Analyzing User-Level Privacy Attack Against Federated Learning

Mengkai Song, *Graduate Student Member, IEEE*, Zhibo Wang, *Senior Member, IEEE*, Zhifei Zhang, *Member, IEEE*, Yang Song, *Member, IEEE*, Qian Wang, *Senior Member, IEEE*, Ju Ren, *Member, IEEE*, and Hairong Qi, *Fellow, IEEE*

***Abstract*— Federated learning has emerged as an advanced privacy-preserving learning technique for mobile edge computing, where the model is trained in a decentralized manner by the clients, preventing the server from directly accessing those private data from the clients. This learning mechanism significantly challenges the attack from the server side. Although the state-of-the-art attacking techniques that incorporated the advance of Generative adversarial networks (GANs) could construct class representatives of the global data distribution among all clients, it is still challenging to distinguishably attack a specific client (i.e., user-level privacy leakage), which is a stronger privacy threat to precisely recover the private data from a specific client. To analyze the privacy leakage of federated learning, this paper gives the first attempt to explore user-level privacy leakage by the attack from a malicious server. We propose a framework incorporating GAN with a multi-task discriminator, called multi-task GAN – Auxiliary Identification (mGAN-AI), which simultaneously discriminates category, reality, and client identity of input samples. The novel discrimination on client identity enables the generator to recover user specified private data. Unlike existing works interfering the federated learning process, the proposed method works "invisibly" on the server side. Furthermore, considering the anonymization strategy for mitigating mGAN-AI, we propose a beforehand linkability attack which re-identifies the anonymized updates by associating the client representatives. A novel siamese network fusing the identification and verification models is developed for measuring the similarity of representatives. The experimental results demonstrate the effectiveness of the proposed approaches and the superior to the state-of-the-art.**

***Index Terms*— Federated learning, user-level privacy, reconstruction attack, linkability attack.**

I. Introduction

INCREASINGLY, modern deep learning technique is beginning to emerge in the networking domain, such as a crowdsourced system for learning tasks. But utilizing conventional centralized training methodology requires local storage of the crowdsourced data, which suffers from the high volume of traffic, highly computational demands and privacy concerns [1]. For collectively reaping the benefits of the shared model trained from this rich data without the need to store it centrally, distributed learning framework was proposed, serving as a mobile edge computing framework for deep learning. Shokri and Shmatikov [2] proposed the collaborative learning Distributed Selective Stochastic Gradient Descent (DSSGD), where the data providers, i.e., clients, train locally on a shared model, and then the server will collect those local models/updates to estimate/update a global model instead of directly access to the private data from clients. Then, the global model is sent back to clients, iterating the aforementioned local training process. In the same token, federated learning [3] proposed a variant of decentralized learning with higher efficiency. The key improvement lies in the way of updating the global model, i.e., DSSGD performs the aggregated update while the federated learning conducts the averaged update, which is more suitable for the commonly non-IID (i.e., non-independent and identically distributed) and unbalanced data distribution among clients in the real world. Fig. 1 illustrates the framework of federated learning.

However, recent works demonstrated that the collaborative learning is vulnerable to the inference attack, e.g., reconstruction attack and membership attack, by malicious servers/clients because the shared model is updated according to those private data, whose pattern is encoded into the model parameters. Therefore, if a corresponding decoder could be constructed,

Manuscript received October 1, 2019; revised February 15, 2020; accepted March 31, 2020. Date of publication June 5, 2020; date of current version September 16, 2020. This work was supported in part by the National Natural Science of China under Grant 61872274, Grant 61702562, Grant 61822207, Grant U1636219, and Grant U19A2067, in part by the Equipment Pre-Research Joint Fund of Ministry of Education of China (Youth Talent) under Grant 6141A02033327, in part by the National Key Research and Development Program of China under Grant 2019YFA0706403, in part by the Natural Science Foundation of Hubei Province under Grant 2017CFB503 and Grant 2017CFA047, in part by the Young Tlite Scientists Sponsorship Program by CAST under Grant 2018QNRC001, in part by the Young Talents Plan of Hunan Province of China under Grant 2019RS2001, and in part by the Fundamental Research Funds for the Central Universities under Grant 2042018gf0043 and Grant 2042019gf0098. *(Corresponding author: Zhibo Wang.)*

Mengkai Song, Zhibo Wang, and Qian Wang are with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: mksong@whu.edu.cn; zbwang@whu.edu.cn; qianwang@whu.edu.cn).

Zhifei Zhang, Yang Song, and Hairong Qi are with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996 USA (e-mail: zzhang61@vols.utk.edu; ysong18@vols.utk.edu; hqi@vols.utk.edu).

Ju Ren is with the Department of Computer Science, Tsinghua University, Beijing 100084, China, and also with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: renju@csu.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2020.3000372

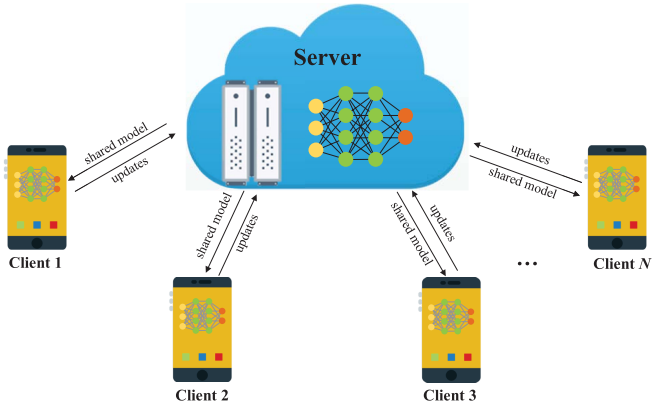


Fig. 1. The framework of federated learning. The server sends the shared model to each client, which trains the shared model locally by its private data. Then, the updates are uploaded to the server, where those updates are aggregated/averaged to improve the shared model.

the private data or statistics will be recovered inversely. Under the assumption of a malicious server, Aono *et al.* [4] partially recovered the private data based on the observation that the gradient of weights is proportional to that of the bias at the first layer of the model, and their radio approximates to the training input. But it could only apply to a simplified setting where the training batch has only one sample. Hitaj *et al.* [5] proposed a GAN-based reconstruction attack against the collaborative learning by assuming a malicious client, which utilized the shared model as the discriminator to train a GAN [6]. Eventually, the learned generator (equivalent to the decoder) will successfully mimic the training samples. Although [5] had demonstrated the effectiveness of GAN-based attack against DSSGD, it presents mainly three limitations: 1) It would change the architecture of the shared model and introduce adversarial influence to the learning process, which unrealistically assumes a powerful malicious client and compromises the learning process. 2) The GAN-based attack suffers from performance degradation when attacking the federated learning because the adversarial influence introduced by the malicious client would become trivial after the average update. 3) In addition, the GAN-based attack could only infer class-wised representatives, which are generic samples characterizing class properties rather than the exact samples from the clients [7].

In this paper, we focus on the privacy issues of federated learning, and analyze its limitations from the perspective of attack. It helps to design more dependable and secure federated learning framework. Motivated by those drawbacks in existing reconstruction attacking techniques, we propose a more generic, practical, and invisible attacking framework against the federated learning without affecting the learning process. In addition, we further target client-level privacy,[1] which is more challenging and empirically meaningful than globally recovering the data representatives that characterize the population distribution as in the GAN-based attack. Instead of assuming a malicious client, the proposed method performs on the server side, which first explores the attack by a malicious server. Specifically, we adapt the original GAN to a multi-task scenario where reality, category, and identity of the target client are considered simultaneously. On the one hand, performing additional tasks when training the GAN improves the quality of synthesized samples [8] without the necessity of modification on the shared model or compromising to the federated learning, achieving invisible attack. On the other hand, the novel task of discriminating client identity enables client-level privacy recovery. For simplicity, the proposed framework is referred to as multi-task GAN for Auxiliary Identification (mGAN-AI).

The key novelty of mGAN-AI is to compromise the client-level privacy that is defined as the client-dependent properties, which specifically characterize the corresponding client and distinguish the client from the others. The key of achieving the discrimination on client identity is to obtain data representatives from each individual client, such that the training of GAN can be supervised by the identity representatives, generating samples with specific identity (i.e., from a specific client). Since the client data is unaccessible, we adopt an optimization-based method to estimate such data representatives from those accessible updates from the clients. The mGAN-AI relaxes the assumption held by existing works that clients own mutually exclusive class labels, achieving a more generic attack to the federated learning.

Compared with our previous work [9], we further consider the enhancement of the proposed mGAN-AI attack against the potential mitigation strategies. When the malicious server performs the attack, the label of the periodic updates, i.e., the client's identity, is required for both the victim and other clients. However, the clients can transmit the model updates without identities over an anonymity network such as Tor or via a trusted third party, to safeguard the identities [3], which would result in the failure of mGAN-AI in reconstructing user-level privacy. To solve this problem, we further propose a beforehand linkability attack, which re-identifies the anonymized model updates by associating the data representatives from different clients. More specifically, we develop a convolutional siamese network fusing the identification and verification models together for measuring the similarity of two given representatives. The joint model learns a more discriminative embedding of client representative, which contributes to more precise matching. For training the model, we propose a shadow federated learning mechanism for collecting the training data, i.e., pairs like (representative, client identity), which re-implements the distributed training with an auxiliary dataset. The advance of the proposed joint model in matching data representatives remains when applied to new dataset. It shows obvious advantage over the existing works, which requires additional knowledge about the identities of a part of clients. In the experiments, we show that the proposed linkability attack succeeds with over 99% accuracy on two benchmark datasets.

In summary, our contributions are mainly in four-fold:

- To the best of our knowledges, we are the first to analyze the privacy issues of federated learning by exploring the attack from a malicious server. In addition, beyond inferring class-wised representatives, we step further to recover user-level privacy in an invisible manner.

[1]In this paper, "user-level" and "client-level" are used interchangeably.

- We propose the generic attack framework mGAN-AI that incorporates a multi-task GAN, which conducts a novel discrimination on user identity, achieving user-level privacy leakage.
- We further propose a beforehand linkability attack against the potential anonymization strategy to mGAN-AI, which re-identifies the anonymized model updates by associating the data representatives from different clients.
- Exhaustive experimental evaluation is conducted to demonstrate the effectiveness and superior of the proposed mGAN-AI as compared to existing works. On the MNIST and AT&T datasets, mGAN-AI successfully recovers the samples from a specific user. Besides, the proposed linkability attack succeeds with over 99% accuracy under anonymous environment.

The rest of this paper is organized as follows. Section II introduces the related works on privacy-preserving distributed learning and attack models. Section III briefs the background knowledge. The threat models are discussed in Section IV, then the proposed mGAN-AI attack under non-anonymous environment and the linkability attack against anonymization strategy are detailed in Section V and VI. Extensive experimental evaluation is conducted in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

This section provides the background of privacy-preserving related learning methods, as well as attacking approaches.

A. Privacy-Preserving Distributed Learning

Distributed learning frameworks can be categorized according to the updating mechanism of the shared model. Reference [2] summarized the updates while [3], [10] averaged the updates, which outperform asynchronous approaches in aspects of communication efficiency and privacy preservation [11]. Existing works on privacy-preserving distributed learning mostly utilize either differential privacy mechanism (DP) or secure multi-party computation (MPC). Pathak *et al.* [12] proposed a DP-based global classifier by aggregating the locally trained classifiers. Hamm *et al.* [13] leveraged knowledge transfer from ensemble local models to a global differentially private model. DSSGD [2] was the first collaborative learning mechanism dealing with practical distribution, which used sparse vector [14] to realize DP. Zhao *et al.* [15] proposed a privacy-preserving collaborative deep learning framework by perturbing the objective function during the training process to achieve DP. Recently, participant-level differentially private distributed learning method [16] was proposed to further protect the membership information of participants. Utilizing MPC, Mohassel and Zhang [17] presented the SecureML framework where a global model is trained on the clients' encrypted data among two non-colluding servers. Other methods based on encryption include secure sum protocol [18] that uses a tree topology, homomorphic encryption, and secure aggregation protocol [19].

B. Attacks on the Learning Model

Different from the attacks on the prediction integrity of ML-based system, such as computer vision system [20], [21] and autonomous driving [22], this paper focuses on the privacy concerns. The inference attack, e.g., membership attack and reconstruction attack, aims to infer the private features of the training data against a learning model. The membership attack [7], [23] is to decide whether a given sample belongs to the training dataset. Shokri *et al.* [23] mounted the attack by utilizing the difference between model outputs from training and non-training inputs, respectively. Hayes *et al.* [24] used GAN to detect overfitting of the target model and recognize inputs through the discriminator which could distinguish different data distributions. Melis *et al.* [7] first proposed the membership attack against the collaborative/federated learning during its training and further inferred the properties by a batch property classifier. Concurrently, Nasr *et al.* [25] presented a comprehensive framework for the privacy analysis of federated learning, using white-box membership inference attacks.

The reconstruction attack aims to construct the training samples by accessing a learning model [4], [5], [26]–[28]. Fredrikson *et al.* [26] proposed the model inversion (MI) attack to recover images from a face recognition service by maximizing the confidence values with respect to a white noise image. Yang *et al.* [27] constructed the reconstruction attack by training a separate model that acts as an inverse of the original model, with black-box access. Against collaborative/federated learning framework, Aono *et al.* [4] partially recovered private data of the clients based on the proportionality between the training data and the updates sent to the server (assuming a malicious server). Hitaj *et al.* [5] assumed a malicious client in the federated learning, who has white-box access to the model at every training iteration. By utilizing GAN to synthesize samples, this attack successfully reconstructed the private data of other clients. Considering the privacy threat of inference data rather than training data, He *et al.* [28] proposed an inversion attack to recover the optimal samples by minimizing the distance of the target data's intermediate output.

This paper proposes a GAN-based reconstruction attack against the federated learning from the perspective of a malicious server, unlike related works that failed to threaten the federated learning in a practical and invisible manner. In detail, [4] could only be applied to a simplified setting where the training batch has one sample. Reference [5] modified the architecture of the shared model which is beyond the power of a malicious client, and it introduced negative influence into the standard training process, tending to worsen the shared model.

III. Preliminary

A. Federated Learning

Compared to the conventional training methods that require to directly access private data from clients, federated learning presents significant advantages in privacy preservation because of the distributed learning, where the clients share and train the model locally without uploading their private data to the server. Fig. 2 illustrates the proposed mGAN-AI, as well as

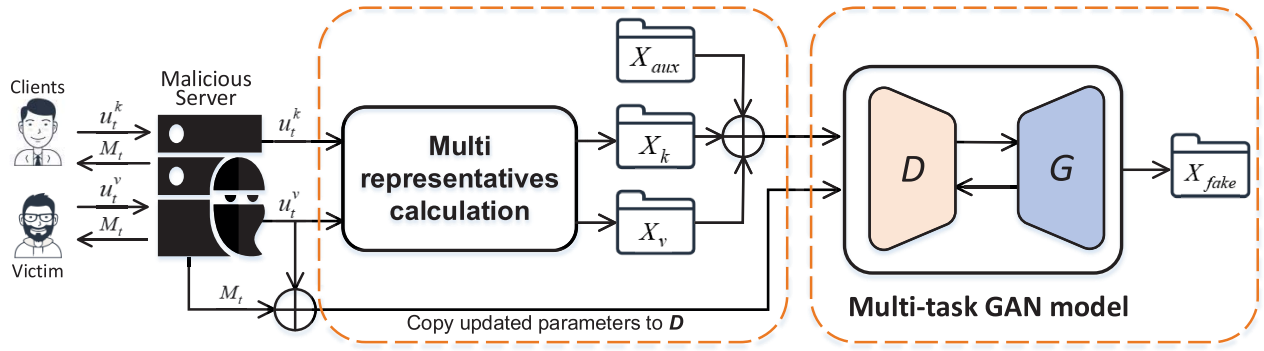


Fig. 2. Illustration of the proposed mGAN-AI from a malicious server in the federated learning. There are N clients, and the vth client is attacked as the victim. The shared model at the tth iteration is denoted as M_t, and u_t^k denotes corresponding update from the kth client. On the malicious server, a discriminator D (orange) and generator G (blue) are trained based on the update u_t^v from the victim, the shared model M_t, and representatives X_k, X_v from each client. X_{aux} denotes an auxiliary real dataset to train D on the real-fake task.

the federated learning. There are N clients, each of which owns its private dataset. During the learning, clients agree on a common objective and model structure. At each iteration, the parameters of current model are downloaded from the server to clients, and then the model is trained locally on each client. Finally, those local updates are sent back to the server, where the updates are averaged and accumulated to the current shared model. Eq. 1 expresses the update of the shared model.

$$M_{t+1} = M_t + \frac{1}{N}\sum_{k=1}^{N} u_t^k, \tag{1}$$

where M_t denotes the shared model at the tth iteration, and u_t^k indicates the update from the kth client at iteration t. [3] demonstrated that a valid model could be achieved by averaging the updates only if the clients receive the model with the same initialization. The federated learning satisfies this condition.

B. Generative Adversarial Networks

Generative adversarial nets (GANs) were first proposed by Goodfellow [6], which could generate samples indistinguishable from those training/real samples. It consists of a generator G and a discriminator D. The generator G draws random samples z from a prior distribution (e.g., Gaussian or uniform distribution) as the inputs, and then G generates samples from z. Given a training set, the discriminator D is trained to distinguish the generated samples from the training (real) samples. Eq. 2 shows the objective of GANs.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))], \tag{2}$$

where p_{data} and p_z denote the training/real distribution and prior distribution, respectively. The two models G and D are trained alternately until this minimax game achieves Nash equilibrium, where the generated samples are difficult to be distinguished from the real ones. The global optimum is achieved at $p_{data} = p_g$ [6], where p_g indicates the distribution of generated samples.

IV. Threat Model

This section introduces the threat model of the proposed mGAN-AI, then the generality and rationality of assuming a malicious server is demonstrated as compared to related works.

A. Learning Scenario

Following the training procedure of federated learning as described in Sec. III-A, we assume N ($N \geq 2$) clients that agree on a common learning objective and train collaboratively on a shared model. The data of all clients is considered as non-IID distributed, which is also consistent with the settings of federated learning [3]. Specifically, the data distribution of an arbitrary client k ($k = 1, 2, \cdots, N$) is independent from that of the other clients, and thus bias from the global distribution of all clients. The biased distribution p_k of the client k is considered as its client-level privacy. For simplicity, the shared model adopts an image classifier in the rest of this paper.

B. Malicious Server

The server in the federated learning is assumed to be malicious, aiming to reconstruct private data of the victim, which refers to the target client, for implementing the client-level privacy attack. The malicious server could either only analyze the periodic updates from the clients (i.e., *passive attack*), or even intentionally isolate the shared model trained by the victim to achieve more powerful attack (i.e., *active attack*). By contrast, most existing attacks against the collaborative/federated learning assumed malicious clients, which are limited at the stage of recovering class-wised representatives rather than prying client-level privacy because the malicious clients can only access aggregated updates (contributed by all the clients) from the server, while client-level attack requires individual update from each client. Although there are attacks targeting at certain clients [5], [7], they impractically assumed that class labels of the clients are mutually different or required extra information of the target client, e.g., class labels or other client-wise properties. More importantly, they would compromise the performance of the shared model. For

example, [7] changed the training objective, and [5] changed the shared model and introduced mislabeled samples into the training. Considering the limitation and drawback of malicious clients, we assume a malicious server that would overcome all above problems. The malicious server rigidly follows the rules of federated learning. Meanwhile, the passive and active attacks are performed without affecting the learning process, i.e., shared model, objective, updates, etc.

C. Communication Protocol

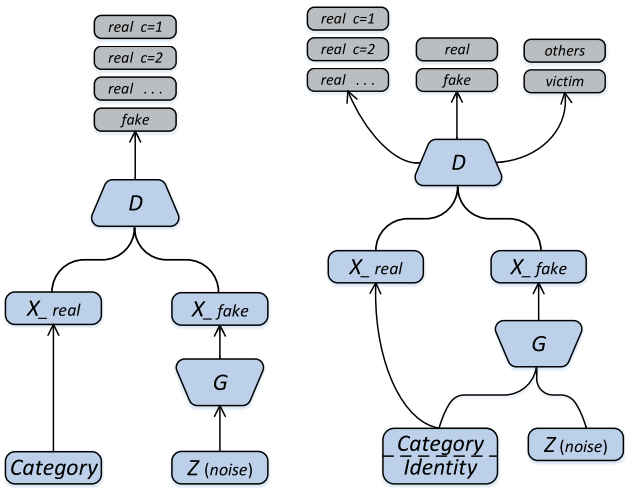
In our threat model, the parameters of the shared model are exchanged between the server and clients in plain text like [3], rather than in a encryption-based protocol [19], [29]. Intuitively, the encryption-based aggregation prevents the adversary from accessing the updates of the clients, thus preserving their privacy. However, recent works [5], [7], which only accessed the aggregated updates, could compromise the clients' privacy by introducing adversarial influence. In addition, the encryption-based aggregation also prevents the server from evaluating the utility of clients' updates [30], degrading the learning model when malicious clients exist. Reference [19] discussed the weakness of encryption-based protocol that it cannot guarantee the correctness of federated learning if there are malicious clients. Besides, anonymization strategy is generally adopted in the scenario of federated learning under privacy concerns [30], [31]. Actually the source of the updates is not needed in the aggregation algorithms, so the clients can transmit their updates without identification over a Tor network [32] or via a trusted third party. In this work, we consider both the non-anonymous and anonymous environment.

V. Approach Under Non-Anonymous Environment

This section details the proposed mGAN-AI attack against the federated learning for reconstructing private data of a specific victim. The main idea is to design a multi-task GAN that could discriminatingly learn the real data distribution of the victim. Section V-A gives a high-level overview of mGAN-AI, and Section V-B details the structure of the GAN with a novel multi-task discriminator, which first achieves discrimination on client identity. Then, sections V-C and V-D discuss the passive attack and active attack, respectively. The former performs in an invisible fashion, while the latter slightly interferes the share model but achieving more powerful attack. Finally, section V-E further details the inference of client-level privacy.

A. Overview of mGAN-AI

The principle of GAN is to train a discriminator on target data and generated data simultaneously. Eventually, the coupled generator would yield samples close to target ones. In the proposed mGAN-AI, the idea of training a GAN is borrowed, but the malicious server cannot access the target data (i.e., client-level private data) in the scenario of federated learning. However, the shared model is trained locally on each client, equivalent to training a discriminator (with the same structure as the shared model) on the target data. Intuitively, directly obtaining the target data to train a discriminator is equivalent to obtaining the network update after feeding the target data to the discriminator. In the federated learning, the only accessible information to the malicious server is the updates from clients, which provide the update to the discriminator after training on the target data. For the generated data, it can be easily obtained from the corresponding generator. Therefore, the advance of GAN could be utilized in the federated learning. In addition, since GANs could generate conditioned samples if supervised by the condition in training [33], we could train mGAN-AI conditoned on the updates from victim, thus it could generate victim-conditioned samples, i.e., client-level privacy.



(a) GAN-based attack (b) mGAN-AI

Fig. 3. Structure comparison between the GAN-based attack (a) [5] and mGAN-AI (b).

Fig. 2 overviews the proposed mGAN-AI. Assume N clients, and the vth client is the victim whose data would be reconstructed by the malicious server. The malicious server acts as a normal server in the federated learning, while at the same time it is also an adversary. At the tth iteration, the malicious server sends the current shared model M_t to each of the N clients and then receives the updates $u_t^1, u_t^2, \cdots, u_t^N$ from them after training on their private data. Specially, u_t^v denotes the update from the victim. For reconstructing the private data of the victim, we propose a new variant of GAN with a multi-task discriminator, which simultaneously discriminates category, reality, and client identity of input samples. Noted that the structure of the discriminator is the same as the shared model regardless of the output layer. Thus, the update u_t^v from the victim v could be either aggregated to the latest shared model and then to update the discriminator (passive attack) or aggregated to the discriminator directly (active attack). Besides, the data representatives of each client, which are synthesized samples that can obtain the same updates as the real samples after fed to the shared model, are calculated to supervise the training of D on identity. Then, the updated discriminator is trained on generated samples from the generator. Correspondingly, the generator is updated to approach the data from the victim. In the following, we will introduce the structure of the proposed multi-task GAN model.

B. Structure of the Multi-Task GAN

Specifically, the discriminator of mGAN-AI is designed to achieve three tasks: 1) real-fake discrimination like standard GAN, 2) categorization on input samples, which performs like the shared model, further increasing the quality of generated samples [8], and 3) identification on the input sample to distinguish the victim from other clients. The novel discrimination on identity is the key to violate user-level privacy. Fig. 3a and Fig. 3b present the model structures of GAN-based attack [5] and the proposed mGAN-AI. In GAN-based attack, an additional class "fake" is added to the output layer of the discriminator and the shared model, which is used for the adversary to train with generated samples by the generator. However, the modification is actually beyond the ability of a malicious client. In the proposed mGAN-AI, the discriminator D shares the structure with the shared model except the output layer because the output format of the three tasks are different. Compared to the GAN-based attack, mGAN-AI incorporates discrimination on identity that is also fed to G as an important condition. The network structure of D for each task in mGAN-AI is illustrated in the following,

$$\begin{aligned} D_{real} &= \text{Sigmoid}\left(\text{FC}_{real}(L_{share})\right) \\ D_{cat} &= \text{Softmax}\left(\text{FC}_{cat}(L_{share})\right) \\ D_{id} &= \text{Sigmoid}\left(\text{FC}_{id}(L_{share})\right), \end{aligned} \tag{3}$$

where D_{real}, D_{id}, and D_{cat} denote the discriminator tasks of real-fake, identification, and categorization, respectively. Correspondingly, FC_{id} and FC_{cat} are the fully connection layer (i.e., output layer). L_{share} indicates the layers from the shared model except the output layer.

The generator G accepts category and identity labels, as well as the noise (z), to conditionally yield user specified samples. The objectives are expressed in Eq. 4.

$$\begin{aligned} \mathcal{L}_{real} &= \mathbb{E}_{x\sim p_{real}}[\log D_{real}(x)] \\ &\quad + \mathbb{E}_{x\sim p_{fake}}[\log(1 - D_{real}(x))] \\ \mathcal{L}_{cat} &= \mathbb{E}_{x,y\sim p_{fake}}[\text{CE}(D_{cat}(x), y)] \\ \mathcal{L}_{id} &= \mathbb{E}_{x\sim p_{victim}}[\log D_{id}(x)] \\ &\quad + \mathbb{E}_{x\sim p_{other}}[\log(1 - D_{id}(x))], \end{aligned} \tag{4}$$

where $x \sim p_{victim}$ and $x \sim p_{other}$ denote the data sampled from the victim and the other clients, respectively. However, the samples from the victim or the other clients are inaccessible for the malicious server. Therefore, we estimate the representatives of each client based on their updates to the server. More details will be discussed later in section V-E. Note that $x \sim p_{real}$ indicates samples from an auxiliary dataset, instead of from the clients. Since the federated learning always requires a testing set to evaluate the learning process, and the testing set tends to be consistent to the global distribution of all the clients, we could adopt such testing set as real data to implement the real-fake task. The $\text{CE}(\cdot)$ denotes the cross entropy. Updating D will minimize $\mathcal{L}_{real} + \mathcal{L}_{id}$, and updating G will minimize $\mathcal{L}_{cat} - \mathcal{L}_{real} + \mathcal{L}_{id}$. The training process of mGAN-AI will not affect the federated learning, thus referred to as passive attack whose training scheme will be detailed in the next section.

C. Passive Attack

In the passive attack, the malicious server is assumed to be honest-but-curious, meaning that it only analyzes the updates from the clients, rather than modifying the shared model or introducing adversarial influence. The training scheme is briefed in Algorithm 1. The shared model M, discriminator D, and generator G are initialized randomly. The inputs to mGAN-AI are the shared model M_t and updates u_t^k $(k = 1, 2, \cdots, N)$ from the kth client at each iteration t. Besides, an auxiliary set X_{aux} is prepared for the training, and a victim indexed by v is specified.

At the tth iteration, the current shared model M_t is sent to the clients, then the clients send the updates back to the server. Following the rule of federated learning, those updates are averaged and added to M_t, obtaining the shared model for the next iteration. The discriminator D is initialized by $M_t + u_t^v$ at each iteration to keep updated to the latest performance of categorization and to bias D towards the victim, assisting the attack targeting the victim. As aforementioned that the identification task requires the samples/representatives from each client. The data representatives are calculated from those updates of the clients (for more details, please refer to section V-E). Finally, D and G are updated sequentially based on the objectives in Eq. 4, where the auxiliary set X_{aux} is considered as the real data, and the generated samples X_{fake} by G are the fake data. The two datasets are fed to $\mathcal{L}_{real}$, achieving the real-fake task. For the identification task (i.e., $\mathcal{L}_{id}$), the representatives from u_t^v is treated as samples from the victim, and those from u_t^k $(k \neq v)$ are from others.

The update of D and G can be written as Eq. 5.

$$\begin{aligned} D &= D - \eta_1 \nabla_{\theta_D}(\mathcal{L}_{real} + \mathcal{L}_{id}) \\ G &= G - \eta_2 \nabla_{\theta_G}(\mathcal{L}_{cat} - \mathcal{L}_{real} + \mathcal{L}_{id}), \end{aligned} \tag{5}$$

where η_1 and η_2 are learning rates. θ_D and θ_G indicate the parameters of D and G, respectively.

To achieve better performance, a balanced training on D is required, trained on balanced real and fake data. Since the parameters of D are overwritten by the shared model which is trained more on the real data as the time of iteration increases, we propose to increase the training epoch of D as t increases, thus to compensate the lack of fake data during the training. The training will repeat until meeting the stop criteria: 1) the federated learning reaches the maximum iteration, or 2) the malicious server achieves desired results, i.e., the classification accuracy of the generated samples from G converges above certain threshold as tested on the shared model.

D. Active Attack

The passive attack is invisible but needs to analyze all updates from the clients. A more efficient and powerful way of attacking a specified client would be to isolate the victim from the others, i.e., considering there contains an affiliated server in the malicious server, which only connects to the victim, and the shared model between them is M^{iso} instead of M. Thus, M^{iso} would not be influenced by the other clients and the server performs mGAN-AI on the victim alone. Initially, $M_0^{iso} = M_0 = D$, where M^{iso} and D share

Algorithm 1: Passive Attack of mGAN-AI

Input: The shared model M_t, updates U_t of clients, auxiliary set X_{aux}, and target client v.
Output: The generator G
Initialize M_0, G, and D
for $t = 0$ *to* T **do**
  Send M_t to the clients, and receive updates $U_t = \{u_t^k\}$ from the clients,
  Update the shared model using the updates, $M_{t+1} = M_t + \frac{1}{N}\sum_{k=1}^{N} u_t^k$;
  Initialize D by the shared model and update u_t^v, $D \leftarrow M_t + u_t^v$
  for *client k = 1 to N* **do**
    Calculate data representatives X_k from u_t^k
    if $k == v$ **then**
      label X_k as victim
    end
    else
      label X_k as others
    end
  end
  Get fake samples X_{fake} from G
  Update D by minimizing $\mathcal{L}_{real} + \mathcal{L}_{id}$, feeding X_{aux}, X_{fake}, and X_k $(k = 1, 2, \cdots, N)$
  Update G by minimizing $\mathcal{L}_{cat} - \mathcal{L}_{real} + \mathcal{L}_{id}$
end

the weights. Training mGAN-AI on the affiliated server by following the similar training scheme to the passive attack, the generator would yield samples with higher quality and more distinguishable identity because mGAN-AI is trained purely on the target real data. Note that the identification task (i.e., $\mathcal{L}_{id}$), as well as the calculation of representatives of clients, will be removed during the training because the victim is the only client w.r.t. the affiliated server.

Since the malicious server will "actively" send M^{iso} to the victim to obtain specific information, we call it active attack. Although the active attack violates the original rule of federated learning, it does not introduce negative effect to the federated learning. Meanwhile, the victim might not perceive the attack since the classification task on D is trained purely on the victim's data, which means that its performance would not be significantly different from the locally-trained model using the victim's data only.

E. Calculation of Client Representatives

In the identification task, data from the victim and other clients are required to train the discriminator, while the sever cannot directly access those data. Aono *et al.* [4] recovered the client data based on the update from the client, which is accessible for a malicious server in the federated learning. However, it is only suitable for a simple setting, where the shared model has to be a fully connected network, and the update is required to be obtained by training on a single sample. Obviously, these limitations significantly impede the adaptation of [4] to those widely adopted learning methods, e.g., convolution neural network, batch learning, etc. Inspired by [4], we estimate the representative data of each client from its update sent to the server. The representatives of a client are defined to be synthesized samples that achieve the same update as the real samples after trained on the shared model.

As described in section V-C, u_t^k is the update to the shared model M_t after training the data of the kth client on M_t. Specifically, u_t^k is obtained through backpropagation by minimizing the classification loss $\mathcal{L}$ on the shared model. By the same token, the corresponding representatives X_k will be fed to M_t and calculate the update $u_t^{X_k}$ through backpropagation by minimizing the same loss $\mathcal{L}$. Ideally, $u_t^k = u_t^{X_k}$. We adopt an optimization-based method to calculate X_k based on the shared model, then the objective can be expressed as

$$\arg\min_{X_k} \|u_t^k - u_t^{X_k}\|_2,\ u_t^{X_k} = -\gamma \frac{\partial \mathcal{L}(X_k; \theta_{M_t})}{\partial \theta_{M_t}}, \tag{6}$$

where θ_{M_t} denotes the parameters of the share model M_t, and γ is a scaling factor for balancing the magnitude of $u_t^{X_k}$. The initial representatives are drawn from random noise.

Since optimization-based methods tend to introduce noise or artifacts, Eq. 6 is further regularized by the total variation (TV) [34] as expressed in the following.

$$\mathcal{L}_{TV}(X_k) = \sum_{x \in X_k} \sum_{i,j} \left((x_{i,j+1} - x_{ij})^2 + (x_{i+1,j} - x_{ij})^2 \right)^{\beta}, \tag{7}$$

where X_k is a set of images, and i, j are row and column indices of the image x. The $\mathcal{L}_{TV}$ computes the neighborhood distance to encourage spatial smoothness of an image.

Finally, the objective of achieving valid representatives X_k for the kth client is

$$\arg\min_{X_k} \left\| u_t^k - u_t^{X_k} \right\|_2 + \lambda \mathcal{L}_{TV}(X_k), \tag{8}$$

where λ balances the effect of TV. We calculate X_k by using the box-constrained L-BFGS. Empirically, valid X_t could be achieved after several updates. The experiments in section VII validate the effectiveness of the representatives in the identification task.

VI. Approach Under Anonymous Environment

In this section, we discuss the effects of potential anonymization strategy for mitigating mGAN-AI, and propose a beforehand linkability attack for associating the updates from the corresponding client, ensuring the reconstruction attack under anonymous environment. In the following, we first give the mathematical formulation, i.e., a verification task, and the overview of the attack mechanism. Then the details of the similarity model used for verification are presented, including the model structure, training mechanism, and how to obtain training samples.

A. Problem Formulation and Overview

In the proposed mGAN-AI, the malicious server requires the periodic updates labeled by client identities for implementing the client-level attack. However, the source of the

updates is actually not needed for aggregation in federated learning. Thus, when the clients transmit the model updates without identities over an anonymity network, such as Tor, the malicious server may not be able to perform mGAN-AI. To strengthen mGAN-AI to survive when anonymization strategies are adopted by clients, we propose a beforehand linkability attack, which associates the periodic model updates from the corresponding client.

Formally, we formulate the linkability attack as a verification task:

$$(u_{t_1}^i, u_{t_2}^j) \rightarrow i \stackrel{?}{=} j \quad t_1 \neq t_2, \tag{9}$$

where the malicious server aims to determine whether the two given set of parameter updates, at different training iterations, belong to the same client or not. Note that, instead of de-anonymizing the clients for personally identifying information such as true identities or IP addresses, the proposed linkability attack re-identifies the anonymized model updates by associating periodic updates from different clients.

Intuitively, there exist client-specific patterns in the periodic parameter updates, reflecting the fixed & biased data distribution of the client. Based on the assumption, it is possible to perform a client linkability inference attack [35], and the sensitive properties inference attack [7] in federated learning, by directly taking the periodic updates as the discriminative patterns. However, the parameter updates are calculated by feeding client data into the last-iteration shared model, which is changing over federated training. That is, the updates suffer from unpredictable external interference, which weakens its discriminative ability. We instead take the proposed data representative as an alternative to the pattern, because it is computed by simultaneously considering the updates and the shared model, ensuring its underlying discriminative ability.

The linkability attack is briefly introduced in Algorithm 2. The malicious server collects the shared model M_t and the anonymized updates U_t^* at each iteration t. Firstly, the server measures the similarity of the updates from adjacent iterations. Then the verification results can be obtained by sorting the similarity without explicitly setting a threshold, since all the collected updates are coming from a fixed group of clients. Formally, the verification problem in Eq. 9 can be transformed as follows:

$$\arg \max_{i \in [1,N]} \mathrm{sim}(X_{t_1}^*, X_{t_2}^i) \quad t_1 \neq t_2, \tag{10}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes the similarity function (a siamese verification network is proposed and the details will be presented later). $X_{t_1}^*$ and $X_{t_2}^*$ denote anonymized representatives at different iterations. To get a more precise matching, we develop a simple voting mechanism. Given a client representatives X_t^*, the multi-iteration de-anonymized representatives are utilized for similarity calculation, and the final identity is obtained by the majority, i.e., the identity with the largest count.

Algorithm 2: Linkability Attack With Voting Mechanism

Input: The shared model M_t, the anonymized updates U_t^* of clients, the voting number K.
Output: The estimated identity for U_t^*
Build a list of queues Q storing clients' representatives,
Calculate the representatives X_1 from U_1^* and label them with client identities (random sequence of $[1, N]$),
Initialize Q: enqueue the X_1 for K times,
for t = *2 to T* **do**
  for *each updates* u_t^* **do**
    Calculate client representatives X_t^* from u_t^*,
    Obtain votes for identity j by sorting similarity $n_j = |\arg \max_{i \in [1,N]} \mathrm{sim}(X_t^*, Q_i[k]) = j, \forall k \in [1, K]|, j \in [1, N]$
    get the estimated identity $id = \arg \max_{j \in [1,N]} n_j$,
    Updates the queue Q_{id}
      $Q_{id}.dequeue()$
      $Q_{id}.enqueue(X_t^*)$
  end
end

B. Representatives Similarity Measurement Design

The key of solving the verification problem is a proper similarity measurement of client representatives. Thus, we design a convolutional siamese network combining a verification model with a classification model, which helps to learn a more discriminative CNN (i.e., convolutional neural network) embedding. The model takes a pair of client representatives as input and outputs the similarity in [0,1]. The structure and training details are presented in the following.

1) Network Structure: Fig. 4 briefly illustrates the structure of the proposed convolutional siamese network. It fuses the identification and verification models together, which simultaneously predicts the identities and similarity measurement. The motivation of designing this joint model is to combine the strengths of the two models and learn a more discriminative embedding of client representative. Given a pair of client representatives, the convolutional layer extracts their feature embeddings first. Then we utilize an average pooling layer for down-sampling the feature representation, which improves the generalization of model and speeds up the training. Finally, the identification layer activated by softmax outputs a probability vector corresponding to each client identity. In the meantime, a square layer computes the difference of features and the following sigmoid layer outputs their similarity.

In detail, the verification model takes a pair of client representatives $(\mathrm{Rep}_1, \mathrm{Rep}_2)$ as input and predicts $(\mathrm{Rep}_1, \mathrm{Rep}_2) \rightarrow 0/1$, representing whether the pair of representatives belong to the same client or not. The output is 1 if the inputs depict the same client, 0 otherwise. The verification model forces the representatives of the same client mapped to vicinal points in the feature space. On the other hand, the points are pushed far apart if the inputs are of different clients. However, there exists a major problem in the verification model that it does not consider the relationship between the input pairs and other inputs during training, but only the weak "same or different" label. That might fail to produce a discriminative embedding. As for the identification model, it learns the nonlinear mapping from an input representative to

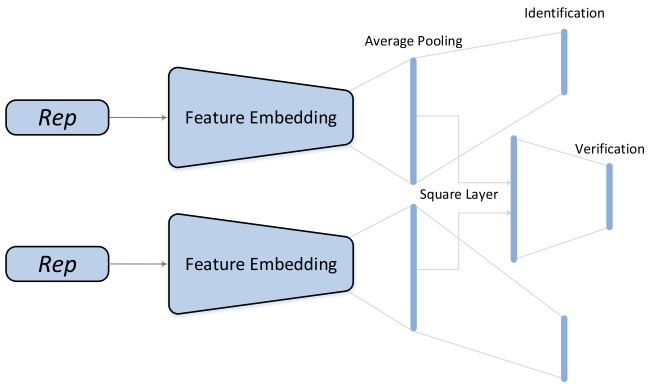


Fig. 4. Structure of the convolutional siamese network which fuses the identification and verification models together.

the client identity. Generally, a softmax function is used after the final fully connected layer of a identification model, which outputs the probability vector corresponding to each identity label. The cross-entropy function (CE) is commonly used as the loss. The major drawback of utilizing the identification model for a matching task is that, the CE loss does not account for the similarity between input pairs during testing. Thus, a joint model fusing a verification model and an identification model can combine the strengths of them and learn a more discriminative embedding.

It is worth noting that verification-identification models have been applied in some fields such as face recognition [36], and image retrieval [37]. There are some differences between our method and theirs. First, we use the CE training loss rather than the contrastive loss used in [36]. Because the contrastive loss would result in overfitting when training dataset is small. So we use the CE loss instead, similar to [37]. Second, existing works train the model from scratch or from a public pre-trained model on ImageNet. In this work, we instead use a pre-trained shared model M_t, which is more suitable for our data domain.

After the similarity model gets well trained, we perform a different testing procedure from learning. In detail, since there exist differences between the training domain and the testing domain, directly applying the verification function of the attack model for similarity measurement may suffer from performance degradation. Thus, we take the output of the average pooling layer as the discriminative descriptor for client representative. Then the similarity is computed by the Euclidean distance between their L_2 normalized CNN embeddings. That is the implementation of funciton $\mathrm{sim}(\cdot,\cdot)$ in Eq. 10.

2) Training Loss: For training the joint model, there include two optimization objectives, the identification loss and the verification loss. They are supervised by the identification label and the verification label.

a) Identification loss: We take two pre-trained shared models in the joint architecture. Then we fine-tune the models on a new dataset. The two identification models share weights during the training. Similar to conventional multi-class classification methods, we use the CE loss for identity prediction. Formally, the loss function is expressed by

$$\mathcal{L}_{id}* = \mathbb{E}_{\mathrm{rep},id}[\mathrm{CE}(M_{id}(\mathrm{rep}), id)] \tag{11}$$

where M_{id} denotes the identification model, and (rep, id) for the training samples.

b) Verification loss: Our siamese network directly takes the high-level feature embedding as the client representative descriptor d_1, d_2. They are compared for similarity measurement. Generally, the high-level feature from the fine-tuned CNN owns a discriminative ability. It is more compact than the activation outputs in the intermediate layers. To fuse the feature, we utilize a non-parametric square layer which takes the two descriptor d_1, d_2 as inputs, and outputs the element-wise L_2 distance, expressed by $d = (d_1 - d_2)^2$. Finally, the following sigmoid function to the 1-dim fully connected layer outputs a value in $[0, 1]$, representing the probability of the two input representatives belonging to the same client. Similar to conventional binary classification methods, we use the CE loss for identity prediction, formulated by

$$\begin{aligned}\mathcal{L}_{ver}* = \mathbb{E}_{(\mathrm{Rep}_1,\mathrm{Rep}_2)}[s \log M_{ver}(\mathrm{Rep}_1, \mathrm{Rep}_2) \\ +(1-s)\log(1 - M_{ver}(\mathrm{Rep}_1, \mathrm{Rep}_2))],\end{aligned} \tag{12}$$

where $(\mathrm{Rep}_1, \mathrm{Rep}_2)$ denotes the training samples, i.e., pair of client representatives. And s is the matching label. $s = 1$ the representatives pair depicts the same client; otherwise, $s = 0$. $M_{ver}(,)$ denotes the verification model.

3) Training Samples - Shadow Federated Learning: Training the proposed similarity model requires labeled client representatives, i.e, the identity information, which the server has no directly access to under anonymous environment. To solve the problem, we develop a shadow federated learning method for generating training samples. The key idea is to implement a "shadow" federated learning that imitates the behaviour of the original one, where we know the periodic parameter updates together with their sources. Then the proxy federated learning produces samples for the supervised training of the attack model. The malicious server owns an Auxiliary dataset X_{aux} for evaluating shared model M_t, which can be used for the shadow training. The same domain between X_{aux} and the real dataset from clients ensures the effectiveness of the attack model during testing.

Formally, the attacker first splits the auxiliary set X_{aux} into N parts. Each part is treated as the training dataset of one shadow client, which are labeled with client identities (random sequence of $[1, N]$). Then the attacker sets a same objective and initiate a shared model as the original federated learning, both of which he has full access to. Thus, a shadow federated learning can be re-performed among these shadow'clients. At each iteration of training, shadow clients train the shared model individually and upload the updates. Let M'_t be the shared model at the iteration t, and let $u_t^{k'}$ be the parameter updates from the kth client at iteration t. Combining M'_t and $u_t^{k'}$, the client representatives $X_{t_1}^{k'}$ can be calculated. Finally, the training samples $(X_t^{k'}, k)$ for identification loss and the samples $(X_{t_1}^{k'_1}, X_{t_2}^{k'_2}, s)$ for verification models are generated. After the similarity model gets well trained, mGAN-AI could be performed under anonymous environment

TABLE I
NETWORK STRUCTURE FOR MNIST

Classifier/ Discriminator	$28^2 \times 1 \xrightarrow{\text{Conv (stride = 2)}} 14^2 \times 32 \xrightarrow{\text{Conv (stride = 1)}}$ $14^2 \times 64 \xrightarrow{\text{Conv (stride = 2)}} 7^2 \times 128 \xrightarrow{\text{Conv (stride = 1)}}$ $7^2 \times 256 \xrightarrow{\text{FC}} 12,544 \xrightarrow{\text{FC, Softmax}} (1, 10, 1)$
Generator	$(100, 1, 1) \xrightarrow{\text{Embedding}} 100 \xrightarrow{\text{FC}} 3^2 \times 384 \xrightarrow{\text{Deconv}}$ $7^2 \times 192 \xrightarrow{\text{Deconv}} 14^2 \times 96 \xrightarrow{\text{Deconv, tanh}} 28^2 \times 1$

TABLE II
NETWORK STRUCTURE FOR AT&T

Classifier/ Discriminator	$64^2 \times 1 \xrightarrow{\text{Conv (stride = 2)}} 32^2 \times 32 \xrightarrow{\text{Conv (stride = 1)}}$ $32^2 \times 64 \xrightarrow{\text{Conv (stride = 2)}} 16^2 \times 128 \xrightarrow{\text{Conv (stride = 2)}}$ $8^2 \times 256 \xrightarrow{\text{FC}} 16,384 \xrightarrow{\text{FC, Softmax}} (1, 40, 1)$
Generator	$(100, 1, 1) \xrightarrow{\text{Embedding}} 100 \xrightarrow{\text{FC}} 4^4 \times 512 \xrightarrow{\text{Deconv}}$ $8^2 \times 256 \xrightarrow{\text{Deconv}} 16^2 \times 128 \xrightarrow{\text{Deconv}}$ $32^2 \times 256 \xrightarrow{\text{Deconv}} 64^2 \times 256 \xrightarrow{\text{Deconv, tanh}} 64^2 \times 1$

$m^2 \times n$ denotes the size of a layer, i.e., $m \times m$ map with n channels.

by associating the representatives of same clients. The experiments in section VII validate the effectiveness of the shadow federated training in the linkability attack.

VII. EXPERIMENTAL EVALUATION

In this section, we first clarify the dataset and experiment setup in Sec. VII-A. Then, Sec. VII-B validates the effectiveness of the proposed linkability attack. The matching precision is above 99% for both experiments on two datasets, which ensures the mGAN-AI to survive the anonymous environment. Then, the performance of the proposed mGAN-AI is presented in Sec. VII-C. Finally, Sec. VII-E conducts qualitative and quantitative comparison between mGAN-AI and another two typical attacks, i.e., the model inversion (MI) attack and GAN-based attack.

A. Datasets and Experiment Setup

1) MNIST: The MNIST dataset [38] contains 70,000 handwritten digits images from 0 to 9 (i.e., 10 classes). The images are in gray scale with the size of 28×28, and divided into the training set (60,000 samples) and testing set (10,000 samples).

2) AT&T: The AT&T dataset [39] consists of facial images from 40 different persons, namely 10 images per person. The images are in gray scale with the size of 92×112. They were taken at different times and with large variation in facial expression (smiling or w/o smiling) and facial details (glasses or w/o glasses). We resize the images into 64×64.

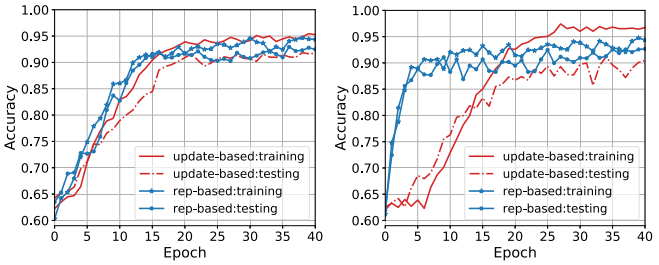
3) Experiment Setup: As discussed in Sec. V, the proposed mGAN-AI involves three components: 1) classifier (i.e., the shared model), 2) discriminator, and 3) generator. The classifier and discriminator are constructed by convolutional neural networks, and they share the structure except the output layer. The generator is constructed with deconvolution layers. Tables I and II show the network structures for MNIST and AT&T. For the generator, the kernel size is 5 with the stride of 2. The input to the generator is formated as concatenation of random noise, categorical label, and identity label (Fig. 3b), whose length are 100, 1, and 1, respectively. The embedding layer squeezes the length of input to 100. For the classifier and discriminator, the kernel size is 3×3. Their output has the similar format as the input of the generator, i.e., the first digits for reality, the next C for category, and the last for identity. The C indicates the number of classes for different datasets, i.e., $C = 10$ for MNIST and $C = 40$ for AT&T. The activation functions ReLU and LReLU are adopted in the generator and discriminator, respectively. In addition, batch normalization is used at the intermediate layers of the generator.

In the training of the shared model, the SGD (i.e., stochastic gradient descent) optimizer is adopted with the learning rate of 0.002. For discriminator and generator (i.e., multi-task GAN), the Adam optimizer is used with the learning rate and momentum term to be 0.0002 and 0.5, respectively. In calculating the representatives, the box-constrained L-BFGS is employed, where $\lambda = 0.00015$, $\beta = 1.25$. Most of the above hyperparameters are set based on the implementation of related studies, with further manually modification for better performance. Besides, for all experiments, we simulate the distributed learning manner in a single machine with an Intel i7-6850K CPU, 64 GB RAM and two NVIDIA GTX 1080 TI GPU, in Keras [40] implementation.

B. Performance of Linkability Attack

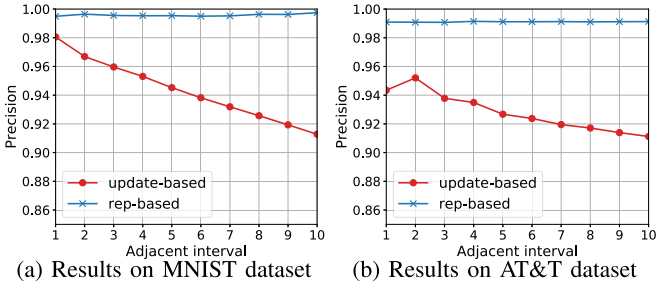
Under the anonymous environment, we consider the server has no knowledge about the sources of the periodic parameter updates. Thus, to ensure the effectiveness of mGAN-AI in this setting, the server performs a beforehand linkability attack. We use two evaluation metrics: 1) Matching Accuracy, 2) Precision. The former describes the accuracy when feeding random-selected pairs that might come non-adjacent iteration, while the latter describes the rank accuracy when feeding a Rep and a matching gallery of Rep_k of different clients from adjacent iterations.

For implementing the shadow federated learning, we split the Auxiliary dataset X_{aux} into N shadow clients. The number of shadow client is set to $N = 10$. Following the settings of biased data distribution on clients in standard federated learning, each shadow client owns equal amount of samples from different classes. For training the attack model, we take the pre-trained shared model weights as the initial to the embedding layer. In the experiments, the shared model achieves convergence around iteration 150, so the $150th$ shared model is used. The optimization objective consists of identification loss and verification loss. We set equal weight for them. During training, we sample a positive/negative pair from the same/different client identities. The SGD optimizer is adopted with the learning rate of 0.001 with momentum $= 0.9$. When performing the majority voting with similarity, the voting number K is set 3. We compare our proposed linkability attack to those utilizing the raw parameter updates as the client-specific patterns, such as [7], [35]. In these works, the parameters of the fully connected (FC) layer, followed by an averaging pooling operation, are used. For clarity, our



(a) Results on MNIST dataset. (b) Results on AT&T dataset.

Fig. 5. Performance comparison on the training/testing matching accuracy curves across training epoches, for our rep-based attack and update-based attack.



(a) Results on MNIST dataset (b) Results on AT&T dataset

Fig. 6. Effect comparison of the adjacent interval adj on the linkability attack, for rep-based attack and update-based attack.

proposed attack and theirs are named as rep-based attack and update-based attack, respectively.

Fig. 5 shows the training/testing matching accuracy curves of verification task across training epoches, for our rep-based attack and update-based attack. Note that, the testing accuracy is computed by feeding random-selected pairs obtained from the original federated learning. In other words, the training samples and testing are disjoint even for their classes. For both evaluation in MNIST and AT&T datasets, the rep-based attacks achieve a high training/testing matching accuracy, which demonstrates the advance of the representatives in discriminative ability. Especially on AT&T dataset, rep-based attack converges faster and achieves higher testing accuracy, compared to update-based attack. While the testing accuracy is reduced obviously for update-based attack, and rep-based attack remains relatively stable. It shows better generalization of rep-based attack for unseen samples. However, the advantage of the rep-based attack is not very obvious on MNIST dataset. As shown in Fig. 5a, the testing accuracy 92.5% of rep-based attack is slightly higher than 91.7% of update-based attack. We attribute the difference in results from the two datasets to the variance of data distribution. The smaller variance on MNIST dataset results in less influence from the shared model to clients' local models. The discriminative ability of periodic updates would not be weakened much by the external interference, making the update-based attack achieve comparable results with rep-based attack. Thus, considering the non-IID data in federated learning, the rep-based attack imposes a stronger threat compared to update-based attack.

To analyze the effect of the adjacent interval adj to the linkability attack, we perform the precision evaluation of matching the representatives from fixed-interval iterations. The results are shown in Fig. 6. Intuitively, the difference of the periodic updates would increase during federated learning. The precision of update-based attack drops as adj increases. On the contrary, the precision remains high for our rep-based attack. It demonstrated that the representatives own a more robust discriminative ability, compared to the raw parameter updates.

The quantitative comparison results are shown in Table III. The precision results are generally higher than matching accuracy, since the matching results are the one with highest similarity in adjacent iterations. The updates are of less difference. Our rep-based attack succeeds with over 99% precision

TABLE III

QUANTITATIVE COMPARISON ON THE TRAINING/TESTING MATCHING ACCURACY AND PRECISION, WHERE REP FOR OUR REP-BASED ATTACK AND UPDATE FOR UPDATE-BASED ATTACK

		MNIST		AT&T	
		training	testing	training	testing
Rep	Accuracy	0.9438	0.9247	0.9435	0.9267
	Precision	0.9990	0.9950	0.9980	0.9909
Update	Accuracy	0.9528	0.9173	0.9672	0.9047
	Precision	0.9990	0.9805	0.9727	0.9434

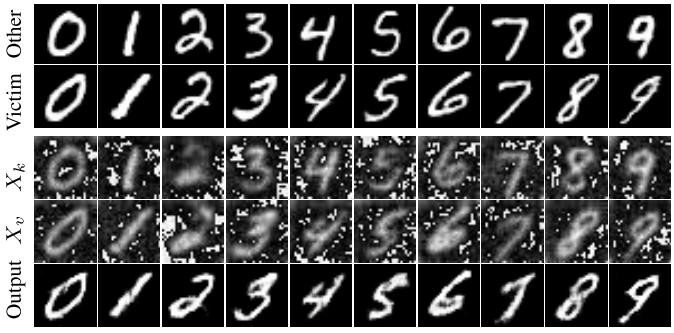


Fig. 7. The results of mGAN-AI on the MNIST dataset. The first two rows are the real samples from the other clients and the victim. The next two rows are corresponding representatives calculated from other (X_k, $k \neq v$) and victim (X_v). The last row is the reconstructed samples, which are similar to victim's, presenting larger rotation as compared to those of the other clients.

on the MNIST and AT&T datasets. It demonstrated that when the clients upload anonymized updates, the malicious server can associate those from same client, ensuring the mGAN-AI to survive the anonymous environment, even if the attacker does not know the actual sources. Compared to the update-based attack, our attack outperforms it on both matching accuracy and precision during testing. So in the following, we focus on the evaluation of mGAN-AI under non-anonymous environment.

C. Client-Level Privacy Attack by mGAN-AI

This section evaluates the effectiveness of passive mGAN-AI by comparing the reconstructed samples with the real samples of the victim. Instead of evenly splitting the training data to the clients, we simulate biased data distribution on clients according to the setting of federated learning.

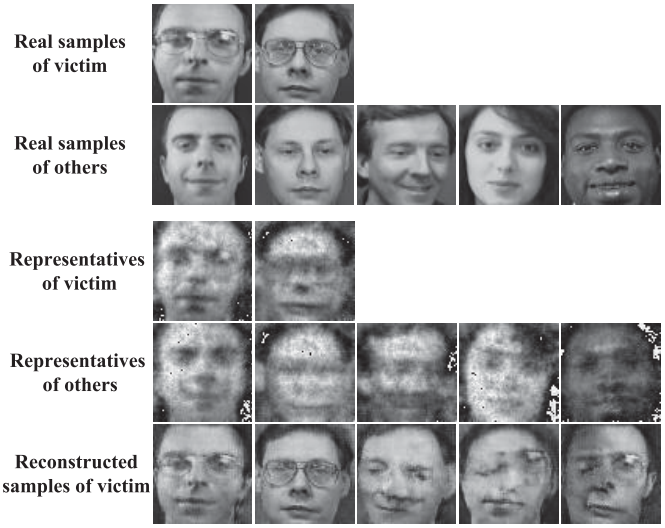
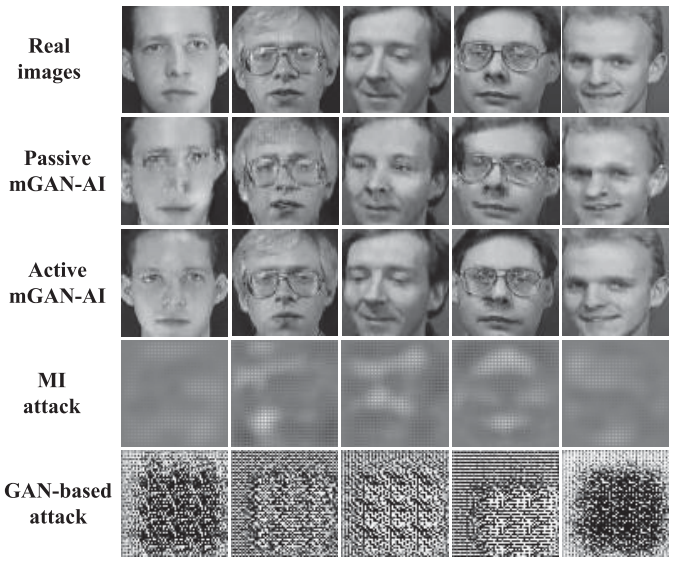
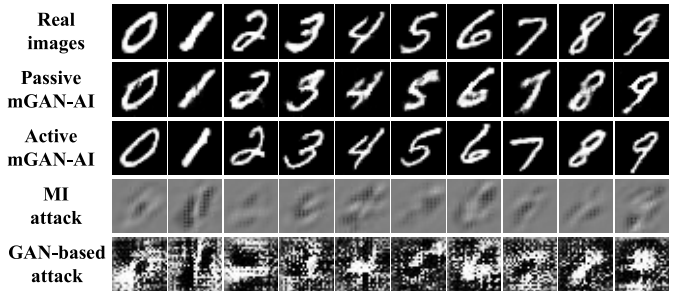


Fig. 8. The results of mGAN-AI on the AT&T dataset. The top two rows are real samples from the victim and other clients, respectively. The corresponding representatives are shown in the next two rows. The last row is the reconstructed samples, which show that mGAN-AI would specifically recover client-dependent property, i.e., wearing glasses.



(a) On AT&T dataset



(b) On MNIST dataset

Fig. 9. Comparison of reconstructed samples by mGAN-AI, MI attack, and GAN-based attack. The top row is the real samples, and the rest are the reconstructed samples by the passive mGAN-AI, active mGAN-AI, MI attack, and GAN-based attack, respectively.

In addition, opposite to [5] that assumed each client consists of a single class, we achieve more flexible condition that each client could own samples from multiple classes. Note that different clients may share the samples from the same class.

1) Evaluation on the MNIST Dataset: We set the number of client $N = 10$. Each client randomly draws 100 samples from three random classes as its private data. To gain distinguishable properties for the victim, we rotate the samples of the victim by InfoGAN [33], which could synthesize rotated version of the digits. Ideally, mGAN-AI should reconstruct rotated digits owned by the victim. Fig. 7 shows the reconstructed client-level samples by mGAN-AI in the federated learning. The first two rows are the samples from other clients and the victim, respectively. The 3rd and 4th rows are the corresponding representatives (X_k in Algorithm 1) calculated from other clients and the victim. The last row shows the reconstructed samples, which are similar to the second row (victim) and different from the first row (other) in rotation obviously. Note that ten classes are presented in Fig. 7 for the victim because the experiment is repeated, so that the victim could walk through all possible classes. This demonstrates that mGAN-AI can successfully and precisely recover the provide data of the victim, i.e., compromising the client-level privacy.

2) Evaluation on the AT&T Dataset: The similar experiment is performed on the AT&T dataset, where the number of clients $N = 10$. The distinguishable property of the victim is assigned to be wearing glasses. As shown in Fig. 8, the first row shows the private data of the victim, i.e., two faces wearing glasses. The second row are faces from other clients that do not wear glasses. Note that the first two faces in the second row are the same person as in the first row. Therefore, successfully attack to the victim should be reconstruction of the first two person wearing glasses. The 3rd and 4th rows are corresponding representatives. The last row are reconstructed samples, where the first two samples are identical to the victim while the rest are significantly distorted. This demonstrates that mGAN-AI could only clearly reconstruct the private data of the victim, i.e., mGAN-AI specifically attacks the victim.

Note that mGAN-AI does not violate the differential privacy (DP), which aims to prevent the recovery of specific samples used during the training. In other words, DP tries to make the adversary fail to tell whether a given sample belongs to the training set. However, without inferring the membership of a given sample, it can still generate samples distinguishable from real samples as demonstrated in Figs. 7 and 8, which obviously leads to severe privacy violation. Besides, convergence of the shared model requires a relatively loose privacy budget when applying DP in the federated learning [7], which means the magnitude of the perturbation should be carefully controlled to ensure high-utility updates used for training mGAN-AI.

D. Quantitative and Qualitative Comparison

This section compares mGAN-AI with two state-of-the-art attacking models, i.e., inversion attack (MI) and GAN-based attack, in aspects of reconstruction quality of the victim and side effect to the federated learning. Note that, although MI was not proposed for attacking federated learning, it can be

TABLE IV
QUANTITATIVE COMPARISON ON INCEPTION SCORE

	Inception Score
Real Images	1.55 ± 0.04
Passive mGAN-AI	1.42 ± 0.02
Active mGAN-AI	1.61 ± 0.05
GAN-based attack	1.18 ± 0.03
MI attack	1.01 ± 0.03

performed in the scenario, i.e., the adversary has the white-box access to the shared model. For a fair comparison, the same experiment setup is followed in the federated learning on both AT&T and MNIST datasets. The results are shown in Fig. 9. For both datasets, the first row are real samples from the victim. The second and third rows are the generated samples from passive attack and active attack of mGAN-AI, respectively. The last two rows are reconstructions by MI attack and GAN-based attack, respectively.

Comparing the results from the proposed mGAN-AI with MI and GAN-based attack, mGAN-AI generates samples with much higher quality and more identical to the real samples. The results of MI attack is significantly blurry, consistent with the results in [5], [23]. The failure is mainly caused by simply maximizing the model output w.r.t. a target label, which would lead to uninterpretable samples when dealing with complicated neural networks. The results of GAN-based attack do not converge at all because it heavily relies on the introduced "adversarial influence" which would be averaged (become trivial) before adding to the shared model. The results of GAN-based attack are consistent with those in its original work, where unrecognizable images were generated without the adversarial influence. Comparing the passive and active attack of mGAN-AI, the active attack performs better but it would slightly degrade the learned model in the federated learning. In contrast, the passive learning does not impact the learning model, which will be demonstrated later.

To quantitatively evaluate the generated images, the inception score [41] is used to statistically compare the three attack models. Inception score has been widely adopted in image quality evaluation, especially in the area of image synthesis. We generate 400 samples from each of the three attacks and compute the inception score for each method. The results are shown in Table IV, where the proposed mGAN-AI shows higher score than MI and GAN-based attack, demonstrating that mGAN-AI generates samples with higher quality.

Finally, we investigate the side effect of the three attacking models to the federated learning. Fig. 10 shows the accuracy of the shared model at each iteration in the federated learning when performing different attacks, i.e, passive mGAN-AI, active mGAN-AI, and GAN-based attack. The GAN-based attack significantly reduces the accuracy and presents drastic oscillation (non convergence) because it introduces adversarial influence. The passive mGAN-AI achieves the highest accuracy because it does not affect the training process of federated learning. The active mGAN-AI gets a bit lower but stable accuracy. With this cost, however, it achieves better reconstruction quality as shown in Fig. 9.

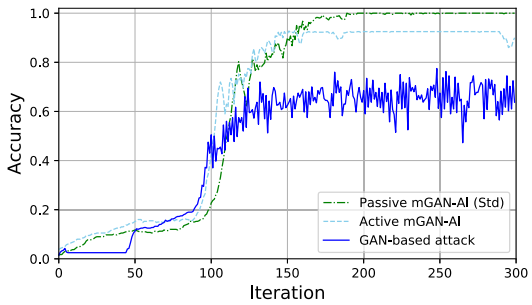


Fig. 10. The accuracy of the shared model in federated learning against different attacks.

E. Discussion of Defenses

This section discusses potential defense strategies to mitigate the reconstruction attacks against federated learning.

1) Encryption-Based Protocol: By utilizing the advance of the encryption techniques in federated learning, e.g., secure aggregation [19], homomorphic encryption [42], the clients can transmit the encrypted parameter updates to the server. Then the server performs the computations on encrypted data. Thus, users' privacy would not be violated. However, there exist two main limitations: 1) encryption-based protocols suffer from computation inefficiency, and 2) it would be difficult to recognize the malicious updates, e.g., the poisoning updates.

2) Trusted Execution Environment (TEE): TEE creates an isolated environment in parallel with OS, guaranteeing the confidentiality and integrity of loaded code and data. In the case of federated learning, the training can be performed inside a TEE on the server. Then the intermediate values, e.g., the parameter updates, cannot be accessed by the malicious server.

3) Dynamic Participation: Rather than participate in the federated learning from the beginning to the end, the users can drop in or drop out during the learning process. Since periodic updates are required for training mGAN-AI, the dynamic participation would violate the attack performance. However, the performance of the shared model might suffer from the frequent drop-in and drop-out, specially under the non-IID assumption of data distribution.

VIII. CONCLUSION

This paper investigated the privacy risk of federated learning, i.e., a privacy-preserving learning framework working in a decentralized manner. Against federated learning, we proposed a generic and practical reconstruction attack named mGAN-AI, which enables a malicious server to not only reconstruct the actual training samples, but also target a specific client and compromise the client-level privacy. The proposed attack does not affect the standard training process, showing obvious advantages over the current attack mechanisms. To step further consider the anonymization strategy for mitigating mGAN-AI, we propose a beforehand linkability attack which re-identifies the anonymized updates by associating the client representatives. The extensive experimental results on two benchmark datasets demonstrate that the

mGAN-AI can reconstruct samples that resemble the victim's training samples, and the linkability attack succeeds with over 99% precision. Both the attacks outperform the state-of-the-art attacking algorithms.

REFERENCES

[1] Z. Wang *et al.*, "When mobile crowdsensing meets privacy," *IEEE Commun. Mag.*, vol. 57, no. 9, pp. 72–78, Sep. 2019.

[2] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2015, pp. 1310–1321.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2016, *arXiv:1602.05629*. [Online]. Available: http://arxiv.org/abs/1602.05629

[4] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Proc. Int. Conf. Appl. Techn. Inf. Secur.* Cham, Switzerland: Springer, 2017, pp. 100–110.

[5] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 603–618.

[6] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[7] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 691–706.

[8] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," 2016, *arXiv:1610.09585*. [Online]. Available: http://arxiv.org/abs/1610.09585

[9] 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. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 2512–2520.

[10] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization:Distributed optimization beyond the datacenter," 2015, *arXiv:1511.03575*. [Online]. Available: http://arxiv.org/abs/1511.03575

[11] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2016, *arXiv:1604.00981*. [Online]. Available: http://arxiv.org/abs/1604.00981

[12] M. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1876–1884.

[13] J. Hamm, Y. Cao, and M. Belkin, "Learning privately from multiparty data," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 555–563.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[15] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Trans. Inf. Forensics Security*, vol. 15, no. 1, pp. 1486–1500, Sep. 2020.

[16] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*. [Online]. Available: http://arxiv.org/abs/1712.07557

[17] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 29–38.

[18] G. Danner and M. Jelasity, "Fully distributed privacy preserving mini-batch gradient descent learning," in *Proc. Int. Conf. Distrib. Appl. Interoperable Syst.*, 2015, pp. 30–44.

[19] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur (CCS)*, Oct. 2017, pp. 1175–1191.

[20] Y. Chen, C. Shen, C. Wang, Q. Xiao, K. Li, and Y. Chen, "Scaling camouflage: Content disguising attack against computer vision applications," *IEEE Trans. Depend. Sec. Comput.*, early access, Feb. 4, 2020, doi: 10.1109/TDSC.2020.2971601.

[21] Z. Wang, M. Song, S. Zheng, Z. Zhang, Y. Song, and Q. Wang, "Invisible adversarial attack against deep neural networks: An adaptive penalization approach," *IEEE Trans. Depend. Sec. Comput.*, early access, Jul. 30, 2019, doi: 10.1109/TDSC.2019.2929047.

[22] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The security of autonomous driving: Threats, defenses, and future directions," *Proc. IEEE*, vol. 108, no. 2, pp. 357–372, Feb. 2020.

[23] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.

[24] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Membership inference attacks against generative models," *Proc. Privacy Enhancing Technol.*, vol. 2019, no. 1, pp. 133–152, Jan. 2019.

[25] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 739–753.

[26] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2015, pp. 1322–1333.

[27] Z. Yang, E.-C. Chang, and Z. Liang, "Adversarial neural network inversion via auxiliary knowledge alignment," 2019, *arXiv:1902.08552*. [Online]. Available: http://arxiv.org/abs/1902.08552

[28] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 148–162.

[29] H. Corrigan-Gibbs, D. I. Wolinsky, and B. Ford, "Proactively accountable anonymous messaging in verdict," in *Proc. USENIX Secur. Symp.*, 2013, pp. 147–162.

[30] C. Xie, S. Koyejo, and I. Gupta, "SLSGD: Secure and efficient distributed on-device machine learning," 2019, *arXiv:1903.06996*. [Online]. Available: http://arxiv.org/abs/1903.06996

[31] C. Fung, J. Koerner, S. Grant, and I. Beschastnikh, "Dancing in the dark: Private multi-party machine learning in an untrusted setting," 2018, *arXiv:1811.09712*. [Online]. Available: http://arxiv.org/abs/1811.09712

[32] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[33] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[34] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5188–5196.

[35] T. Orekondy, S. Joon Oh, Y. Zhang, B. Schiele, and M. Fritz, "Gradient-leaks: Understanding and controlling deanonymization in federated learning," 2018, *arXiv:1805.05838*. [Online]. Available: http://arxiv.org/abs/1805.05838

[36] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[37] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person reidentification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, pp. 13:1–13:20, 2017.

[38] L. Yann, C. Corinna, and J. Christopher. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yhann.lecun.com/exdb/mnist

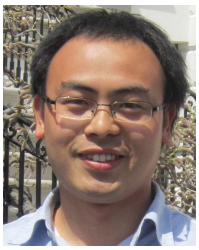
[39] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Dec. 1994, pp. 138–142.

[40] F. Chollet *et al.* (2015). *Keras*. [Online]. Available: https://keras.io

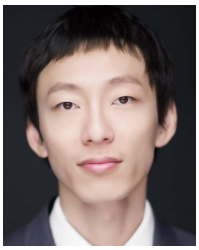
[41] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[42] 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.

Mengkai Song (Graduate Student Member, IEEE) received the B.S. degree in computer science from Wuhan University in 2017, where he is currently pursuing the master's degree with the School of Cyber Science and Engineering. His research interests include privacy and security in deep learning.



Zhibo Wang (Senior Member, IEEE) received the B.E. degree in automation from Zhejiang University, China, in 2007, and the Ph.D. degree in electrical engineering and computer science from the University of Tennessee, Knoxville, in 2014. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University, China. His current research interests include the Internet of Things, network security, and privacy protection. He is a member of ACM.



Zhifei Zhang (Member, IEEE) received the B.S. degree from Northeastern University, Shenyang, China, in 2010, the M.S. degree from Zhejiang University, Hangzhou, China, in 2013, and the Ph.D. degree from the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA, in 2018. He is currently a Research Engineer at Adobe, San Jose, CA, USA. His research interests include signal and image processing, pattern recognition, machine learning, and computer vision.

Yang Song (Member, IEEE) received the B.S. degree in electrical engineering from Northeastern University, Shenyang, China, in 2010, and the M.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2013. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN, USA.

Her research interests include signal and image processing, pattern recognition, and computer vision.

Qian Wang (Senior Member, IEEE) received the B.S. degree from Wuhan University, China, in 2003, the M.S. degree from the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China, in 2006, and the Ph.D. degree from the Illinois Institute of Technology, USA, in 2012, all in electrical engineering. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests include wireless network security and privacy, cloud computing security, and applied cryptography. He is an Expert under the 1000 Young Talents Program of China. He was a co-recipient of the Best Paper Award at the IEEE ICNP 2011.

Ju Ren (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Central South University, China, in 2009, 2012, and 2016, respectively. From 2013 to 2015, he was a visiting Ph.D. student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is currently a Professor with the School of Information Science and Engineering, Central South University, China. His research interests include the Internet of Things, wireless communication, network computing, and cloud computing.

Dr. Ren was a recipient of the Best Paper Award at the IEEE IoP 2018 and the Most Popular Paper Award of the *Chinese Journal of Electronics* from 2015 to 2018. He was a TPC member of many international conferences, including IEEE INFOCOM'19/18, Globecom'17, WCNC'17, WCSP'16, and so on. He also served as a Poster Co-Chair for the IEEE MASS'18, a Track Co-Chair for IEEE VTC'17 Fall and IEEE I-SPAN'18, and an Active Reviewer for over 20 international journals. He currently serves/served as an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and *Peer-to-Peer Networking and Applications*.

Hairong Qi (Fellow, IEEE) received the B.S. and M.S. degrees in computer science from Northern Jiaotong University, Beijing, China, in 1992 and 1995 respectively, and the Ph.D. degree in computer engineering from North Carolina State University, Raleigh, in 1999. She is currently a Gonzalez Family Professor with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville. Her current research interests include advanced imaging and collaborative processing in resource-constrained distributed environment, hyperspectral image analysis, and bioinformatics.