PEFL: A Privacy-Enhanced Federated Learning Scheme for Big Data Analytics

Jiale Zhang[†], Bing Chen[†‡], Shui Yu[§], and Hai Deng[†]
[†]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[‡]Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China
[§]School of Computer Science, University of Technology Sydney, Sydney, Australia
Email:{jlzhang, cb_china, denghai}@nuaa.edu.cn; Shui.Yu@uts.edu.au

***Abstract*—Federated learning has emerged as a promising solution for big data analytics, which jointly trains a global model across multiple mobile devices. However, participants' sensitive data information may be leaked to an untrusted server through uploaded gradient vectors. To address this problem, we propose a privacy-enhanced federated learning (PEFL) scheme to protect the gradients over an untrusted server. This is mainly enabled by encrypting participants' local gradients with Paillier homomorphic cryptosystem. In order to reduce the computation costs of the cryptosystem, we utilize the distributed selective stochastic gradient descent (DSSGD) method in the local training phase to achieve the distributed encryption. Moreover, the encrypted gradients can be further used for secure sum aggregation at the server side. In this way, the untrusted server can only learn the aggregated statistics for all the participants' updates, while each individual's private information will be well-protected. For the security analysis, we theoretically prove that our scheme is secure under several cryptographic hard problems. Exhaustive experimental results demonstrate that PEFL has low computation costs while reaching high accuracy in the settings of federated learning.**

***Index Terms*—Federated learning, big data analysis, privacy-preserving, secure data aggregation.**

I. INTRODUCTION

The proliferation of mobile devices, such as smartphones, wearables, and smart vehicles, are boosting the emergence of big data era [1]. Specifically, the Internet of Things (IoT) technology and its networking applications have shown an explosive growth tendency and generate massive sensing data every minute. The sensed big data contains a large amount of meaningful information which can be used to further analysis by learning the potential patterns embedded in the raw data. Deep learning is an effective tool that is envisioned to be an indispensable service universally for intelligent big data analytics [2]. However, such computation scenario demands participants to outsource their sensitive data to a third party in order to carry out deep learning services, which causes a significant privacy concern for mobile users [3].

Federated Learning (FL) [4], [5] has been proposed recently to provide intelligent big data analysis while protecting user's data privacy. FL trains a global model across thousands of mobile devices while keeping the training data locally, and uploading only the model updates (i.e., parameters of gradient and weight) to the central server [6]. Those local updates will be aggregated and processed to jointly optimize the current global model at the server side until the convergence condition meets. In this paper, we consider a scenario that applying federated learning framework to big data analytics.

As mentioned above, although FL can provide a basic privacy guarantee for each participant's raw data, a user's private data is possibly leaked to an *untrusted server* even with a small portion of the uploaded parameters [7], such as model inversion attack and gradient inference attack as described in [8] and [9], respectively. Existing literatures [9]–[11] demonstrate that the untrusted server has sufficient knowledge to reveal participants' original data from their uploaded local gradients. Firstly, the untrusted server is entitled to obtain a large amount of auxiliary knowledge (e.g., model structure and initialization parameters) about each participant's local training model. Secondly, it may infer some unintended information (e.g., change of gradients and truth label) according to each participant's uploaded gradients. Thirdly, it also can collude with a subset of the corrupted participants to reveal more precise privacy information (e.g., each individual's gradient) of other users.

Moreover, several existing approaches [6], [11]–[14] have been proposed to protect users' data privacy from the perspective of providing Differential Privacy (DP) guarantees. However, these privacy solutions rely on the presence of a trusted server to perturb the global model parameters and publish these noised parameters to each participant securely. That means the server is able to see each individual's uploaded gradients. Thus, if there exists an untrusted and knowledgeable server in the federated learning system, it is impossible to prevent the gradient-based privacy leakage by only considering DP in the server side. Therefore, it is an important challenge to preserve training data privacy against an untrusted server in federated learning.

In this paper, we solve the aforementioned challenge by considering a Privacy-Enhanced Federated Learning scheme, named *PEFL*, through the combination of federated learning and additively homomorphic cryptosystem [15]. In particular, to reduce the computation costs of the cryptosystem, we utilize the DSSGD method [18] to the local training procedure, combining with homomorphic encryption to realize the distributed encryption. Moreover, the additive homomorphic property of our used Paillier cryptosystem can sufficiently achieve secure sum aggregation of all participants' gradients at the server side. In summary, the main contributions of this

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

paper are as follows.

- We propose a privacy-enhanced federated learning scheme to protect the local model gradients against an untrusted server by using the Paillier homomorphic cryptosystem. The server in PEFL can only obtain the aggregated gradients while each individual's private information can be well-protected.
- We utilize the DSSGD method in the local training phase to distributedly execute the encryption algorithm, so that the computation costs of cryptosystem are reduced. Besides, our proposed scheme also provides the authentication mechanism to verify the identity information of all the participants.
- We theoretically prove that our scheme is secure under several well-known cryptographic hard problems. We also conduct exhaustive experiments to illustrate that our scheme has low computation costs while showing high accuracy on an image classification task.

The rest of this paper is organized as follows. In Section II, we briefly introduce the preliminaries. The security requirements and system overview are described in Section III. The construction of the proposed PEFL scheme is detailed in Section IV, and the exhaustive security analysis is provided in Section V. The performance evaluation as well as discusses are presented in Section VI. Finally, Section VII gives the summary and future work.

II. Preliminaries

A. Federated Learning Protocol

Federated learning [4] achieves decentralized machine learning services by distributing the training phase across n participants, which presents significant advantages in client-side data privacy because of the localized model training. After a certain communication round t, client i downloads the parameters of the current global model from the server, then the model is trained locally on its own private dataset to generate the gradient updates $G_{t+1}^{(i)}$. Finally, those local updates are uploaded to the central server and further aggregated and averaged to obtain the new global model:

$$w_{t+1}^{(global)} = w_t^{(global)} + \frac{1}{n}\sum_{i=1}^{n} G_{t+1}^{(i)} \tag{1}$$

where $w_t^{(global)}$ indicates the shared model at t-th communication round. The aforementioned steps will be iteratively executed until the global model training is completed.

B. Paillier homomorphic cryptosystem

To prevent the privacy leakage of participants' uploaded gradients from the untrusted aggregator, we utilize the Paillier homomorphic cryptosystem [15] to guarantee the confidentiality and secure aggregation in the whole federated learning procedure. The concrete description of Paillier cryptosystem is shown as follows:

- *KeyGen:* Randomly choosing two large primes (p, q), and the RSA modulus can be computed as $n = pq$. We define the Carmichael function as $\lambda = (p-1)(q-1)$ and g as a generator of $\mathbb{Z}_{n^2}^*$ with an order n, where $g^n \mod n^2 = 1$. Then, functions $L(u) = \frac{u-1}{n}$ and $\mu = \left(L(g^\lambda \mod n^2)\right)^{-1}$ are calculated. Thus, the public and private key pair can be shown as $(pk, sk) = \{(n, g), (\lambda, \mu)\}$.
- *ENC:* For any plaintext $m \in \mathbb{Z}_n$, randomly generate a number r such that $gcd(r, n) = 1$, the ciphertext can be calculated as $c = g^m \cdot r^n \mod n^2$.
- *AGG:* Given a set of ciphertexts $c^{(i)}$, the addition of all these ciphertexts can be computed as $c^{(agg)} = \sum_{i=1}^{k} c^{(i)}$, where k is the number of ciphertexts.
- *DEC:* Given the aggregated ciphertext $c^{(agg)}$, the corresponding aggregated plaintext can be recovered as $m = L(c^{(agg)^\lambda} \mod n^2)\mu \mod n$.

The aforementioned Paillier homomorphic cryptosystem is proved to be indistinguishability against Chosen Plaintext Attack (CPA) [16] based on decisional composite residuosity problem, which means the ciphertexts will leakage no bit of information about the plaintexts.

III. Security Requirements and System Overview

A. Security Requirements

Different from Hitaj et al. [17] where the learning participants are considered as the active attackers to infer other users' private training data, our security model assumes that the central server is the untrusted (honest-but-curious) adversary while all the participants are seen as trusted entities. Honest-but-curious server means that it will faithfully follow the designed federated learning protocol but attempt to infer private information from each user's local update. In this situation, to prevent the indirect privacy leakage of users' updated gradients, the following security requirements must be guaranteed.

- *Confidentiality*: Protecting the users' uploaded gradient vectors from the untrusted server, even if there exists an external attacker who can eavesdrop the communication channels or collude with the server, although the latter is not the mainly concerned in our work.
- *Authentication*: Ensures that all the local updates are really from the authorized participants and the received updates cannot be denied by participants.
- *Privacy-preserving*: As long as the confidentiality of gradients and the authentication of participants can be guaranteed, the privacy of sensitive information embedded in the gradient vectors can also be protected.

Since an untrusted and knowledgeable server can infer sensitive information embedded in each participant's local gradients (e.g., gradient inference attack [9]), it is necessary to provide a strong privacy guarantee against the untrusted server. Thus, the following objectives should be achieved. 1) *Security and privacy:* the above-stated security requirements should be satisfied; 2) *Accuracy:* PEFL should perform high model accuracy; 3) *Efficiency:* the computational overheads of cryptosystem is required to be low.

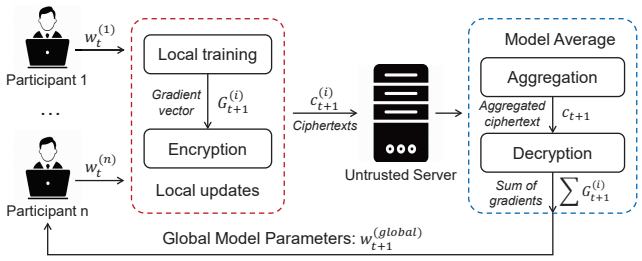

Fig. 1. Privacy-enhanced federated learning system

B. System Overview

To achieve the above mentioned security requirements, we consider a federated learning system for big data analysis where multiple participants jointly train a global model maintained by an untrusted server. Specifically, users upload the gradients information by training the global model on their private datasets. An untrusted server is assumed to be the attacker and try to infer any private information about users' training datasets using all gradients it receives. Fig. 1 gives the high-level description of our proposed PEFL system. Assuming a federated learning scenario contains one untrusted aggregator and n participants denoted as $[n] = \{1, 2, \cdots, n\}$. In a certain communication round, each participant $i \in [n]$ downloads the global model and trains the model on the local datasets $\mathcal{D}$ to generate the model update $G_t^{(i)}$, which can be mapped to a certain domain $\mathcal{G} \in [0, R)$. Let $G_t \in \mathcal{G}$ represents a vector of uploaded gradients that the untrusted server wants to infer sensitive information from it. In order to prevent privacy leakage, we use the Paillier homomorphic cryptosystem to encrypt each user's gradient vector, so the untrusted server cannot recognize the real values of users' updates. In this situation, the aggregator can only execute the sum aggregation operation to these ciphertexts based on the federated learning protocol while each individual's gradients information is invisible to the server, so as to provide a strong privacy guarantee of users' private training data.

IV. Proposed PEFL Scheme

This section presents the details of the proposed PEFL scheme for protecting user privacy against an untrusted server. The main idea is utilizing Paillier homomorphic cryptosystem to prevent information leakage of gradient and meanwhile achieving secure aggregation at the server side.

A. System Initialization

In our proposed PEFL, there exists a single trusted authority (TA) who can bootstrap the federated learning system and transmit the cryptographic materials (i.e., system parameters and keys) to the users and server. In general, the TA is only involved in the initialization phase and will not join the subsequent process. TA runs the $KeyGen$ algorithm of Paillier homomorphic cryptosystem, and generate the private/public key pair $(pk, sk) = \{(n, g), (\lambda, \mu)\}$. Then, TA chooses two secure cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H_2 : \mathbb{G} \rightarrow \mathbb{Z}_p^*$. At last, it releases the system parameters $sps = (n, g, \mathbb{G}, \mathbb{G}_T, H_1, H_2)$ to the sampled participants and assigns the master keys $msk = (\lambda, \mu, p, q)$ to the server via a secure channel.

B. Registration and Authentication

When a participant i joins the designed PEFL system, a registration phase is needed to authenticate i's identity. First, i chooses a random value $x_i \in \mathbb{Z}_p^*$ as his authentication private key and the corresponding public key can be represented as $y_i = g^{x_i} \in \mathbb{G}$. Then, i picks another value $k_i \in \mathbb{Z}_p^*$ randomly as the blinding factor to compute $e_i = H_1(k_i||ID_i||TS_t)$, where ID_i is the unique identifier of i in the time stamp TS_t. At last, participant i sends the registration knowledge $\{y_i, \alpha_i, \beta_i\}$ to the server for authentication, where $\alpha_i = g^{e_i}$ and $\beta_i = e_i - x_i H_2(\alpha_i)$.

After receiving the registration knowledge $\{y_i, \alpha_i, \beta_i\}$ from participant i, the server can authenticate i's identifier information by verifying $\alpha_i \stackrel{?}{=} g^{\beta_i} y_i^{H_2(\alpha_i)}$. If it does hold, participant i will be allowed to join the federated learning protocol, otherwise i will be rejected.

C. Local Updates Generation

Once the participant i has been successfully authenticated, i will be required to upload the local update by training a global model on his local datasets. The local training procedure can be executed by running the stochastic gradient descent (SGD) algorithm which main purpose is to find the parameter w by minimizing the loss function $\mathcal{L}(w, d_i)$, where $d_i \in \mathcal{D}$ is the training samples of participant i. Thus, the gradient $g_t^{(i)}$ in a certain communication round t can be calculated as:

$$g_t^{(i)} = \nabla_{w_t} \mathcal{L}(w_t, d_i) \tag{2}$$

In the gradient descent process, we utilize the distributed selective stochastic gradient descent (DSSGD) method [18] to achieve distributed computation capability. DSSGD splits the weight vector w_t and the gradient vector g_t into n parts, namely $w_t = (w_t^1, \cdots, w_t^n)$ and $g_t = (g_t^1, \cdots, g_t^n)$, so the local parameters update rule can be represented as:

$$w_{t+1}^{(i)} = w_t^{(i)} - \eta \cdot g_t^{(i)} \tag{3}$$

Then, the local model update G_{t+1} which used in the next communication round can be obtained by executing the following equation:

$$G_{t+1}^{(i)} = w_{t+1}^{(i)} - w_t^{(global)} \tag{4}$$

Here, global model parameters $w_t^{(global)}$ of communication round t is downloaded from the central server.

While participant i received the public key PK_P assigned by TA, he executes the *ENC* algorithm of Paillier cryptosystem to generate the ciphertext for his gradient vector G_{t+1}. Note that, G_{t+1} consists of a large amount of real numbers and each number z can be represented as $\lfloor z \cdot 2^{\epsilon} \rfloor \in \mathbb{Z}$, where ϵ is the precision of bits. In this way, both weight vector $w^{(i)}$ and gradient vector $G^{(i)}$ are in the plaintext domain $[0, R)$, which means the cryptosystem can be easily constructed on these two real vectors. Thus, to obtain the ciphertext format of

gradient vector, the participant i randomly generates a value $v_t^{(i)} \in \mathbb{Z}_{n^2}^*$ and computes the ciphertext as:

$$c_{t+1}^{(i)} = g^{(G_{t+1}^{(i)})} \cdot (v_t^{(i)})^n \mod n^2 \quad (5)$$

Then, i sends the ciphertext $c_{t+1}^{(i)}$ to the server for secure aggregation and further updating the current global model. Benefiting from the DSSGD method, the encryption phase of our scheme can be executed in a distributed manner, so as to greatly reduce the computation costs.

Algorithm 1 *Overall PEFL Algorithm.* N is the total number of participants; $E_k \in (1, \cdots, E)$ is the number of local epochs; b is the local mini-batch size sliced client's holding dataset $\mathcal{B}$; T is the number of total communication rounds; m_{t+1} is the number of sampled participant in communication round $t+1$; $\mathcal{L}(w, d_i)$ is the loss function, d_i is training dataset.

procedure SERVER EXECUTION
EVENT: Receive updated ciphertexts $c_{t+1}^{(i)}$
    Choose initial global parameters $w_t^{(global)}$
    for all $i \in m_{t+1}$ **do**
        $c_{t+1} = \prod_{i=1}^{m_{t+1}} c_{t+1}^{(i)} \mod n^2$
        $G_{t+1} = \frac{L({c_{t+1}}^{\lambda} \mod n^2)}{L(g^{\lambda} \mod n^2)} \mod n$
        $w_{t+1}^{(global)} = w_t^{(global)} + \frac{1}{m_{t+1}} G_{t+1}$
    end for
    Send $w_{t+1}^{(global)}$ to the participant;
end procedure
function CLIENT UPDATE
    Initialize: $w_t^{(i)} \leftarrow w_t^{(global)}$
    for each $E_k \in (1, \cdots, E)$ **do**
        for batch $b \in \mathcal{B}$ **do**
            $g_t^{(i)} = \nabla_{w_t} \mathcal{L}(w_t, d_i)$
            $w_{t+1}^{(i)} = w_t^{(i)} - \eta \cdot g_t^{(i)}$
        end for
        $G_{t+1}^{(i)} = w_{t+1}^{(i)} - w_t^{(global)}$
    end for
    for each client $i \in (1, \cdots, m_{t+1})$ **do**
        $c_{t+1}^{(i)} = g^{(G_{t+1}^{(i)})} \cdot (v_t^{(i)})^n \mod n^2$
    end for
    Upload $c_{t+1}^{(i)}$ to the central server.
end function

D. Aggregation and Decryption

Let m_{t+1} be the total sampled participants in a certain communication round $t+1$. Once received m_{t+1} encrypted updates, the server firstly runs the *AGG* algorithm with Paillier homomorphic property and gets the aggregated result:

$$\begin{aligned} c_{t+1} &= \textstyle\prod_{i=1}^{m_{t+1}} c_{t+1}^{(i)} \mod n^2 \\ &= \textstyle\prod_{i=1}^{m_{t+1}} g^{(G_{t+1}^{(i)})} \cdot (v_t^{(i)})^n \mod n^2 \\ &= g^{\sum_{i=1}^{m_{t+1}} (G_{t+1}^{(i)})} \cdot \textstyle\prod_{i=1}^{m_{t+1}} (v_t^{(i)})^n \mod n^2 \\ &= g^{G_{t+1}} \cdot v_t^n \mod n^2 \end{aligned} \quad (6)$$

where $G_{t+1} = \sum_{i=1}^{m_{t+1}} (G_{t+1}^{(i)})$ and $v_t = \prod_{i=1}^{m_{t+1}} (v_t^{(i)})$

Then, the aggregator further decrypts the aggregated result by executing the *DEC* algorithm with the Paillier private key $sk = (\mu, \lambda)$ as:

$$G_{t+1} = \sum_{i=1}^{m_{t+1}} (G_{t+1}^{(i)}) = \frac{L({c_{t+1}}^{\lambda} \mod n^2)}{L(g^{\lambda} \mod n^2)} \mod n. \quad (7)$$

E. Global Model Update

At last, the server further executes the federated average procedure to generate global model of communication round $t+1$:

$$w_{t+1}^{(global)} = w_t^{(global)} + \frac{1}{m_{t+1}} G_{t+1}. \quad (8)$$

The whole federated learning procedure will be executed iteratively until the global model $w^{(global)}$ tends to convergence. The pseudo-code of our overall PEFL scheme was shown in Algorithm 1.

V. Security Analysis

In this section, we present the security analysis of our PEFL scheme. In particular, according to the security requirements and design goals described in section III, our security analysis will focus on the participant authentication, confidentiality, and privacy-preserving.

A. Participant Authentication

The registration and authentication phase in our designed PEFL scheme basically relies on the extended Schnorr's signature and verification mechanism [19], which is proved to be secure based on the discrete logarithm assumption. The correctness of participants authentication can be represented as follow.

$$g^{\beta_i} y_i^{H_2(\alpha_i)} = g^{e_i - x_i H_2(\alpha_i)} g^{x_i H_2(\alpha_i)} = g^{e_i} = \alpha_i. \quad (9)$$

Here, we demonstrate that the participant i's identity can be efficiently authenticated as long as the discrete logarithm assumption holds.

Practically, it is impossible that an attacker can find a collision e_i' to fabricate the registration knowledge $\{y_i, \alpha_i, \beta_i\}$ because the value of x_i was selected privately and kept at the participant side. Besides, even if an untrusted server can obtain i's real identifier ID_i and current time stamp TS_t, it still cannot get the real value of e_i. That is mainly because e_i was further hidden by using a secure one-way hash function H_1 and a secretly blind factor k_i. Therefore, without the value of e_i, the possibility of an untrusted server obtaining the authentication private key x_i is negligible.

B. Confidentiality and Privacy-Preserving

To prevent the disclosure of the participants' training data privacy, we utilize the Paillier cryptosystem to encrypt the uploaded gradient vectors and aggregate the ciphertext based on the additively homomorphic property. In this way, the update of each client is encrypted as $c_{t+1}^{(i)} = g^{(G_{t+1}^{(i)})} \cdot (v_t^{(i)})^n \mod n^2$ and the aggregated data in the server-side can be formed as $c_{t+1} = g^{G_{t+1}} \cdot v_t^n \mod n^2$, which is still a valid ciphertext of Paillier cryptosystem. Since the Paillier cryptosystem is semantically secure against Chosen Plaintext Attack (CPA)

TABLE I
RUNNING TIMES WITH DIFFERENT NUMBERS OF GRADIENTS

Number of gradients	50000	100000	150000	200000	250000	300000	350000	400000	450000	500000
Encryption time (s)	0.791	1.558	2.286	2.976	3.752	4.635	5.712	6.964	7.891	9.013
Decryption time (s)	0.688	1.401	2.112	2.785	3.473	4.306	5.477	6.398	7.286	8.354

[15], [16], no sensitive information will be leaked. Even with the decryption result for the aggregated ciphertext, it still impossible to obtain the individual gradient vector, as the plaintext is compressed as $G_{t+1} = \sum_{i=1}^{m_{t+1}} (G_{t+1}^{(i)})$.

Specifically, although the server in our PEFL scheme is assumed as untrusted and it can obtain both each participant's ciphertext $c_{t+1}^{(i)}$ and the aggregated result c_{t+1}, it still cannot infer any individual private information. On the one side, without each participant's private key, the untrusted server is unable to decrypt the individual ciphertext but aggregating the ciphertexts to get the aggregated result. On the other side, although the server can decrypt the aggregated result and get the compressed plaintext G_{t+1}, it still cannot recovers each individual's plaintext $G_{t+1}^{(i)}$ from this summed result. In summary, the confidentiality and privacy of each participant's gradient vector can be perfectly protected, so as to leakage no information about users' training datasets.

VI. Performance Evaluation and Discussions

In this section, we conduct two sets of experiments to evaluate the performance of our proposed PEFL scheme, including the computational costs of Paillier cryptosystem and the classification accuracy comparison on MNIST dataset.

A. Dataset and Experimental Setup

1) Dataset: MNIST is one of the publicly available datasets which contains 70000 instances of handwritten digits ranging from 0 to 9 (i.e., 10 classes). Each image is with the size of 28×28 pixels, and the whole MNIST dataset is split into 60000 training records and 10000 testing instances.

2) Experimental Setup: For the computational overheads evaluations of Paillier cryptosystem, we use the public Pairing-Based Cryptography (PBC) library to eliminate the time costs operations, where the RSA modulus n is set to 1024bits and the security parameter p is 160 bits. For the classification accuracy comparison experiment, we utilize the Convolutional Neural Network (CNN) based architecture to construct an image classification model on MNIST dataset. The network structure of CNN consists of two convolutional layers and two dense layers and the kernel size is 4×4. The activation function of two dense layers is *ReLU* and the output of this network is a *Softmax* layer of 10 classes. All the experiments are implemented on a RHEL7.5 server with NVidia Quadro P4000 GPU and 32GB RAM.

B. Computational Costs of Cryptosystem

Firstly, we estimate the computational costs of encryption and decryption of Paillier homomorphic encryption mechanism under different lengths of the gradient vectors. Here, we consider the precision of each real number of gradient vector is 32 bits. Table I shows the running times for encryption and decryption on different numbers of gradient, using 1024 bits RSA modulus n and 160 bits security parameter p under PBC library. Note that the encryption and decryption processes in Paillier cryptosystem require two exponentiation operations in $\mathbb{Z}_{n^2}$ and the multiplication operations in $\mathbb{Z}_{n^2}$ are considered negligible. Fig. 2 further depicts the time costs of encryption and decryption operations, we can see that although the running times increase with the rise of the number of gradients, it still maintains at a low level.

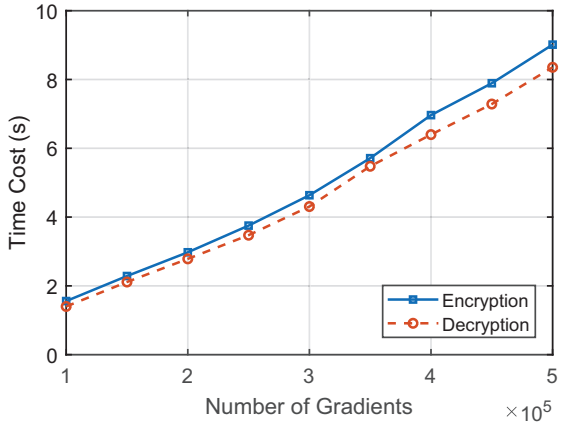

Fig. 2. Computational costs of encryption and decryption

Moreover, we also apply the Paillier cryptosystem to the generated gradient vectors by training a CNN model on MNIST dataset to further evaluate the computational costs. Our CNN model with MNIST dataset can be formed as: 784 (input)→256 (dense)→64 (dense)→10 (output). Thus, the total number of gradients for our used CNN model is (784+1)×256+(256+1)×64+(64+1)×10=218058. Fig. 3 illustrates the time costs of encryption, decryption, and aggregation on the overall training procedure as the number of participants increases. Since the participants' models are trained in a distributed manner, the impact of encryption remains at a constant value approximately among different participants settings. Meanwhile, the time costs of aggregation and decryption phases on the server side are very low compared with the training time.

C. Performance of PEFL on MNIST Dataset

For the performance evaluation, we implement our proposed PEFL scheme as an image-classification task over the benchmark dataset MNIST to quantify the classification accuracy over communication rounds. The experiment is conducted under federated learning settings using the PyTorch framework. During the local training procedure, we set the local epoch $E = 2$ with the learning rate of $\eta = 0.1$ and η

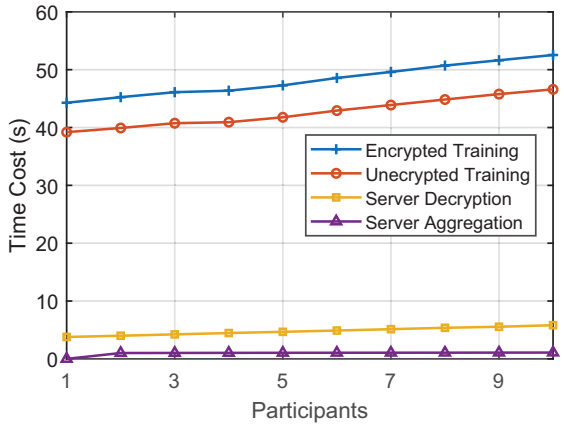

Fig. 3. Computational costs of cryptosystem on MNIST dataset

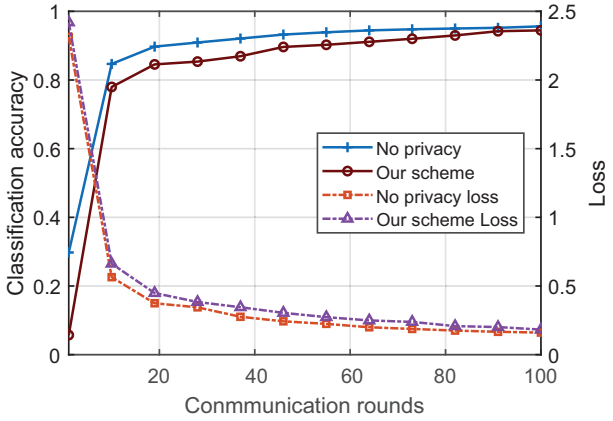

Fig. 4. Accuracy comparison over communication rounds

drops by 0.01 every two epochs. The sampled participants m_t are fixed with 20. At each communication round, the participants' models are trained individually and distributedly before being averaged into the global model. We choose a similar process of MNIST dataset as [4] and [12], where the whole dataset was divided into 10 shards and each client got two shards. The training accuracy and loss results are shown in Fig. 4, we can see that the accuracy with our PEFL setting is very close to the accuracy with no privacy setting, which means our PEFL can integrate cryptographic method in federated learning system without sacrifice too much model accuracy.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802303, in part by the National Natural Science Foundation of China under Grant 61672283, and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX18_0308.

VII. SUMMARY AND FURTHER WORK

In this paper, we presented a privacy-enhanced federated learning protocol, PEFL, based on the additively homomorphic Cryptosystem. PEFL can prevent users' privacy leakage from an untrusted server, meaning the server in our scheme cannot learn any private information from each individual's local model gradients. Security analysis proves that the properties of authentication, confidentiality, and privacy-preserving can be achieved in our scheme. Performance evaluation demonstrates that our proposed PEFL can achieve high model accuracy on the classification task with low computation overheads. In regards to future work, we plan to design a decentralized secure aggregation mechanism without any trusted authority, and further improve the availability.

REFERENCES

[1] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for Big Data: A Survey," *IEEE Communications Surveys & Tutorials.*, vol. 19, no. 1, pp. 531-549, 2017.
[2] M. A. Alsheikh, D. Niyato, S. Lin, H.-P. Tan, and Z. Han, "Mobile big data analytics using deep learning and apache spark," *IEEE Network.*, vol. 30, no. 3, pp. 22-29, 2016.
[3] S. Yu, "Big Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data," *IEEE Access.*, vol. 4, pp. 2751-2763, 2016.
[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A.y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc of AISTATS'17*, Fort Lauderadale, Florida, USA, Apr. 2017, pp. 1-10.
[5] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology.*, vol. 10, no. 2, pp: 1-19, 2019.
[6] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in *Proc of ACM CCS'15*, Denver, Colorado, USA, Oct. 2015, pp. 1310-1321.
[7] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proc of ACM CCS'17*, Dallas, Texas, USA, Oct. 2017, pp. 1175-1191.
[8] F. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in *Proc of ACM CCS'15*, Denver, Colorado, USA, Oct. 2015, pp. 1322-1333.
[9] L. T. Phong, Y. Aono, T. Hayashi. L. Wang, and S. Moriai, "Privacy-Preserving Deep Learning via Additively Homomorphic Encryption," *IEEE Trans. Inf. Forensics Security.*, vol. 13, no. 5, pp. 1333-1345, May. 2018.
[10] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning," in *Proc of IEEE INFOCOM'19*, Paris, France, April. 2019, PP. 2512-2520.
[11] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep Learning with Differential Privacy," in *Proc of ACM CCS'16*, Vienna, Austria, Oct. 2016, pp. 308-318.
[12] R. C. Geyer, T. Klein, and M. Nabi, "Differentially Private Federated Learning: A Client Level Perspective," in *Proc of NIPS'17*, Long Beach, CA, USA, Dec. 2017, pp. 1-7.
[13] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, and R. Zhang, "A Hybrid Approach to Privacy-Preserving Federated Learning," *arXiv preprint arXiv:1812.03224*, 2018.
[14] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3, pp. 211-407, Aug. 2014.
[15] P. Paillier, "Public-Key Cryptosystems based on Composite Degree Residuosity Classes," in *Proc of ACM EUROCRYPT'99*, Prague, Czech Republic, May. 1999, pp. 223-238.
[16] O. Goldreich, "Foundations of Cryptography: Basic Applications" in *Press. Cambridge University*, vol. 2, Cambridge, UK, 2004.
[17] B. Hitaj, G. Ateniese, and F. P-Cruz, "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning," in *Proc of ACM CCS'17*, Dallas, TX, USA, Oct. 2017, pp. 603-618.
[18] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le and A. Y. Ag, "Large Scale Distributed Deep Networks," in *Proc of NIPS'12*, Lake Tahoe, Nevada, USA, Dec. 2012, pp. 1232-1240.
[19] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology.*, vol. 4, no. 3, pp. 161-174, 1991.