharing its local model with the cloud/aggregation server without transmitting the private data through the network [3]. Despite the advantages of collaborative learning, a couple of primary concerns are poisoning attacks and the vulnerability of locally trained models to information leakage. In a poisoning attack, a malicious entity contributes adversarial updates to the shared model parameters. In contrast, it tries to infer the properties of the sensitive training dataset of neighboring data owners in case of an information leakage attack [4]. Another potential concern is input data privacy, wherein an attacker performs model-inversion attacks to restore the original training dataset; hence it is essential to keep the local models private. Apart from the privacy issues, the cloud/aggregation server may also behave maliciously by returning forged aggregated results to the devices to impact model updates. Under worse circumstances, a server may also return carefully crafted results to the devices to analyze the uploaded data's statistical characteristics and unintentionally provoke the devices to expose sensitive information. Additionally, since the IIoT devices are typically low-powered, they may become unavailable or dead (i.e., device dropout may happen) at any time in the network. Lastly, the involved parties (e.g., IIoT devices, servers, etc.) may also collude to launch one or more of the attacks, as mentioned earlier, simultaneously.

The recent state-of-the-art works [1–3] have focused on handling one or more of the above concerns. However, they have mainly been addressed in an isolated manner. For example, defense against poisoning and information leakage attacks has been discussed separately in the literature using various techniques like secure aggregation, differential privacy, etc. A few such works [5, 6] have also

introduced fog as a middleware platform between devices (i.e., data owners) and cloud/aggregation servers to bring computation to the edge further and reduce trust dependence on the cloud. However, to the best of our knowledge, no substantial work has solved all of the concerns mentioned above concurrently. The latest works [4, 7] have also proposed fully decentralized blockchain frameworks with innovative consensus mechanisms to eliminate the concept of centralized servers alongside addressing some of the aforementioned concerns for a collaborative learning setup. However, industries or companies may not always be willing to design their customized blockchain platform with integrated consensus mechanisms. Further, the attack vectors on such newly designed consensus mechanisms are also unknown, unlike the traditional ones. Therefore, it may instead be more suitable to utilize the benefits of an already existing permissioned or permissionless blockchain platform in such real-life application scenarios. Thus, in our work, we focus on integrating a permissionless blockchain platform as the backbone of the collaborative learning setup.

II. BLOCKCHAIN: A PROMISING SOLUTION

A blockchain is a tamper-resistant, immutable, distributed verifiable ledger. Information stored in a blockchain is made up of a chain of blocks where each block consists of a series of transactions. The blocks are typically hash-linked so that if a transaction is modified in one block, it has to be altered in all the subsequent blocks [8]. Popular blockchain platforms like Ethereum support Turing-complete languages to write smart contracts. A Smart Contract is a self-enforcing piece of a computer program that can be used to formalize simple agreements between two parties and control the transfer of digital currencies (i.e., cryptocurrency) or assets between them [9]. Once a contract is deployed, its execution can be triggered via transactions processed by miners, who are special nodes responsible for validating and adding transactions to blocks [8]. Moreover, in smart contracts, apart from exchanging cryptocurrencies, programs can be written to execute operations such as access policy verification. Blockchain is chosen as the fundamental backbone for this work because it is capable enough to host distributed applications with the help of smart contracts. Moreover, blockchain keeps a record of all the necessary transactions occurring in the network and thereby maintains transparency. It also acts as a trusted intermediary and resolves any disputes by performing payment settlements. This motivates us to integrate blockchain into the collaborative learning setup to utilize the benefits of the distributed nature of blockchain. Apart from that, by introducing blockchain, we can also defend against the said attacks (like poisoning attacks). Lastly, blockchain also allows us to efficiently handle disputes and collusion cases by designing smart contracts and incentivizing the parties involved.

III. OUR PROPOSED SCHEME

The basic idea of our proposed scheme is to train a regression model (i.e., linear/logistic regression) in a distributed fashion where the IIoT devices, fog nodes, and the cloud collaboratively execute a global gradient computation. Thethree major aims of the scheme are as follows:

- Prevent information leakage attacks by keeping the trainingdata of each IIoT device private and on-premise.
- Prevent poisoning attacks by verifying the local modelupdates sent out by each IIoT device.
- Prevent colluding parties from making sufficient gains.

Our proposed scheme tries to achieve the aforementioned goals while converging to the optimal global model. Initially, we use polynomial commitment [10] along with verification to defend against poisoning attacks. Polynomial commitment takes an input and maps it to a point on the elliptic curve. Later, Shamir's Secret Sharing [11], coupled with additive homomorphism, was used to blind the locally trained model updates from the IIoT devices before sending them out to the fog nodes. Since t shares are enough to reconstruct the secret, therefore Shamir's secret sharing also ensures robustness to IIoT device dropout in the network up to a certain specified threshold. Finally, a secure aggregation scheme based on the previously used polynomial commitment is employed to achieve verifiable aggregation of the global model. Our scheme also deploys smart contracts on the blockchain to act as an enforcer of rules. We have deployed a Contract (**S**) to maintain fairness and manage all necessary operations. Additionally, in case of any dispute, the Turing-completeness (i.e., the ability to simulate any calculation) of the blockchain platform has been used to codify all necessary actions. Further, the native currency of blockchain is used to control and distribute incentives. The immutable property of blockchain has been used to record all data and control message exchanges in the system to provide a transparent infrastructure.

Working Principle: Each IIoT device holds a local dataset and its corresponding model parameter, which gets updated in each round of iteration. Each dataset is a $m_j \times k$ matrix representing m_j training samples with k features where m_j can vary from device to device. The model parameter is a matrix of coefficients with l number of output classes. In our proposed scheme, $M = \sum_{j=1}^{N} m_j$ (where N is the total number of IIoT devices in the system) and l are publicly known by every participant, whereas k is private and is only known to the corresponding data owner. A fog-based two-tier clustered architecture forms the backbone for our proposed scheme. Figure 1 shows the workflow of the proposed scheme, which consists of the following steps:

- ***System Setup:*** Given two groups, G_1 and G_T, of large prime order q with generators g_1 and g_2, respectively, there exists a bilinear pairing defined as a map $\hat{e}: G_1 \times G_1 \rightarrow G_T$ which would be used for polynomial commitments. A Trusted Authority (TA) generates (pk_i, sk_i) key-pair for each fog node F_i. Lastly, it also generates a collision-resistant one-way hash function H: $\{0, 1\}^* \rightarrow \{0, 1\}^n$.
- ***Local Training:*** Each IIoT device D_{ij} feeds its own training dataset and the model parameter coefficient matrix as inputs at each round of iteration. The local training is performed over these two input parameters to generate local gradient g_{ij}. To keep the local gradient private, the IIoT device calls contract **S** to publish a random differentially private noise η_{ij} sampled from a normal distribution. The device then masks its gradient g_{ij} using this noise and sends a 3- tuple packet consisting of the masked update ($g_{ij} + \eta_{ij}$), a commitment to the unmasked update, and a commitment to the noise to the nearest online fog node F_i.

  NOTE: When a malicious IIoT device wants to add a poisonous update g'_{ij} to the global model, it also needs to perturb the noise η_{ij} to make the masked update ($g'_{ij} + \eta'_{ij}$) look like a legitimate one. Since, in our case, the noise is generated by the smart contract, which is tamper-resistant, such malicious activities are by default prevented.
- ***Verification by Fog:*** When a F_i receives this 3-tuple packet from an IIoT device, it first computes a commitment over the received masked update. Then it proceeds to check whether this computed

commitment is consistent with the received commitments to the noise and the unmasked update. The consistency checking is performed using the homomorphic property of commitments. On successful verification, F_i is assured that the received masked update ($g_{ij} + \eta_{ij}$) was indeed computed using the received commitments to the noise and the unmasked update otherwise, F_i discards the 3- tuple packet. When a F_i receives a considerable number of such masked updates, it proceeds to select the best updates using statistical methods like mean and standard deviation. The top n such IIoT devices with the best updates are selected as legitimate devices while the rest are rejected. Only such legitimate devices will be allowed to contribute their model updates for the final aggregation of the global model. Hence, F_i signs the commitment to the unmasked update for the selected devices using its private key. It then sends the signed update back to the concerned IIoT devices.

- ***Secret Sharing:*** When each legitimate D_{ij} receives the signed update, it first performs signature verification to verify the authenticity of the fog node F_i. Next, it proceeds to split g_{ij} into n shares [where n is the number of IIoT devices selected as legitimate in the previous step] as per Shamir's Secret Sharing scheme [11]. It also computes a witness wit_{in} respective to each share of g_{ij}. These witnesses will allow the corresponding IIoT devices to verify that the secret share was indeed computed over the received verified commitment. Finally, each D_{ij} distributes a 3-tuple packet consisting of the secret share, the associated witness, and the signature received from F_i in the previous step to the other corresponding D_{ij}. Each $D_{ij,}$ on receiving this 3-tuple packet, does the following:

(a) It first verifies the signature to ensure that the received update is from a legitimate node. If the verificationfails, it discards the 3-tuple packet; else, it proceeds to the next step.
(b) Next, it utilizes the witness to verify that the received secret share is a part of the verified commitment. If the verification fails, it means that either a malicious node has sent its secret share or a legitimate node has sent a malicious secret share. In both cases, the node is declared malicious and is thereby penalized. Onsuccessful verification, D_{ij} proceeds to the next step.
(c) Once each D_{ij} receives sufficient secret shares, it then locally adds the received shares to product $S_{D_{ij}}$.
(d) Computes the hash of $S_{D_{ij}}$ and stores it as a transaction in the blockchain.
(e) Finally, each D_{ij} sends $S_{D_{ij}}$ to its nearest online fog node F_i.

- ***Computation by Fog***: Each fog node F_i receives $S_{D_{i1}}$, $S_{D_{i2}}$, . . . , $S_{D_{in}}$ values from n IIoT devices lying within its communication range. F_i then calls contract S to verify the integrity of the received $S_{D_{ij}}$ using its hash value stored in the blockchain. If the verification fails, F_i concludes that $S_{D_{ij}}$ has either been sent incorrectly or tampered midway, and it thus discards this $S_{D_{ij}}$ value. Otherwise, F_i is assured that the integrity of the received $S_{D_{ij}}$ is preserved. Once F_i verifies a sufficient number of $S_{D_{ij}}$ values, it proceeds to compute the cumulative gradient ci using the reconstruction mechanism described in Shamir's Secret Share [11]. Here, even if some IIoT devices are offline or busy during this stage and don't send their share of $S_{D_{ij}}$, the reconstruction is still feasible given at least t IIoT devices participate. This operation basically computes the summation part of the Gradient Descent algorithm [2]. Each F_i then stores the commitment, i.e., $COM(c_i)$ and the hash of c_i as a transaction in the blockchain. Lastly, it forwards c_i to the cloud.

- ***Data Aggregation:*** When the Cloud receives c_i values from the fog nodes $F_1, \ldots, F_m$, it calls contract **S** to verify the integrity of the received c_i using its hash value stored in the blockchain. If the verification fails, the cloud concludes that c_i has either been sent incorrectly or tampered midway, and it thus discards this c_i value. Otherwise, the cloud is assured that the integrity of the received c_i is preserved. Once cloud verifies a sufficient number of $S_{D_{ij}}$ values, it generates $c = c_1 + c_2 + \ldots + c_m$ and sends a c back to each F_i.

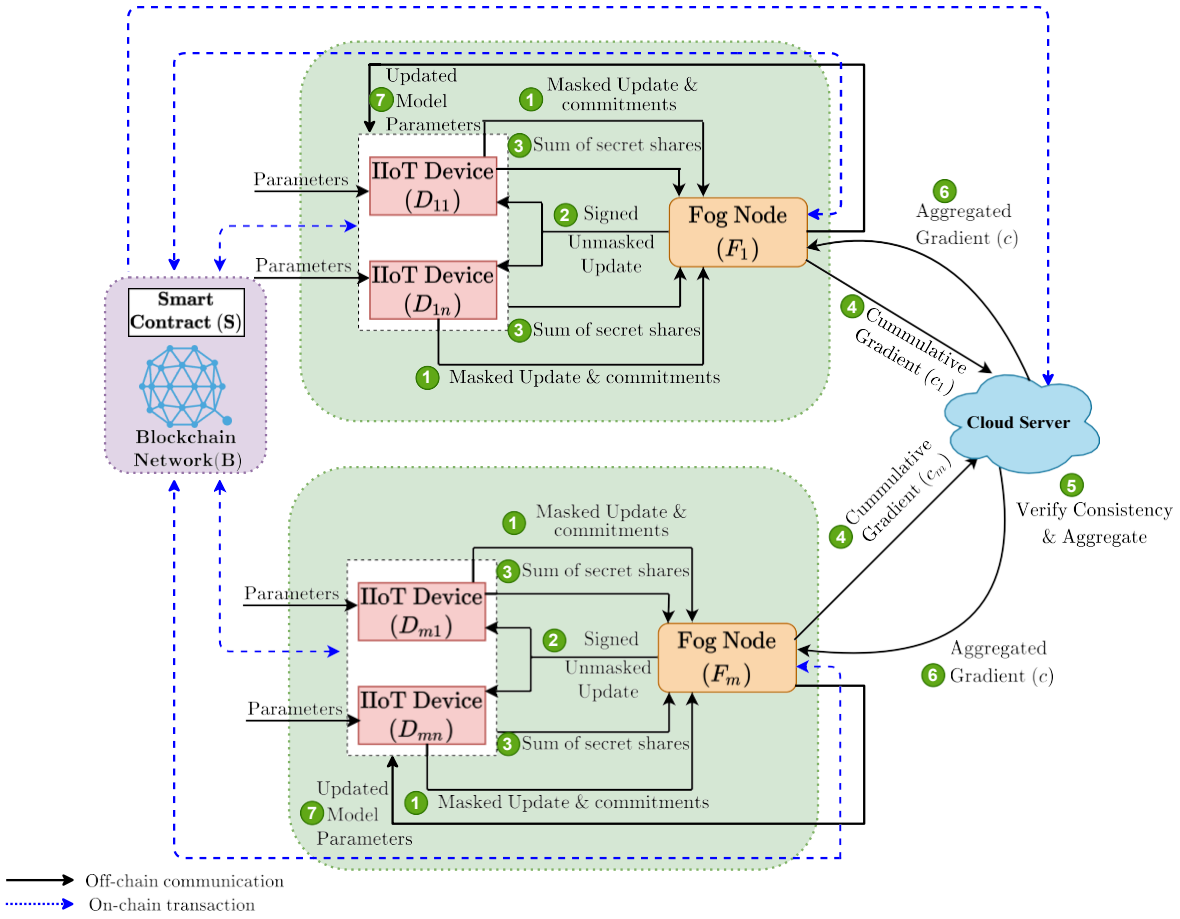

Fig. 1. *Workflow of our Proposed Scheme*

- ***Result Verification:*** Each F_i on receiving c from the cloud computes commitment, i.e., COM © and calls contract **S,** to verify whether the commitment to the received c is consistent with the commitments to c_i stored previously by each individual F_i in the blockchain.

$$COM(c) = \prod_{i=1}^{m} COM(c_i)$$

If this equality holds, then F_i proceeds to the next step. Otherwise, it means that the cloud has behaved maliciously while aggregating the global model. Hence, contract **S** rejects the result and penalizes the cloud for misconduct.

- ***Update Model Parameters:*** On acceptance of the result, F_i computes the updated model parameter based on the latest trained global model. F_i then sends back this updated model parameter to each device $D_{ij,}$ who updates its respective model parameter which is then used for the next training iteration.

From the above discussion, it is evident that by leveraging the immutability property of blockchain, we have been able to ensure the verifiability of the data exchanged in the system. Further, the introduction of smart contracts has also helped us prevent the players' malicious activity. This is because smart contracts are tamper-resistant and hence cannot be mischievously modified to serve' ne's purpose. Lastly, through the use of smart contracts coupled with the inherent cryptocurrency of blockchain, we have also been able to maintain fairness by penalizing the malicious entities as and when required. Thus, we can rightfully claim that we have been able to achieve additional security by introducing blockchain as the inherent backbone.

IV. Conclusion and Future Work

We have presented a blockchain-enabled collaborative learning setup suitable for Industrial IoT scenarios. Our proposed scheme uses the traditional permissionless blockchain as the backbone of the infrastructure design. By leveraging the benefits of a smart contract-enabled blockchain platform, we have ensured additional security (e.g., verifiability). The proposed scheme provides defense poisoning and information leakage attacks and presents malicious activities from the involved parties (i.e., device, fog, and cloud). Apart from that, the scheme also handles collusion cases. In the future, we plan to provide a detailed algorithmic construction of the proposed scheme along with its security analysis. We would also develop a prototype in Ethereum to establish the validity of our scheme.

REFERENCES

[1] Kalikinkar Mandal and Guang Gong. PrivFL: Practical Privacy- Preserving Federated Regressions on High-Dimensional Data over Mobile Networks. In *Proceedings of the 2019 ACM SIGSAC Conferenceon Cloud Computing Security Workshop*, page 57–68, 2019.

[2] Yanjun Zhang, Guangdong Bai, Xue Li, Caitlin Curtis, Chen Chen, and Ryan K. L. Ko. PrivColl: Practical Privacy-Preserving Collaborative Machine Learning. In *Computer Security – ESORICS*, pages 399–418, 2020.

[3] Beongjun Choi, Jy-yong Sohn, Dong-Jun Han, and Jaekyun Moon. Communication-Computation Efficient Secure Aggregation for Federated learning. *CoRR*, abs/2012.05433, 2020.

[4] Muhammad Shayan, Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. Biscotti: A Blockchain System for Private and Secure Federated Learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1513–1525, 2021.

[5] Yeting Guo, Fang Liu, Zhiping Cai, Li Chen, and Nong Xiao. FEEL: A Federated Edge Learning System for Efficient and Privacy-Preserving Mobile Healthcare. In *49th International Conference on Parallel Processing - ICPP*, 2020.

[6] Chunyi Zhou, Anmin Fu, Shui Yu, Wei Yang, Huaqun Wang, and Yuqing Zhang. Privacy-Preserving Federated Learning in Fog Computing. *IEEE Internet of Things Journal*, 7(11):10782–

10793, 2020.
[7] Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. A Blockchain-Based Decentralized Federated Learning Framework with Committee Consensus. *IEEE Network*, 35(1):234–241, 2021.
[8] P. Banerjee, N. Nikam, and S. Ruj. Blockchain Enabled Privacy Preserving Data Audit. *CoRR*, abs/1904.12362, 2019.
[9] J Sengupta, S Ruj, and S Das Bit. A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT. *Journal of Network and Computer Applications*, 149:102481, 2020.
[10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In *Advances in Cryptology - ASIACRYPT 2010*, pages 177–194, 2010.
[11] Adi Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, November 1979

V. BIOGRAPHY SECTION

Jayasree Sengupta received her MTech degree in Distributed and Mobile Computing from Jadavpur University, Kolkata, India, in 2017. She is currently pursuing her Ph.D. with the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India. She has published research articles in reputed peer-reviewed journals and international conference proceedings. Her research interests include Applied Cryptography, Blockchains, Fog computing, IoT, and IIoT. She is a Student Member of IEEE and ACM.

Sushmita Ruj received her B.E. degree in Computer Science from Bengal Engineering and Science University, Shibpur, India, and her Masters and Ph.D. in Computer Science from Indian Statistical Institute. She was an Erasmus Mundus Post-Doctoral Fellow at Lund University, Sweden, and a Post-Doctoral Fellow at the University of Ottawa, Canada. She is currently a Senior Lecturer at the University of New South Wales, Sydney, Australia. Prior to that, she had served as a Senior Research Scientist at CSIRO Data61, Australia, and also as an Associate Professor at Indian Statistical Institute, Kolkata. Her research interests are in Blockchains, Applied Cryptography, and Data Privacy. She serves as an Associate Editor of Elsevier Journal, Information Security and Applications, and is involved with a number of conferences as a Program Co-Chairs or committee member. She is a recipient of the Samsung GRO award, NetApp Faculty Fellowship, Cisco Academic Grant, and IBM OCSP grant. She is a Senior Member of the ACM and IEEE.

Sipra Das Bit is a Professor in the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India. She also served as a visiting professor in the Department of Information and Communication Technology, Asian Institute of Technology, Bangkok, in 2017. A recipient of the Career Award for Young Teachers from the All India Council of Technical Education (AICTE), she has more than 30 years of teaching and research experience. Professor Das Bit has published many research papers in reputed journals and refereed international conference proceedings. She also has three books to her credit. Her current research interests include the Internet of Things, wireless sensor network, delay-tolerant network, mobile computing, and network security. She is a Senior Member of IEEE.