Combating Deep Leakage from Gradients in Cross-Silo Federated Learning with QKD

Xiaoyu Wang[1], Yangming Zhao[2,3,4], Chen Tian[2], Kai Chen[5], Qi Li[6], Kun Yang[2,3,7], and Chunming Qiao[8]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2] State Key Laboratory for Novel Software Technology, Nanjing University
[3] School of Intelligent Software and Engineering, Nanjing University - Suzhou Campus
[4]Hefei National Laboratory, University of Science and Technology of China
[5]Hongkong University of Science and Technology [6]Tsinghua University
[7]University of Essex [8]University at Buffalo

***Abstract*— Deep Leakage from Gradients (DLG) could reveal training data privacy from gradients transmitted over an insecure channel in Cross-Silo Federated Learning (CSFL) systems. So far, One-Time Pad (OTP) based on secret keys generated by Quantum Key Distribution (QKD) is the only perfectly secure approach to defending channel security and preserving privacy. Nevertheless, current QKD systems cannot generate keys at a rate high enough to support OTP in practical CSFL systems, while we find that encrypting only part of the gradients or several bits of each gradient is not adequate to preserve data privacy. To overcome these challenges, we propose QuGrad to encrypt each gradient using only one bit of secret keys. In QuGrad, it is unpredictable which or how many bits of each gradient will be changed and the encrypted gradient vector will be orthogonal to the original one, which potentially hides the maximum amount of training data information. By implementing QuGrad on a testbed and conducting extensive experiments, we show that QuGrad can reduce the average Jaccard similarity between the recovered data and the original ones by up to 89% compared with the state-of-the-art technique to defend training data against DLG.**

I. Introduction

Cross-Silo Federated Learning (CSFL) [1, 2] is widely used by industry organizations (*e.g.*, Alibaba, Microsoft, Amazon, *etc.*) to train Machine Learning (ML) models with the data collected by their data centers (*i.e.*, data silos) geographically located at different places connected through a Wide Area Network (WAN). The conventional wisdom is that training data privacy can be protected in CSFL systems since only the model parameters or gradients, instead of training data, are transmitted through insecure channels over the WAN. However, it has been recently shown that sensitive information about the training data could still be leaked from the gradients - a phenomenon referred to as Deep Leakage from Gradients, or DLG [3]–[5]. In this paper, we will study how to combat such DLG in CSFL systems.

The work of Yangming Zhao was supported in part by the National Natural Science Foundation of China under Grant 62272428; in part by Innovation Program for Quantum Science and Technology under grant 2021ZD0300705; in part by the Anhui Provincial Natural Science Foundation under Grant 2208085MF167; and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20221261.

Yangming Zhao is the corresponding author.

Quantum Key Distribution (QKD) [6, 7], a technique to distribute *perfectly secure* keys between two communication parties, is considered as the ultimate solution to ensure channel security. With the perfectly secure secret keys generated by QKD devices, we may XOR every bit of plaintext with one bit of disposable secret keys. This is called One-Time Pad (OTP) and has been proven to be the only encryption technique that is impossible to break [8]. To achieve OTP when encrypting gradients transmitted over the WAN in a CSFL system, QKD devices have to distribute keys between every worker (*i.e.*, a data silo) and the parameter server at a rate as fast as the gradient transmission over the WAN.

Current QKD techniques suffer from a low rate of secret key distribution and cannot enable us to achieve OTP in CSFL systems. A pair of commodity QKD devices, which cost hundreds of thousands of dollars, can distribute keys at the rate of only tens of kilobits per second [9]. Even in an experimental environment, a first-class QKD system can distribute only several megabits of secret keys per second [10]. As a result, current QKD techniques are not able to support OTP at runtime in CSFL systems, where the gradient transmission rate (over a WAN) is usually tens of megabits per second [11]. If we collect enough secret keys before starting each training epoch or each training job, it is time consuming and will result in a huge waste of computation resources. Accordingly, it is impractical to encrypt every bit of the gradients with one bit of disposable secret keys generated by existing QKD systems.

In this paper, we propose QuGrad to defend training data privacy in CSFL systems using keys generated by a QKD system. The central idea of QuGrad is to encrypt each gradient (*e.g.*, 32 bits) with just one bit of disposable secret keys, which significantly reduces the number of required secret keys. There are two main challenges in QuGrad: (i) QuGrad needs to hide the maximum amount of information about the real training data; and (ii) QuGrad should not impact the convergence rate of the training process and the accuracy of the derived model. To overcome these two challenges, we propose a deliberate encryption mechanism that changes unpredictable number of bits in every gradient and ensures that an encrypted gradient vector lies in the orthogonal space of the original gradient

vector. In addition, with proposed decryption method, the parameter server can get a scale of the original gradients.

Though QuGrad deviates from an ideal implementation of OTP, our extensive experiments using a real QKD system show that (i) QuGrad can better defend training data privacy against DLG compared with the State-Of-The-Art (SOTA) anti-DLG schemes, *e.g.*, Soteria [12] and FedCDP [13]. Specifically, when the training data are images, compared with SOTA anti-DLG schemes, QuGrad can increase the Learned Perceptual Image Patch Similarity (LPIPS) [14], a metric to quantify the difference of two figures according to human's natural perception, by up to $2.3\times$ (a smaller LPIPS indicates that a human being feels that two images are more similar to each other). It also decreases the Jaccard similarity [15], a metric to evaluate the similarity of two images from the computer vision perspective, by up to 89% and decreases HaarPSI [16], a metric to evaluate the similarity of two images from the signal processing perspective, by up to 69%; (ii) QuGrad will not impact the convergence rate of the training process nor the accuracy of a derived model, even if the QKD system suffers from a Quantum Bit Error Rate (QBER) of 10%.

To the best of our knowledge, QuGrad is the first work that applies the current QKD technique to a prevailing application system such as CSFL to ensure the channel security, and in particular, the first effort in turning an ideal OTP proposal into a practical implementation. The technique proposed in our work can be easily extended to other communication systems to defend channel security and preserve data privacy.

The main technical contributions of this work are

- Show that encrypting only part of the gradients or several bits of each gradient cannot perfectly preserve training data privacy via both analysis and experiments (Section IV-A);
- Propose QuGrad to defend training data privacy against DLG in CSFL systems (Section IV-B);
- Implement QuGrad using a real QKD testbed and conduct extensive experiments to show the superior performance of QuGrad (Section V).

II. Background and Motivation

In this section, we introduce some preliminary knowledge to provide background and motivation for our work.

A. Federated Learning and Privacy Leakage

Cross-Silo Federated Learning. In this work, we focus on a Cross-Silo Federated Learning (CSFL) [1, 2] system using the standard synchronous parameter server approach [17, 18]. In a CSFL system, multiple data centers belonging to the same organization (called *workers*), *i.e.*, data silos, spreading over different locations will collaborate to train a unified ML model. To this end, each worker keeps a copy of this model (called *local model*). In each training epoch, every worker trains its local model with its private data (called *local data*) and pushes the derived gradients to the parameter server. When receiving gradients from all workers, the parameter server aggregates them, updates the global model parameters, and pushes back the updated model parameters to every worker who will perform the next epoch of training.

In CSFL, the gradients derived from local training should be sent through a WAN. In a practical WAN, the data transmission rate is usually tens of megabits per second [11], which is much smaller than the rate to generate gradients. Accordingly, the available bandwidth in the WAN determines the data transmission rate between a worker and the parameter server in a practical CSFL system.

Deep Leakage from Gradients (DLG). A recent study [3] showed that training data privacy may be leaked from publicly shared gradients in CSFL. More specifically, suppose $F(x; \boldsymbol{w})$ is a differentiable ML model, $\boldsymbol{w}$ are the parameters of model F, and $l(F(x, \boldsymbol{w}), y)$ is the loss function associated with training sample (x, y), then given the gradients $\boldsymbol{v}$ transmitting in the WAN, one can recover the training data by finding (x^*, y^*) to minimize the distance between $\boldsymbol{u}^*$ and $\boldsymbol{v}$, *i.e.*, $D(\boldsymbol{u}^*, \boldsymbol{v})$, where $\boldsymbol{u}^* = \frac{\partial l(F(x^*, \boldsymbol{w}), y^*)}{\partial \boldsymbol{w}}$. Usually, $D(\boldsymbol{u}^*, \boldsymbol{v})$ is defined as cosine distance, *i.e.*, $D(\boldsymbol{u}^*, \boldsymbol{v}) = 1 - \frac{\langle \boldsymbol{u}^*, \boldsymbol{v} \rangle}{\|\boldsymbol{v}^*\| \|\boldsymbol{v}\|}$. [5] showed that training data privacy would also be leaked from the summation of gradients associated with a batch of training data. All DLG algorithms assumed that a smaller distance between two gradient vectors indicates a smaller distance between the two corresponding training data.

B. Previous Works on Combating DLG

The intuitive method to combat DLG is using an encryption based approach, *e.g.*, encrypting plaintext according to Transport Layer Security (TLS) and Advanced Encryption Standard (AES). Recently, homomorphic encryption based methods [19]–[21] have become popular since they permit users to perform computation on encrypted data without first decrypting it. The keys in all existing encryption based schemes would be reused, which incurs a risk that the keys will be broken based on the ciphertext.

Customized for combating DLG, some noise-based methods [12, 13, 22]–[25] that do not need secret keys are proposed. Among these works, Soteria [12] and FedCDP [13] are two representatives. Soteria noted that most high-quality information of the training data is contained in the representation layer. Accordingly, it proposed to perturb the gradients associated with the parameters in the representation layer. By limiting the magnitude of perturbation enforced to the gradients, Soteria can ensure the convergence of FedAvg [26] while defending the training data privacy against DLG.

FedCDP enforced perturbation to all gradients instead of perturbing the gradients only in the representation layer. FedCDP first scaled the gradients associated with every layer, such that the norm of these gradients (in each layer) has a limited magnitude. Then, by introducing Gaussian noise to the scaled gradients, FedCDP can achieve instance-level per-example differential privacy, which is robust to DLG.

Both Soteria and FedCDP achieved a good performance in combating conventional DLG algorithms [3]–[5]. However, neither of them is suitable to the CSFL scenario, where an attacker may get the training data features before trying to recover (or distill the privacy of) training data.

C. Challenges to Combating DLG in CSFL

In CSFL, a unique feature is that a DLG attacker itself may be able to collect some training data and distill some a priori knowledge about the data by *e.g.*, training a Generative Adversary Network (GAN). With such a GAN, say it is $G(\boldsymbol{z})$, we can find $(\boldsymbol{z}^*, y^*)$ to minimize $D(\boldsymbol{u}^*, \boldsymbol{v})$, where $\boldsymbol{u}^* = \frac{\partial l(F(G(\boldsymbol{z}^*), \boldsymbol{w}), y^*)}{\partial \boldsymbol{w}}$, and $G(\boldsymbol{z}^*)$ will be the recovered data. With the help of the GAN, the search space is reduced and hence more information can be recovered from training data. The representative of leveraging a GAN to recover training data is Generative Gradient Leakage (GGL) [27].

Since recovered data in GGL is directly generated by a GAN, it may be significantly different from the original one. To solve this problem, ROGS [15] first obtained low-resolution recovered data based on conventional DLG algorithms (without the help of a GAN), such as iDLG [4], and then leveraged a GAN to improve the resolution of the recovered data (*e.g.*, output by iDLG). In this way, ROGS recovered training data at the semantic level and was able to derive recovered data more similar to the original ones.

In the simulation (*i.e.*, Section V), we will see that when an attacker has a priori knowledge about the training data features, which is practical in CSFL, the SOTA schemes to defend data privacy against DLG, *e.g.*, Soteria and FedCDP, no longer work. Both ROGS [15] and GGL [27] can distill useful information, even recover the original picture, from the gradients protected by either Soteria or FedCDP.

D. Quantum Key Distribution

Quantum Key Distribution (QKD) is a secure communication method that implements a cryptographic protocol involving components of quantum mechanics. It enables two parties to produce shared random secret keys known only to them. In general, QKD techniques have two main features.

Low QKD Rate. Since a quantum particle, *e.g.*, a photon, has very little energy, it is very likely to be lost during transmission along a quantum channel. Accordingly, the existing QKD systems suffer from a low rate of distributing secret keys. With a commodity QKD device, we can distribute secret keys at the rate of tens to hundreds of kilobits per second [9]. Even in an experimental environment, the secret keys can be generated at a rate of only several megabits per second [10], which is much lower than the data transmission rate in a practical WAN.

Non-negligible Quantum Bit Error Rate (QBER). Due to the imperfections of the physical devices and the channel through which the quantum states propagate, the state of a quantum particle my change during transmission and it will incur a non-negligible QBER. According to the current technique, without a Quantum Error Correction (QEC) scheme, the QBER will be about 3.4% [28]. In this work, we will propose an encryption method that is robust to quantum bit errors.

III. Concept and Design Goals

In this section, we will first introduce the proposed conceptual design to defend training data privacy against DLG based on QKD in CSFL systems, and then present the desirable properties of such a system.

A. QKD-based Gradient Encryption Systems

In a QKD-based system to defend training data privacy against DLG in CSFL systems, we have to deploy one or multiple pairs of QKD devices between each worker and the parameter server to distribute secret keys. When a worker has produced the gradients via local training, it will first encrypt these gradients based on the secret keys generated by its QKD devices and then send the encrypted gradients to the parameter server through a classic channel. When the parameter server receives the encrypted gradients from a worker, it will decrypt them based on the secret keys from QKD devices before aggregating gradients from all workers and updating the global model parameters.

Though the parameter server has to send the updated parameters back to all workers, we do not need to encrypt these parameters. Even if a malicious attacker recovers a training data from the global model parameters, it cannot figure out from which worker such a training sample comes. Accordingly, we can preserve the data anonymity. On the other hand, without caring about the data ownership, one may generate similar training data with a generative model trained based on some data collected by itself. Furthermore, we can use the same defensive scheme as encrypting gradients sent from a worker to the parameter server to encrypt the updated parameters sent from the parameter server to workers at the cost of another set of QKD devices to distribute more keys.

To encrypt gradients on time so that the training process will not be significantly prolonged, suppose the gradient transmission rate between a worker and the parameter server is B bits per second (bps), the worker generates gradients at the rate of G bps, and E ($0 < E \leq 1$) bit of secret key is used to encrypt one bit of gradients on average, the secret keys should be generated at least at the rate of R bps, such that

$$R \geq E \min\{B, G\} \tag{1}$$

In a practical CSFL system, each worker generates gradients at the rate of several to tens of gigabits per second (Gbps), while the available bandwidth between two data silos is tens to hundreds of megabits per second (Mbps). However, a SOTA QKD device can only distribute secret keys at the rate of several Mbps. As a result, we have to encrypt tens of bits of gradients based on one bit of secret keys generated by QKD devices.

In this work, we assume a gradient is quantized with 32 bits and presented following the IEEE 754 Floating-Point Standard (*i.e.*, FP32 numbers). We leave the case that gradients are encoded in other formats, *e.g.*, [29], to our future work.

B. Desired Properties

We identify the following goals when designing a QKD-based system to defend training data privacy in CSFL.

- Privacy preservation: Our proposed system has to defend training data privacy against DLG, even if a malicious attacker has a set of data following the same distribution of the training data. With such a set of training data, the attacker can train a Generative Adversarial Network

(GAN) [30] to help reduce search space of recovering training data from gradients.

- Scalability: All the gradients transmitted from a worker to the parameter server should be encrypted based on the secret keys online generated by QKD devices. Since we do not know how many epochs a training job will run, it is difficult for us to estimate how many secret keys we should prepare for a specific training job. If we online collect secret keys for each training epoch and then start the corresponding gradient transmission, we may need too much time to collect enough keys and the CSFL system will stay idle. It results in serious resource wastage.
- Robustness: In a robust system, the training convergence and model accuracy should not be significantly degraded when QKD devices suffer a modest QBER (*e.g.*, about 3.4% [28]).

IV. QuGrad Design

In this section, we will design QuGrad to preserve training data privacy in CSFL systems. We first propose some principles that QuGrad should follow based on experiments and analysis, and then present QuGrad in detail. At last, we theoretically analyze the performance of QuGrad and discuss some practical issues on using QuGrad in practice.

A. Preliminary Analysis

The straightforward way to encrypt gradients with a limited number (and distribution rate) of secret keys is to XOR only some of the gradients or several (but not all) bits of each gradient with secret keys. In this section, we will demonstrate that we cannot achieve a good performance in combating DLG in CSFL following this line of thought via both experiments and analysis. To test the performance of each straightforward defensive scheme, we assume that an attacker leverages ROGS [15] or GGL [27] to recover the training data from gradients of training a ResNet18 [31] model based on the ImageNet [32] data set. Since both ROGS and GGL need to infer the ground truth label of training data and they cannot correctly recover training data from gradients based on a wrong label, to design a more general data privacy defensive scheme (rather than only preventing an attacker to obtain the ground truth label), we always input ground truth labels instead of inferring them using iDLG [4].

For brevity, here we only show the case for recovering one specific data sample (*i.e.*, one picture in ImageNet). Similar results can be derived based on other training samples. All experiment results for analysis in this section are summarized in Tab. I. When no defensive scheme is adopted (in the column labeled as "None"), ROGS can derive a fuzzier copy of the original figure. Though GGL derives a picture different from the original one, we can recognize the main information of the original figure, namely, there is a crab on the rock.

Encrypt All gradients at the Representation layer (EAR). Soteria [12] has shown that the gradients or parameters in the representation layer contain the most high-quality information of the training data. Thus, one may consider only encrypting the gradients in the representation layer. Since it may result in a number that does not satisfy the IEEE 754 Floating-Point Standard by XORing every bit of a gradient with secret keys, to present a conservative analysis, we test the performance of combating DLG by dropping all gradients in the representation layer and applying ROGS and GGL to recover the training data. From the results shown in Tab. I (column EAR), we observe that compared with the case without any defensive schemes, despite both ROGS and GGL getting a fuzzier figure, neither of them suffer from more semantic information loss.

Encrypt Several Bits of Some Gradients (ESBSG). An alternative to encrypting only the gradients at the representation layer is to encrypt more gradients, but doing so by only XORing some of the bits in the encrypted gradients with secret keys. The ESBSG column in Tab. I shows the performance of encrypting 25% of the gradients, but for each gradient, only the sixth to the ninth bits will be XORed with secret keys (the resulting number will always obey the IEEE 754 Floating-Point Standard). Using this approach, the magnitude of each gradient can be scaled by as much as 16 times. From the derived results, we can observe that although we can achieve a better defense performance than dropping all gradients at the representation layer, the major semantic information in the training data is still recovered by either ROGS or GGL.

Encrypt One bit of Every Gradient (EOEG). Following the idea of encrypting several bits of some of the gradients, the extreme case is to encrypt only one bit of every gradient. By randomly choosing a bit to encrypt, in most cases, we will end up with encrypting a mantissa bit. Since we cannot significantly change the value of a gradient by flipping its mantissa bit, doing so would not be adequate for defending data privacy against DLG. If we encrypt an exponent bit, only the gradients whose magnitudes are significantly changed help hide training data privacy. However, the attacker can simply drop all those gradients having an extremely large magnitude before attempting to recover training data. The remaining option is to encrypt the sign bit. By doing so, we will get gradients much different from original ones and it is difficult to identify which gradients have been changed. Accordingly, it would be the best choice to encrypt the sign bit of all gradients. However, the gradient magnitude information is kept and this still may reveal too much information to the attacker, especially when a GAN can be adapted.

To verify our analysis, we conduct an experiment to investigate the performance of defending training data privacy by encrypting the sign bit of every gradient. The experiment results are shown in the EOEG column of Tab. I. From the results, we can see that encrypting the sign bit of all gradients achieve a better performance in combating DLG than other two encrypting approaches discussed above and an attacker can get little useful information by using ROGS, however, we can still get some useful information via GGL that there is a crab-like reptile on the rack in the training sample.

Takeaways. Based on the above analysis, we have the following takeaways: (i) to defend training data privacy against DLG, it is not adequate to encrypt only a small percentage of

TABLE I
VISUALIZED RESULTS FOR PRELIMINARY ANALYSIS

Original	DEF / ATK	None	EAR	ESBSG	EOEG
	GGL				
	ROGS				

the gradients, no matter whether they are randomly selected or strategically selected (*e.g.*, at the representation layer); and (ii) it is also not adequate to encrypt only some strategically selected bits (*e.g.*, the sign bit) in each gradient. Accordingly, we have the following proposition.

Proposition 1. *QuGrad should be capable of changing a number of bits in every gradient.*

In order to design a high-performance defensive scheme, we still need to know what properties the encrypted gradients should have such that we can hide the most information of training data. For brevity, we assume that a DLG algorithm uses the cosine distance. Then, we have $0 \leq D(\boldsymbol{u}, \boldsymbol{v}) \leq 2$ for any vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. According to the encrypted gradient vector $\hat{\boldsymbol{v}}_i$, the attacker will synthesize a data sample that can derive a gradient vector $\hat{\boldsymbol{u}}_i$ as close to $\hat{\boldsymbol{v}}_i$ as possible. Accordingly, to hide the maximum amount of information about the training data, the encrypted gradients $\hat{\boldsymbol{v}}_i$ should be as far away from the original gradients $\boldsymbol{v}_i$ as possible. When $D(\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i) < 1$, we can change $\hat{\boldsymbol{v}}_i$ to increase $D(\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i)$, such that a DLG algorithm will derive data further away from the original data and hence the recovered data contains less information about the original one. When $D(\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i) > 1$, since $D(-\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i) = 2 - D(\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i) < 1$, the larger $D(\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i)$ is, the better evidence $-\hat{\boldsymbol{v}}_i$ will be for a DLG algorithm to recover the training data. Accordingly, by trying to infer the training data based on both of $-\hat{\boldsymbol{v}}_i$ and $\hat{\boldsymbol{v}}_i$, a DLG algorithm will obtain a data such that the distance between its corresponding gradient vector and the original one is less than 1. As a result, the best way to encrypt the gradients is to map the original gradient vector $\boldsymbol{v}_i$ to $\hat{\boldsymbol{v}}_i$, such that $D(\hat{\boldsymbol{v}}_i, \boldsymbol{v}_i) = 1$. In other words, $\langle \hat{\boldsymbol{v}}_i, \boldsymbol{v}_i \rangle = 0$, *i.e.*, the encrypted gradients should lie in the orthogonal space of the original gradients. According to above discussions, we propose the following proposition.

Proposition 2. *The encrypted gradients in QuGrad should lie in the orthogonal space of the original gradients.*

B. QuGrad in Detail

Based on the two propositions we proposed above, we design the following encryption scheme.

Encryption at the worker's side. To encrypt the gradients $\boldsymbol{v}_i = \{v_{ik}\}|_{k=1}^K$, a worker i first collects K bits of secret keys from QKD devices and generates a vector $\boldsymbol{s}_i = \{s_{ik}\}|_{k=1}^K$ such that $s_{ik} = 1$ if the k^{th} bit of secret keys is "1", and $s_{ik} = 0$ otherwise. Then, the worker encrypts $\boldsymbol{v}_i$ following

$$\hat{\boldsymbol{v}}_i = \begin{cases} \frac{\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle}{\|\boldsymbol{v}_i\|^2} \boldsymbol{v}_i - \boldsymbol{s}_i, & \text{if } \langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle \geq 0 \\ \boldsymbol{s}_i - \frac{\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle}{\|\boldsymbol{v}_i\|^2} \boldsymbol{v}_i, & \text{if } \langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle < 0 \end{cases} \tag{2}$$

where $\hat{\boldsymbol{v}}_i = \{\hat{v}_{ik}\}|_{k=1}^K$ are the encrypted gradients. Specifically, after encryption, a gradient v_{ik} will become either $\hat{v}_{ik} = \frac{\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle}{\|\boldsymbol{v}_i\|^2} v_{ik} - s_{ik}$ if $\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle \geq 0$ or $\hat{v}_{ik} = s_{ik} - \frac{\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle}{\|\boldsymbol{v}_i\|^2} v_{ik}$ if $\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle < 0$. Accordingly, the values of all gradients will change. In addition, by scaling the real gradients and combining them with the secret keys, a (unpredictable) number of bits in each gradient value will change and which bits that are changed in each gradient value is also unpredictable. Last but not least, we can verify that $\langle \boldsymbol{v}_i, \hat{\boldsymbol{v}}_i \rangle = 0$, *i.e.*, the encrypted gradients $\hat{\boldsymbol{v}}_i$ is orthogonal to the original gradients $\boldsymbol{v}_i$.

Decryption at the parameter server's side. Via QKD, the parameter server has the secret keys (*i.e.*, $\boldsymbol{s}_i$) used to encrypt the gradients. In addition, the parameter server will receive the encrypted gradients (*i.e.*, $\hat{\boldsymbol{v}}_i$) from the classic communication channel. Note that

$$\langle \hat{\boldsymbol{v}}_i, \boldsymbol{s}_i \rangle = \begin{cases} (\cos^2\theta - 1)\|\boldsymbol{s}_i\|^2 < 0, & \text{if } \langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle \geq 0 \\ (1 - \cos^2\theta)\|\boldsymbol{s}_i\|^2 > 0, & \text{if } \langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle < 0 \end{cases} \tag{3}$$

where θ is the angle between $\hat{\boldsymbol{v}}_i$ and $\boldsymbol{s}_i$, the parameter server could decrypt the gradients according to

$$\bar{\boldsymbol{v}}_i = \begin{cases} \hat{\boldsymbol{v}}_i + \boldsymbol{s}_i, & \text{if } \langle \hat{\boldsymbol{v}}_i, \boldsymbol{s}_i \rangle < 0 \\ \boldsymbol{s}_i - \hat{\boldsymbol{v}}_i, & \text{if } \langle \hat{\boldsymbol{v}}_i, \boldsymbol{s}_i \rangle > 0 \end{cases} \tag{4}$$

As a result, the decrypted gradients derived by the parameter server will become

$$\bar{\boldsymbol{v}}_i = \frac{|\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle|}{\|\boldsymbol{v}_i\|^2} \boldsymbol{v}_i \tag{5}$$

Though this is not exactly the same as the original gradients, $\boldsymbol{v}_i$, when all workers are hosting IID data, QuGrad will only impact the learning rate but not the convergence of training process (see details in Theorem 1).

Non-IID training data. When all workers are hosting non-IID data, the parameter server has to recover the exact gradients from each worker. According to (5), the decrypted gradient vector $\bar{\boldsymbol{v}}_i$ is a scaling of $\boldsymbol{v}_i$ with the factor $\frac{|\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle|}{\|\boldsymbol{v}_i\|^2}$. Since $\frac{|\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle|}{\|\boldsymbol{v}_i\|^2} = \frac{\bar{v}_{ik}}{v_{ik}}$ for all worker i and all gradient index k, each worker i can send one of its gradients $v_{ik} \neq 0$ to the parameter server, and the latter can recover the original gradients as $\boldsymbol{v}_i = \frac{v_{ik}}{\bar{v}_{ik}} \bar{\boldsymbol{v}}_i$. However, in QuGrad, worker i would not send such a plain value to the parameter server as long as it does not impact the convergence of the training process, since it may incur additional privacy leakage.

C. Theoretical Analysis

As discussed earlier, the parameter server may only get a scaled gradient vector from each worker and the QKD may suffer non-negligible QBER. In this section, we will analyze how these two issues impact the performance of model training in CSFL through the following two theorems. Due to the space limitation, we omit the proofs of these two theorems and we will show them in our journal version.

Theorem 1. *When every worker hosts IID training data, though the parameter server in QuGrad cannot recover the original gradients, it only impacts the learning rate but not the convergence of the training procedure.*

Theorem 2. *When the QBER is less than $\frac{1}{2}$, QuGrad will ensure the convergence of the training process.*

D. Practical Discussions

Match the QKD and data transmission rates. In QuGrad, one bit of secret keys is needed to encrypt one gradient. With a gradient transmission rate of B bps, and each gradient as an FP32 number, the secret key distribution rate should be at least $B/32$ bps. In a WAN with a typical rate up to 150 Mbps, the QKD rate needed is 4-5 Mbps. Though current SOTA QKD technology [10] can meet this requirement experimentally, multiple QKD devices are needed for practical deployment, which is costly but feasible for large organizations.

To further reduce the cost to distribute secret keys, gradients can be pruned, *i.e.*, small gradients set to 0 and not encrypted. This can significantly decrease the keys needed without impacting model accuracy [33, 34]. Additionally, increasing the batch size can reduce the gradients generated and transmitted, lowering the secret key distribution rate requirement.

When DLG algorithms minimize l_2 norm distance. Some DLG algorithms, such as [3], minimize the l_2 norm distance instead of the cosine distance between the original and recovered gradients [3]. In such cases, scaling the original gradients, as done in QuGrad (*i.e.*, scaling with a factor of $\frac{\langle \boldsymbol{v}_i, \boldsymbol{s}_i \rangle}{\|\boldsymbol{v}_i\|^2}$), can help preserve data privacy. Section V-C demonstrates that QuGrad maintains good performance in preserving training data privacy even when minimizing the l_2 norm distance.

V. Experimental Evaluation

We experimentally evaluate QuGrad using a QKD-based testbed running federated learning tasks. The key findings are:

- QuGrad can eliminate data privacy leakage from gradients in CSFL systems even if the attacker knows the training data characteristics and adopts a SOTA DLG algorithm.
- QuGrad can increase the average LPIPS between the recovered image and the original one by up to $2.3\times$, reduce the Jaccard similarity [15] and HaarPSI [16] by up to 89% and 69%, respectively, compared with the SOTA defensive scheme to combat DLG.
- QuGrad will not impact the convergence rate of the training process and the accuracy of the derived ML model, even if the QBER is as large as 10%.

A. Implementation

We implement QuGrad on a testbed with four hosts connected by an Ethernet switch and three pairs of QKD devices. Three of the hosts, each of which is carrying a NVIDIA RTX 3090 GPU, perform as workers and the remaining one works as the parameter server. To emulate a WAN environment, we limit the available bandwidth of each of the switch ports connecting to a worker as 50 Mbps, while the port connecting to the parameter server is 150 Mbps.

Between each worker and the parameter server, we allocate a pair of QKD devices to distribute secret keys. Each pair of QKD devices are connected through a 50 km fiber (*i.e.*, quantum channel). In our experiment environment, each pair of QKD devices can distribute secret keys at the rate of about 1.5 Mbps over a 50 km fiber. Since a quantum error correction scheme is enforced in our QKD devices, the QBER is as low as 10^{-5} on our testbed. For every QKD device, we implement a key manager to collect secret keys at its corresponding host. As a result, there are three key managers at the parameter server and every worker hosts only one key manager. The two key managers associated with the same pair of QKD devices will receive the same sequence of secret keys. During each experiment, a key manager is continuously collecting secret keys from the corresponding QKD device. The unused keys will be stored in the key manager for future use.

In each training epoch, when a worker has prepared its gradients, it will fetch secret keys from the key manager to encrypt the gradients and send the encrypted gradients to the parameter server through a classic channel. If there are not enough secret keys, the training process (*i.e.*, the gradient transmission process) will be held until enough keys are collected. When the parameter server receives a batch of encrypted gradients from a worker, it will fetch secret keys from the associated key manager and decrypt these gradients. It should be noted that there will always be enough secret keys in the parameter server's key managers. When gradients from all workers are received and decrypted, the parameter server will update the global parameters and push the new parameters back to all workers to invoke the next training epoch.

B. Methodology

Setting up. We have conducted experiments to investigate the performance of QuGrad in two aspects: (i) the performance to defending training data privacy against DLG; (ii) how QuGrad will impact the performance of CSFL, *e.g.*, the convergence rate and the accuracy of the derived model. For the first purpose, we will investigate the performance of QuGrad with CelebA [35] and ImageNet [32]. The first data set has about 200 thousand pictures of people's faces, which have been categorized into males and females, while the second data set has a thousand classes of ten million pictures. In the former data set, the data have much fewer features than that in the latter one. We compare QuGrad with two SOTA schemes to defend data privacy against DLG, *i.e.*, FedCDP [13] and Soteria [12]. After enforcing different defensive schemes to gradients, we will leverage ROGS and GGL to recover the training data. Since both ROGS and GGL need a GAN to recover training data, when we conduct experiments with ImageNet, we use the GAN provided in [36] and when conducting experiments with CelebA, we train a GAN based on CelebA ourselves. Without loss of generality, the gradients will be generated by training a ResNet18 [31] model.

For the second purpose, we will train two models, *i.e.*, ResNet18 and ShuffleNet V2 [37], based on CIFAR100 [38] and miniImageNet [39], respectively. In both cases, we evenly distribute training data among all three workers and train a uniform model based on FedAvg, and observe how the training process will be impacted by the QBER.

When either ROGS or GGL is used to recover the training data, for a conservative evaluation and showing the generality

TABLE II
VISUALIZED RESULTS BASED ON CELEBA

Original	DEF / ATK	None	Soteria	FedCDP	QuGrad
	GGL				
	ROGS				
	GGL				
	ROGS				

TABLE III
VISUALIZED RESULTS BASED ON IMAGENET

Original	DEF / ATK	None	Soteria	FedCDP	QuGrad
	GGL				
	ROGS				
	GGL				
	ROGS				

of QuGrad, we assume a DLG algorithm always knows the ground truth labels of training data. Without this assumption, both ROGS and GGL will suffer from errors of recovering labels and get less useful (or correct) information about the training data from the gradients. Correspondingly, in practice, QuGrad actually will achieve more performance improvement than that we will present in the following.

Metrics. We will use the following metrics to evaluate the performance of a defensive scheme. All of the following metrics are used to evaluate the similarity of two pictures. Accordingly, we will test the following metrics by comparing a picture recovered by different DLG algorithms according to the gradients protected by different defensive schemes with the original picture (*i.e.*, the ground truth).

- Learned Perceptual Image Patch Similarity (LPIPS) [14], also called perceptual loss. This is a metric to quantify the difference of two figures according to human's natural perception. A small LPIPS indicates that a human being will feel two comparing figures are similar to each other.
- Jaccard similarity [15]. Suppose A and B are the set of features for two figures, then the Jaccard similarity of these two figures is calculated as $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. The Jaccard similarity score is a value between 0 and 1. The larger the Jaccard similarity score is, the two figures are more similar.
- HaarPSI [16]. This is a metric to evaluate the similarity of two pictures from the signal processing perspective. The HaarPSI utilizes the coefficients obtained from a Haar wavelet decomposition to assess local similarities between two pictures, as well as the relative importance of image areas. The same to the previous evaluation metric, two figures similar to each other will result in a large HaarPSI.

C. Experiment results

Microcosmic performance. To recover a training data with a DLG algorithm, such as ROGS and GGL, it will take 2-3 hours on a single server carrying a NVIDIA RTX 3090 GPU. Due to such a large time complexity, we cannot try to recover all pictures in the two training data sets used in our experiments. To address this issue, we randomly select 100 training data samples in each of the two training data sets for testing. For a clear observation of the performance of QuGrad in a microcosmic perspective, we first pick out two pictures from each of CelebA and ImageNet, respectively, to visualize the performance of different defensive schemes to combat different DLG algorithms in Tabs. II & III. From these two tables, we can make the following observations.

At first, without deploying any defensive scheme, both ROGS and GGL can derive a picture very similar to the original one in most cases. Even if a DLG algorithm cannot derive the exact original picture (*e.g.*, the first test case in Tab. III), we can identify a lot of critical information from the recovered picture.

Secondly, when the training data has few features, *i.e.*, for the data in CelebA, an (existing) defensive scheme may help hide more critical information in gradients (compared with the case that no defensive scheme is enforced). Either Soteria or FedCDP may achieve a performance better than the other one in defending data privacy when different DLG algorithms are adopted, however, neither of them can defend data privacy against all existing DLG algorithms. For example, in Tab. II, we can observe that Soteria can better defend data privacy against GGL, while FedCDP would be a better defensive scheme to combat ROGS. Neither Soteria nor FedCDP is the better defensive scheme against both GGL and ROGS.

Thirdly, when the training data has many features, *i.e.*, for the data in ImageNet, neither existing defensive scheme can help hide much more data privacy compared with the case that no defensive scheme is adopted as shown in Tab. III. This is because more features will provide more hints for a DLG algorithm to distill data privacy, especially when the training data feature distribution is known in advance.

Lastly, neither ROGS nor GGL can recover any valuable useful information from the gradients once QuGrad is enforced. Though GGL can sometimes generate a picture containing some information of the original one, for example, GGL can generate a human face with the same gender as that in the original picture, this is because all training samples in CelebA are human faces and the GAN used in GGL can learn this fact and always generate a face-like picture. In addition, for conservative evaluation and showing the generality of QuGrad, we input the ground truth labels to the GAN in GGL. Therefore, the human face on every image recovered by GGL

TABLE IV
PERFORMANCE EVALUATION IN LPIPS.

Dataset	CelebA				ImageNet			
Trained	Yes		No		Yes		No	
DEF \ ATK	GGL	ROGS	GGL	ROGS	GGL	ROGS	GGL	ROGS
None	0.1650(0.1424)	**0.8128**(0.8305)	0.1262(0.1497)	0.2990(0.3743)	0.5713(0.6750)	0.7428(0.7154)	0.5684(0.5572)	0.3222(0.3579)
Soteria	0.2716(0.4591)	0.7877(**0.8376**)	0.3773(0.3386)	0.2977(0.3706)	0.6263(0.6286)	0.7808(0.8135)	0.6245(0.6042)	0.3195(0.3561)
FedCDP	0.3420(0.5977)	0.7941(0.8007)	0.2150(0.1681)	0.4292(0.4461)	0.6669(0.6561)	**0.7916**(0.8169)	0.6769(0.5890)	0.3437(0.4039)
QuGrad	**0.5993**(**0.6348**)	0.8087(0.7880)	**0.7009**(**0.6993**)	**0.7901**(**0.8602**)	**0.7191**(**0.7120**)	0.7238(**0.8248**)	**0.8746**(**0.7977**)	**0.8633**(**0.8035**)

shows the correct gender. However, the faces generated by GGL are absolutely different from the original ones. When a DLG algorithm does not directly generate a picture by using a GAN, *e.g.*, ROGS who only leverages a GAN to improve the resolution of the output of iDLG, it cannot get any useful information from the gradients encrypted using QuGrad.

Macroscopic performance. To investigate the overall performance of QuGrad in an entire data set, we randomly pick out 100 training samples from each of the training data sets and calculate the three metrics at the beginning of the training process and after 100 training epochs, respectively. The average metrics are shown in Tabs. IV–VI. In these tables, every item consists of two values. The performance metric values outside of brackets are derived when a DLG algorithm minimizes the cosine distance, while the values in brackets are derived by minimizing l_2 norm distance. The property "trained" is "Yes" (or "No") means a DLG algorithm recovers training data from the gradients of a model that has been trained for 100 epochs (or at the beginning of the training).

From these tables, we can observe that in most scenarios, QuGrad achieves the largest average values of LPIPS, and the smallest values of Jaccard similarity and HaarPSI among all comparison defensive schemes. Especially in the CelebA data set, compared with the case without any defensive schemes, QuGrad can help increase the average ILIPS by $4.6\times$, and reduce Jaccard similarity and HaarPSI by 92% and 69%, respectively, at the beginning of the training process (*i.e.*, the model has not been trained). Even compared with the SOTA defensive schemes (such as Soteria and FedCDP), QuGrad can increase the average ILIPS by $2.3\times$, and reduce Jaccard similarity and HaarPSI by up to 89% and 69%, respectively. In other words, with QuGrad, the data recovered from the gradients protected by QuGrad will be further away from (*i.e.*, less similar to) the original one than that recovered from the gradients protected by SOTA schemes to combat DLG.

There are some exceptions that Soteria or FedCDP achieves the best performance. Even in some cases, for example, when ROGS is used to recover the training data in CelebA from the gradients of a trained model, disabling all defensive schemes would achieve the best performance metric (see the values outside of brackets in the second column in Tab. IV). This is because the gradients of a model that has been trained for 100 epochs themselves contain little information about the training data and a DLG algorithm cannot distill too much data privacy from these gradients (verified in [40]). As a result, all defensive schemes have the similar performance. It is demonstrated by the observation that when QuGrad is not the best in some performance metric, all defensive schemes incur the similar performance metric values.

Another observation that we can make from Tabs. IV–VI is that compared with ROGA, GGL can derive a picture more similar to the original one, regardless of which defensive scheme is adopted and whether or not the gradients are from a trained model. In the meanwhile, when GGL is used to recover the training data, QuGrad will always be the best defensive scheme to protect data privacy. This shows that QuGrad helps hide the most useful information of the training data among all comparison defensive schemes.

Extend to l_2 norm distance. So far, we only discussed that case that a DLG algorithm minimizes the cosine distance between the encrypted gradients and that associated with the recovered data in order to recover the training data. However, a DLG algorithm can also adopt other distances, such as the l_2 norm distance, to evaluate the quality of a recovered data. To demonstrate that QuGrad is robust to the distance metric based on which a DLG algorithm recovers training data, in Tabs. IV–VI, we also show the macroscopic performance of QuGrad when ROGS and GGL minimize the l_2 norm distance in order to recover training data from gradients. The corresponding performance metric values are shown in brackets. Again, at the beginning of the training procedure, all defensive schemes will be able to increase the average value of LPIPS and decrease the values of Jaccard similarity and HaarPSI. QuGrad is always the best scheme to defend data privacy against DLG. When training an ML model with the CelebA data set and GGL is used to recover the training data, QuGrad can help increase the average ILIPS by $1.3\times$, and reduce Jaccard similarity and HaarPSI by 92% and 51%, respectively, compared with Soteria. Compared with FedCDP, QuGrad can increase the average ILIPS by $3.2\times$, and reduce Jaccard similarity and HaarPSI by 90% and 69%, respectively.

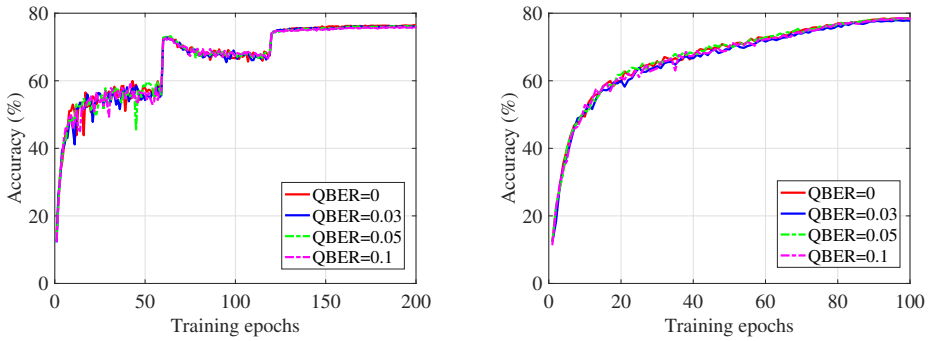
When the model has been well-trained (trained for 100 epochs in our experiments), disabling all defensive schemes would lead to the minimal HaarPSI value. This only appears when ROGS is used to recover training data and all defensive schemes lead to the similar (and small) HaarPSI value. Since a DLG cannot achieve a good performance even if no defensive scheme is adopted, QuGrad may incidentally lead

TABLE V
PERFORMANCE EVALUATION IN JACCARD SIMILARITY.

Dataset	CelebA				ImageNet			
Trained	Yes		No		Yes		No	
ATK / DEF	GGL	ROGS	GGL	ROGS	GGL	ROGS	GGL	ROGS
None	0.4852(0.4291)	0.0282(0.0279)	0.5066(0.3596)	0.3677(0.0890)	0.4848(0.5397)	0.1270(0.1325)	0.5569(0.4889)	0.2841(0.2007)
Soteria	0.4122(0.5376)	**0.0278**(0.0274)	0.3900(0.4322)	0.3984(0.0690)	0.5491(0.5292)	**0.0667**(**0.1250**)	0.4800(0.4817)	0.2902(0.2407)
FedCDP	0.3508(0.3541)	0.0299(0.0279)	0.4250(0.4152)	0.2323(0.0929)	0.4928(0.4656)	0.1181(0.1339)	0.4484(0.4900)	0.2973(0.2365)
QuGrad	**0.1590**(**0.2451**)	0.0290(**0.0267**)	**0.0450**(**0.0410**)	**0.0280**(**0.0303**)	**0.3632**(**0.3635**)	0.1422(0.1333)	**0.1944**(**0.2252**)	**0.1694**(**0.0509**)

TABLE VI
PERFORMANCE EVALUATION IN HAARPSI.

Trained	CelebA				ImageNet			
Model	Yes		No		Yes		No	
ATK / DEF	GGL	ROGS	GGL	ROGS	GGL	ROGS	GGL	ROGS
None	0.6220(0.6955)	0.1846(**0.1399**)	0.6740(0.6760)	0.4899(0.4429)	0.3317(0.2933)	0.3138(0.3325)	0.2855(0.2380)	0.5467(0.5171)
Soteria	0.4273(0.2750)	**0.1814**(0.1540)	0.3237(0.3798)	0.4942(0.4429)	0.2784(0.2710)	0.3006(0.2871)	0.2651(0.3203)	0.5460(0.5161)
FedCDP	0.3826(0.2133)	0.1954(0.1861)	0.5423(0.6098)	0.4114(0.3619)	0.2849(0.2513)	**0.2937**(0.2838)	0.2243(0.3165)	0.5280(0.4867)
QuGrad	**0.2424**(**0.1689**)	0.2009(0.1860)	**0.2204**(**0.1869**)	**0.1525**(**0.0821**)	**0.2646**(**0.2481**)	0.3080(**0.2757**)	**0.1678**(**0.2220**)	**0.2218**(**0.1190**)



(a) Train a ResNet18 based on CIFAR100. (b) Train a ShuffleNet based on mini-ImageNet.

Fig. 1. How QBER impacts the performance of the model training process.

to a recovered data more similar to the real one (in terms of the evaluation metrics). However, there would be little useful information in the recovered data samples when they have such small similarities with the original ones. When a DLG algorithm has potential to distill useful information from the gradients, *e.g.*, when GGL, who can achieve a smaller LPIPS, and larger Jaccard similarity and HaarPSI, is adopted to recover training data, QuGrad will always the best defensive scheme compared with other counterpart schemes even if the l_2 norm distance is used in DLG algorithms.

Performance of model training. Since a quantum error correction scheme is adopted in our QKD devices, we will experience a low QEBR. To investigate the impact of QEBR to the performance of model training process, we randomly flip the keys distributed to the parameter server with one of the probabilities in $\{0.03, 0.05, 0.1\}$. The simulation results are shown in Fig. 1. From both test cases, we can see that even if the QBER is as large as 0.1, *i.e.*, 10%, the convergence rate of the training process and the accuracy of the derived model will not be significantly impacted. This demonstrates the robustness of QuGrad to the QEBR.

VI. CONCLUSIONS

In this work, we have proposed QuGrad, the first-of-its-kind Quantum Key Distribution (QKD) based gradient encryption approach to defending the training data privacy against leakage from gradients in Cross-Silo Federated Learning (CSFL) systems. Though One-Time Pad (OTP) is proven to be the only perfect encryption technique, current QKD system cannot distribute secret keys at the rate of data transmission in a practical Wide Area Network (WAN). To overcome the challenge due to the limited QKD rate, QuGrad uses a simple yet elaborate scheme to encrypt every gradient using only one bit of the secret keys based on QKD. Thanks to several salient features of the encrypted gradients, *e.g.*, it is unpredictable which or how many bits of each gradient will be changed and the encrypted gradient vector will be orthogonal to the original one, QuGrad is effective in preserving the privacy of the training data. We have also implemented QuGrad on a real testbed. Through extensive experiments, we have demonstrated that an attacker cannot recover the training data from the encrypted gradients in QuGrad, even if it knows the features and distribution of the training data set in advance.

REFERENCES

[1] N. Zhang, Q. Ma, and X. Chen, "Enabling long-term cooperation in cross-silo federated learning: A repeated game perspective," *IEEE Transactions on Mobile Computing*, 2022.

[2] M. Tang and V. W. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *IEEE INFOCOM*, 2021.

[3] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.

[4] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *CoRR*, vol. abs/2001.02610, 2020. [Online]. Available: http://arxiv.org/abs/2001.02610

[5] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 332–16 341.

[6] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, 1984.

[7] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical review letters*, vol. 67, no. 6, pp. 661–663, 1991.

[8] C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, October 1949.

[9] TOSHIBA, "QKD System Specifications," 2022. [Online]. Available: https://www.global.toshiba/ww/products-solutions/security-ict/qkd/products.html

[10] Z. Yuan, A. Plews, R. Takahashi, K. Doi, W. Tam, A. Sharpe, A. Dixon, E. Lavelle, J. Dynes, A. Murakami, M. Kujiraoka, M. Lucamarini, Y. Tanizawa, H. Sato, and A. J. Shields, "10-mb/s quantum key distribution," *J. Lightwave Technol.*, vol. 36, no. 16, pp. 3427–3433, Aug 2018.

[11] C. BasuMallick, "Wide area network (wan) vs. local area network (lan): Key differences and similarities," *Spiceworks*, August 2022. [Online]. Available: https://www.spiceworks.com/tech/networking/articles/wide-area-network-vs-local-area-network-differences-and-similarities/

[12] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[13] W. Wei, L. Liu, Y. Wut, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2021, pp. 797–807.

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features As a Perceptual Metric," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[15] K. Yue, R. Jin, C.-W. Wong, DrorBaron, and H. Dai, "Gradient obfuscation gives a false sense of security in federated learning," in *USENIX Security*, 2022.

[16] R. Reisenhofer, S. Bosse, G. Kutyniok, and T. Wiegand, "A haar wavelet-based perceptual similarity index for image quality assessment," *Signal Processing: Image Communication*, vol. 61, 2018.

[17] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *IEEE Conference on Computer Communications (INFOCOM)*, April 2018, pp. 63–71.

[18] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, "Scaling distributed machine learning with the parameter server," in *Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation*, 2014, pp. 583–598.

[19] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48, 2016, pp. 201–210.

[20] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 19–38.

[21] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of the USENIX ATC*, 2020.

[22] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu, "A framework for evaluating client privacy leakages in federated learning," in *ESORICS*, 2020, p. 545–566.

[23] B. Wang, F. Wu, Y. Long, L. Rimanic, C. Zhang, and B. Li, "Datalens: Scalable privacy preserving training via gradient compression and aggregation," in *Proceedings of the ACM CCS*, 2021, p. 2146–2168.

[24] D. Yu, H. Zhang, W. Chen, and T.-Y. Liu, "Do not let privacy overbill utility: Gradient embedding perturbation for private learning," *International Conference on Learning Representations (ICLR)*, 2021.

[25] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 304–317.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54, 2017, pp. 1273–1282.

[27] Z. Li, J. Zhang, L. Liu, and J. Liu, "Auditing privacy defenses in federated learning via generative gradient leakage," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[28] W.-Y. Liu, X.-F. Zhong, T. Wu, F.-Z. Li, B. Jin, Y. Tang, H.-M. Hu, Z.-P. Li, L. Zhang, W.-Q. Cai, S.-K. Liao, Y. Cao, and C.-Z. Peng, "Experimental free-space quantum key distribution with efficient error correction," *Opt. Express*, vol. 25, pp. 10 716–10 723, May 2017.

[29] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Annual Conference on Neural Information Processing Systems*, 2017, pp. 1508–1518.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1097–1105.

[33] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *International Conference on Learning Representations (ICLR)*, 2018.

[34] Y. Zhao, J. Fan, L. Su, T. Song, S. Wang, and C. Qiao, "SNAP: A communication efficient distributed machine learning framework for edge computing," in *2020 IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2020.

[35] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[36] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," in *International Conference on Learning Representations (ICLR)*, 2019.

[37] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *15th European Conference on Computer Vision (ECCV)*, 2018.

[38] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, and X. Wang, "Applied machine learning at facebook a datacenter infrastructure perspective," 2017. [Online]. Available: https://research.fb.com/wp-content/uploads/2017/12/hpca-2018-facebook.pdf

[39] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[40] F. Wang, E. Hugh, and B. Li, "More than enough is too much: Adaptive defenses against gradient leakage in production federated learning," in *IEEE INFOCOM*, 2023, pp. 1–10.