IEEE TRANSACTIONS ON
**INFORMATION FORENSICS
AND SECURITY**

IEEE Xplore® Digital Library
http://ieeexplore.ieee.org/Xplore

GANobfuscator: Mitigating Information Leakage under GAN via Differential Privacy

Chugui Xu, *Student Member, IEEE*, Ju Ren, *Member, IEEE*, Deyu Zhang, *Member, IEEE*, Yaoxue Zhang, *Senior Member, IEEE*, Zhan Qin, *Member, IEEE*, Kui Ren, *Fellow, IEEE.*

***Abstract*—By learning generative models of semantic-rich data distributions from samples, generative adversarial network (GAN) has recently attracted intensive research interests due to its excellent empirical performance as a generative model. The model is used to estimate the underlying distribution of a dataset and randomly generate realistic samples according to their estimated distribution. However, GANs can easily remember training samples due to the high model complexity of deep networks. When GANs are applied to private or sensitive data, the concentration of distribution may divulge some critical information. It consequently requires new technological advances to mitigate the information leakage under GANs. To address this issue, we propose GANobfuscator, a differentially private GAN, which can achieve differential privacy under GANs by adding carefully designed noise to gradients during the learning procedure. With GANobfuscator, analysts are able to generate an unlimited amount of synthetic data for arbitrary analysis tasks without disclosing the privacy of training data. Moreover, we theoretically prove that GANobfuscator can provide strict privacy guarantee with differential privacy. In addition, we develop a gradient pruning strategy for GANobfuscator to improve the scalability and stability of data training. Through extensive experimental evaluation on benchmark datasets, we demonstrate that GANobfuscator can produce high-quality generated data and retain desirable utility under practical privacy budgets.**

***Index Terms*—Information Leakage; Generative Adversarial Network; Deep Learning; Differential Privacy.**

I. Introduction

WITH the continued advances in mobile computing and the surging popularity of social media, a massive amount of semantic-rich data (e.g., image, text, audio, video) about individuals is being collected. Many data mining methodologies have been developed for analyzing those big data sets. One representative example is deep learning, which typically needs a huge amount of training samples to achieve promising performance. However, there exists domains where it is impossible to get as much data as we want. For example, we can never get enough medical data from all patients for privacy and sensitivity reasons. Thus, the problem of building high-quality medical analytics models remains very challenging at present. Fortunately, generative models [1], [2] provide us a promising direction to alleviate the data scarcity issue. By sketching the data distribution from a small set of training data, we are able to sample from the distribution and generate much more samples for our study. Generative Adversarial Network (GAN) [3] and its variants have demonstrated impressive performance in modelling the underlying data distribution by combining the complexity of deep neural networks and game theory, to generate high quality "fake" samples that are hard to be differentiated from real ones [4], [5].

However, GANs are facing the risk of implicitly disclosing privacy information of the training samples. The adversarial training procedure and the high model complexity of deep neural networks, jointly encourage a distribution that is concentrated around training samples. By repeatedly sampling from the distribution, there is a considerable chance of recovering the training samples [6]. While analyzing and understanding such data entails tremendous commercial value (e.g., targeted advertisements and personalized recommendations), individual privacy should be considered in such practice. Generally, privacy protection can be enforced in two settings. In the interactive setting, a trusted data curator collects data from individuals and provides a privacy-preserving interface for the analyst to execute queries over the data. In the more challenging non-interactive setting, the data curator releases a "sanitized" version of the data, simultaneously providing analysis utility for the analyst and privacy protection for the individuals represented in the data [7]. Ideally, with the high quality generative distribution in hand, we can protect the privacy of raw data by releasing only the data distribution instead of the raw data to the public or constrained individuals, and can even sample datasets to fit our needs and conduct further analysis. However, the adversary still can perform further inferences on the high quality generated data. For example, Hitaj etal. [8] introduced an active inference attack model that can reconstruct training samples from the generated ones. Therefore, it is highly demanded to have generative models that not only generates high quality samples but also protects the privacy of the training data.

Differential Privacy (DP) [9], [10] has been recently accepted as a promising way to preserve the data privacy without sacrificing data utility. Following this trend, increasing researchers are focusing on applying DP in deep learning networks [8], [11], [12]. Most of them apply DP by injecting noise on the weights of the neural networks during the data training phase. However, since incorporating DP with GANs in such a way generally causes significant impacts on the stability and scalability of training GAN models, existing solutions are not suitable for mitigating information leakage in GANs

Chugui Xu, Ju Ren, Deyu Zhang and Yaoxue Zhang are with the School of Computer Science and Engineering, Central South University, Changsha, China, 410083. E-mails: {chuguixu, renju, zdy876, zyx}@csu.edu.cn.

Zhan Qin and Kui Ren are with Institute of Cyberspace Research, Zhejiang University, Hangzhou, China, 310058. E-mail: zhan.qin@utsa.edu, kuiren@buffalo.edu.

Corresponding author: *Ju Ren*.

Manuscript received *** **, 2018; revised ** **, 2018.

Digital Object Identifier: 10.1109/TIFS.2019.2897874

while keeping the quality of generated data. It consequently motivates the study on how to efficiently apply DP in GANs to train a differentially private generator that can generate an infinite number of high-quality data without violating the privacy of training data.

To this end, we propose GANobfuscator, a differentially private generative adversarial network to mitigate information leakage under GAN. GANobfuscator applies a combination of carefully designed noise and gradient pruning, and adopts the Wasserstein distance [6] as an approximation of the distance between probability distributions, which is a more reasonable metric than JS-divergence in GAN. We prove that the gradients in GANobfuscator can be bounded to avoid unnecessary distortion. This not only keeps the loss function with Lipschitz property but also provides a sufficient privacy guarantee. Moreover, unlike the privacy preserving deep framework mentioned in [13], whose privacy loss is proportional to the amount of data needed to be labelled in public data set, the privacy loss of our GANobfuscator is irrelevant to the amount of generated data. This makes GANobfuscator applicable to a wide variety of real-world scenarios. Summarily, our contributions of this paper are as follows.

- We propose GANobfuscator, a differentially private GAN model, in which we achieve differential privacy under GANs by adding carefully designed noise to gradients during the learning procedure.
- We develop a gradient pruning strategy that not only successfully incorporates privacy enhancing mechanisms within training deep generative model, but also significantly improves the stability and scalability of generative model training itself.
- We theoretically prove that the GANobfuscator can provide privacy guarantee with differential privacy.
- We evaluate GANobfuscator under various benchmark datasets and network structures, and demonstrate that GANobfuscator can generate high-quality data with sufficient protection for differential privacy under reasonable privacy budgets.

The remainder of this paper is organized as follows. We introduce the preliminaries in section 2 and problem description in section 3. Section 4 presents the design details of GANobfuscator, while section 5 shows our experimental results. Finally, we review the related works in section 6, followed by a conclusion in section 7.

II. Preliminaries

In this section, we first review two concepts used in our work, namely, differential privacy [14] and generative adversarial network (GAN) [15]. Then, we briefly introduce the moments accountant approach [11] that is applied in measuring privacy guarantee. The mathematical notations frequently used in this paper are summarized in Table I.

TABLE I: Frequently Used Symbols

Symbol	Description
ϵ, δ	Differential privacy parameters
$\mathcal{D}$	A dataset
$\mathcal{D}_1, \mathcal{D}_2$	Any two neighbouring datasets
G, D	A generator and discriminator
X, Y	Vectors in datasets
$\mathcal{A}$	A randomized algorithm
$p(x)$	The true data distribution
$\mathcal{L}$	Privacy loss
σ^2	The variance of a Normal distribution
$p_\theta(z)$	The input noise distribution of G
τ	An auxiliary input
w	The discriminator parameters
α_d, α_g	Learning rate of discriminator and generator
T_d, T_g	The number of discriminator and generator iterations
m, M	The size of batch and training data
c_g	The bound on the gradient of Wasserstein distance
$g_w(\cdot)$	The function of gradient
$S(G), S^{'}(G)$	Inception scores, Jensen-Shannon scores
$\mathcal{D}_{pub}, \mathcal{D}_{pri}$	public and private data

A. Differential Privacy

Let $\mathcal{D}$ be a sensitive dataset to be published. Differential privacy refers to the process that the dataset $\mathcal{D}$ is modified using a randomized algorithm $\mathcal{A}$, such that the output of $\mathcal{A}$ does not reveal much information about any particular tuple in $\mathcal{D}$. The formal definition of differential privacy is detailed as follow.

Definition 1. *((ϵ, δ)-differential privacy [14]). If a (randomized) algorithm $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy, for all inputs $\mathcal{D}_1$ and $\mathcal{D}_2$ differing in at most one user's one attribute value, and for all sets of possible outputs $O \subseteq Range(\mathcal{A})$, we have*

$$Pr\left[\mathcal{A}\left(\mathcal{D}_1\right) \in O\right] \leq \exp(\epsilon) \cdot Pr\left[\mathcal{A}\left(\mathcal{D}_2\right) \in O\right] + \delta, \qquad (1)$$

where $Pr\left[\cdot\right]$ denotes the probability of an event.

Intuitively, it can be derived that (ϵ, δ)-differential privacy is equivalent to ϵ-differential privacy when $\delta = 0$ [9]. Since δ is non-negative, any mechanism that satisfies ϵ-differential privacy also satisfies (ϵ, δ)-differential privacy for any value of δ. When $\delta > 0$, (ϵ, δ)-differential privacy relaxes ϵ-differential privacy by ignoring outputs of $\mathcal{A}$ with very small probability (controlled by parameter δ). In other words, an (ϵ, δ)-differentially private mechanism satisfies ϵ-differential privacy with a probability controlled by δ.

Among the mechanisms to achieve differential privacy, the widely used ones are Laplace and Gaussian noise mechanisms. We are primarily interested in the latter, because of the improved privacy bounds analysis provided by the moments accountant method described in section II-C. The Gaussian noise mechanism is defined as follows

$$f\left(\mathcal{D}\right) \triangleq f\left(\mathcal{D}\right) + \mathcal{N}\left(0, s_f^2, \sigma^2\right) \qquad (2)$$

where s_f is the sensitivity of f (i.e., $s_f = \left|f\left(\mathcal{D}_1\right) - f\left(\mathcal{D}_2\right)\right|$ for $f : \mathcal{D} \to \mathbb{R}$), and $\mathcal{N}\left(0, s_f^2, \sigma^2\right)$ is the Gaussian distribution with the mean 0 and the standard deviation $s_f\sigma$.

According to Definition 5, the added noise protects the membership of a data point in the dataset. For example, when conducting a clinical experiment, a person may not want the observer to know that he or she is involved in the experiment. This is due to the fact that the observer can link the test results to the appearance/disappearance of certain person and harm the interest of that person. A proper membership protection would ensure that replacing this person with another one will not affect the result too much. This property holds only if the algorithm itself is randomized, i.e., the output is associated with a distribution. And this distribution will not change too much if certain data is perturbed or even removed. This is exactly what differential privacy tries to achieve.

Privacy composition will be useful to understand how privacy parameters for each steps of an algorithm compose into privacy guarantees for the entire algorithm. The following useful theorem is a special case of a theorem proven by Dwork, Rothblum and Vadhan [16].

Theorem 1. *(Privacy Composition [16]). Let $\epsilon > 0, \delta < 1$, and let $A_i (0 \leq i \leq \mathcal{T})$ be a non-interactive privacy mechanism which satisfies ϵ_i-differential privacy.*

$$\epsilon_i \leq \frac{\epsilon}{\sqrt{8\mathcal{T}\log\left(\frac{1}{\delta}\right)}},$$

where $\mathcal{T}$ is a positive integer. Then the output of mechanism $A(\mathcal{D}) = (A_1(\mathcal{D}), ..., \mathcal{A}_i(\mathcal{D}))$ over the database $\mathcal{D}$ is (ϵ, δ)-differential privacy.

B. GAN and WGAN

Generative adversarial network [3] simultaneously trains two models: a generative model G that transforms input distribution to output distribution that approximates the real data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than the output of G. Let $p_\theta(z)$ be the input noise distribution of G and $p(x)$ be the real data distribution. GAN aims at training G and D to play the following two-player minimax game with value function $V(G, D)$

$$\min_G \max_D V(G, D) = \mathbb{E}_{x\sim p(x)}\left[\log\left(D\left(x\right)\right)\right] + \mathbb{E}_{z\sim p_\theta(z)}\left[\log\left(1 - D\left(G\left(z\right)\right)\right)\right]. \tag{3}$$

WGAN [6] improves GAN by using the Wasserstein distance instead of the Jensen-Shannon divergence as the objective function. It solves a different two-player minimax game given by

$$\min_G \max_{w\in W} \mathbb{E}_{x\sim p(x)}\left[f_w\left(x\right)\right] - \mathbb{E}_{z\sim p_\theta(z)}\left[f_w\left(G\left(z\right)\right)\right], \tag{4}$$

where w is a discriminator parameter, and functions $\left\{f_w\left(x\right)\right\}_w \in W$ are all K-Lipschitz (with respect to x) for some K. Our approach exploits such K-Lipschitz property in WGAN and solves Formula 4 in a differentially private manner.

C. Moments Accountant Approach

To derive the final composition of privacy budget, we adopt the moment accountant mechanism [11]. The moments accountant can keep track of a bound on the moments of the privacy loss random variable (defined below in Eq. 5). It generalizes the standard approach of tracking (ϵ, δ) and using the strong composition theorem.

Definition 2. *Let $\mathcal{A} : \mathcal{D} \to \mathbb{R}$ be a randomized mechanism, $\mathcal{D}_1$ and $\mathcal{D}_2$ be a pair of adjacent databases, and τ be an auxiliary input. The moments accountant of α at the λ-th moment is defined as*

$$\alpha_{\mathcal{A}}\left(\lambda\right) \triangleq \max_{\tau,\mathcal{D}_1,\mathcal{D}_2} \alpha_{\mathcal{A}}\left(\lambda; \tau, \mathcal{D}_1, \mathcal{D}_2\right) \tag{5}$$

where $\alpha_{\mathcal{A}}\left(\lambda; \tau, \mathcal{D}_1, \mathcal{D}_2\right) \triangleq \log \mathbb{E}\left[\left(\lambda L\left(\mathcal{A}, \tau, \mathcal{D}_1, \mathcal{D}_2\right)\right)\right]$ is a moment-generating function.

Moments accountant can be seen as the "worst situation" of the moment generating function. The definition of moments accountant enjoys good properties as mentioned in [11], where the composability property shows that the overall moments accountant can be easily bounded by the sum of moments accountant in each iteration. It brings a result that privacy is proportional to iterations. The tail bound can also be applied in privacy guarantee. We will use this theorem to deduce our own result in subsection 4.3. Comparing with strong composition theorem [16], moments accountant saves a factor of $\sqrt{\log\left(T_d/\delta\right)}$, where T_d is the number of discriminator iterations. According to Definition 5, for a large iteration, this is a significant improvement.

III. Problem Description

We investigate a setting where a data holder would like to publish a dataset $\mathcal{D}$ in a privacy preserving fashion. Each row in $\mathcal{D}$ contains both private variables (represented by Y), public variables (represented by X) and $Y \in X$. The goal of the data holder is to generate $\hat{X}$ in a way to satisfy that: (a) $\hat{X}$ is as good as the representation of X, and (b) an adversary cannot use $\hat{X}$ to reliably infer Y. To tackle this problem, our objective is to design tailored techniques to mitigate information leakage under GANs with DP by adding carefully designed noise to gradients during the learning procedure, and retain desirable data utility in the released model. Our method includes two learning components: a generator, whose task is to output a sanitized version of the public variables (subject to some distortion constraints) and a discriminator, whose task is to learn the private variables from the sanitized data. The generator and discriminator achieve their goals by competing in a constrained minimax, zero-sum game. It requires that the generator is designed to minimize the discriminator's performance in inferring Y reliably, and can adopt the best inference strategy to maximize its utility.

To show that the GANobfuscator indeed preserves differential privacy, we demonstrate that the parameters of the generator (through discriminator parameters) guarantee differential privacy with respect to the sampled training data. Hence, any generated data from the generator will not disclose the privacy of the training data. Through the moments accountant approach, we can compute the final composition result. Since the privacy bound produced by the strong composition theorem is often too loose, we exploit the moments accountant technique developed by Abadi et al. [11] for analyzing their DP-SGD algorithm. To give the main idea of the method, we define the privacy loss as follow.

Definition 3. *Let $\mathcal{A} : \mathcal{D} \to \mathbb{R}$ be a randomized mechanism, $\mathcal{D}_1$ and $\mathcal{D}_2$ be a pair of adjacent databases, and τ be an auxiliary input. For an output $o \subset \mathbb{R}$, the privacy loss at o is defined as*

$$\mathcal{L}\left(o; \mathcal{A}, \tau, \mathcal{D}_1, \mathcal{D}_2\right) \triangleq \log \frac{Pr\left[\mathcal{A}\left(\tau, \mathcal{D}_1\right) = o\right]}{Pr\left[\mathcal{A}\left(\tau, \mathcal{D}_2\right) = o\right]}. \tag{6}$$

And the privacy loss random variable $\mathcal{L}\left(o; \mathcal{A}, \tau, \mathcal{D}_1, \mathcal{D}_2\right)$ is defined as $\ell\left(\mathcal{A}(\mathcal{D}_1); \mathcal{A}, \tau, \mathcal{D}_1, \mathcal{D}_2\right)$.

Note that, since we assume that the supports of two distributions associated with $\mathcal{A}(\tau, \mathcal{D}_1)$ and $\mathcal{A}(\tau, \mathcal{D}_2)$ are generally the same, it is safe to evaluate them at the same point o. This is a critical assumption since if there is an area s in $\mathcal{A}(\tau, \mathcal{D}_1)$ but not in $\mathcal{A}(\tau, \mathcal{D}_2)$, the evaluating $\mathcal{L}\left(\mathcal{A}, \tau, \mathcal{D}_1, \mathcal{D}_2\right)$ in s will result in ∞ and violate the privacy. In conclusion, the moments accountant method has the bounds on the moments of the privacy loss random variable and then leverages Markov inequality to obtain the tail bound on this random variable with regard to ϵ and δ.

IV. Design of GANobfuscator

In this section, we elaborate the design of GANobfuscator, a differentially private generative adversarial network to mitigate information leakage, and retain desirable utility in the released model.

A. GANobfuscator Framework

In most real-world problems, the true data distribution $p(x)$ is unknown and needs to be estimated empirically. Since we are primarily interested in data synthesis, we will turn to generative models, and in particular we are going to use GANs as the mechanism to estimate $p(x)$ and draw samples from it. If trained properly, GANobfuscator will mitigate sensitive inference during the data analysis.

The framework of GANobfuscator is shown in Figure 1. Sensitive data X is fed into a discriminator D with a privacy-preserving layer. This discriminator is used to train a differentially private generator G to produce a private artificial dataset $\tilde{X}$. Similar to the work on differentially private deep learning (e.g., [11]), GANobfuscator achieves DP by injecting random noise in the training procedure (e.g., stochastic gradient descent [17]). A alternative solution is to inject noise in training both G and D, the minimax game formulation however makes it difficult to tightly estimate the privacy loss, resulting in excessive degradation in the produced models.We opt to add random perturbation only in training D. The rationale behind our design choice is as follows. First, the real data is directly accessible only by D; thus, it suffices to control the privacy loss in training D. Second, in comparison with G, which often employs building blocks such as batch normalizations [18] and residual layers [19] in order to generate realistic samples, D often features a simpler architecture and a smaller number of parameters, which make it possible to tightly estimate the privacy loss.

Despite the fact that the generator does not have access to the real data X in the training process, one cannot guarantee differential privacy because of the information passed through with the gradients from the discriminator. A simple high level example will illustrate such breach of privacy. Let the datasets $X, X^{'}$ contain small real numbers. The only difference between these two datasets is the number $x^{'} \in X^{'}$, which happens to be extremely large. Since the gradients of the model depend on $x^{'}$, one of the updates of the discriminator trained on $X^{'}$ may be very different from the rest, and this difference will the be propagated to the generator breaking privacy in general case.

B. The Implementation of GANobfuscator

Our method focuses on preserving the privacy during the training procedure instead of adding noise on the final parameters directly that usually suffers from low utility. We add noise on the gradient of the Wasserstein distance with respect to the training data. The parameters of discriminator can be shown to guarantee differential privacy with respect to the sample training data. We note that the privacy of the data that has not been sampled for training is guaranteed naturally. This is because replacing these data won't cause any change in output distribution, which is equivalent to the case of $\epsilon = 0$ in Definition 5. The parameters of generator can also guarantee differential privacy with respect to the training data. This is because there is a post-processing property of differential privacy [7], which says that any mapping (operation) after a differentially private output will not invade the privacy. Here the mapping is in fact the computation of parameters of generator and the output is the differentially private parameter of discriminator. Since the parameters of generator guarantee differential privacy of data, it is safe to generate data after training procedure. In short, this also means that even if the observer gets generator itself, there is no way to invade the privacy of training data.

The GANobfuscator procedure is summarized in Algorithm 1. At a high level, GANobfuscator is built upon the improved WGAN framework and enforces DP by injecting random noise in updating the discriminator D. Specifically, when computing D's gradients with respect to a real sample x (line 7), we first prune the gradients by injecting the designed noise (line 8), ensuring that the sensitivity is bounded by ϵ. Then, we add random noise sampled from a Gaussian distribution. The $AdaptRate$ in line 8 and line 13 is an optimization algorithm that can adaptively adjust the learning rate according to the magnitude of gradients [20]. In line 9, the pruning guarantees that $\{f_w(x)\}_{w \in W}$ are all K_w-Lipschitz with respect to x for some unknown K_w and act in a way to bound the gradient from each data. Additionally, we use a privacy accountant similar to [21] to track the cumulative privacy loss. This process iterates until convergence or exceeding the privacy budget.

C. Privacy Analysis of GANobfuscator

In order to use moments accountant, we need $g_w\left(x^{(i)}, z^{(i)}\right)$ to be bounded (by pruning the norm in Algorithm 1 in [11]) and add noise according to this bound. We do not prune the norm of $g_w\left(x^{(i)}, z^{(i)}\right)$, instead we show that by only pruning

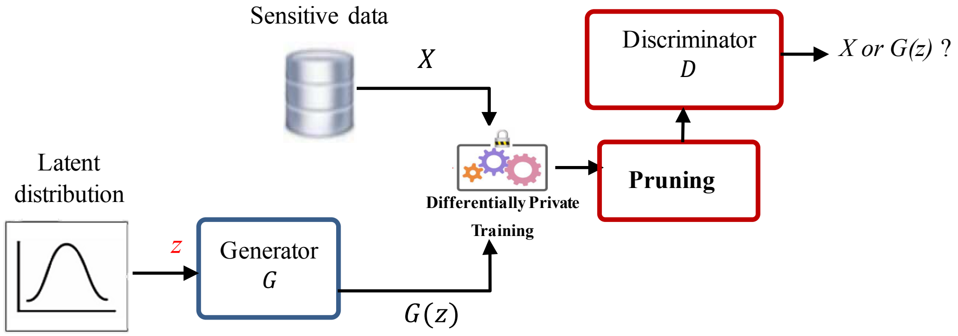


Fig. 1: The Framework of GANobfuscator

Algorithm 1: Mitigating Information Leakage under GAN via Differential Privacy

```
Input: $\alpha_d$, learning rate of discriminator. $\alpha_g$, learning
       rate of generator. $c_p$, constant for parameter
       pruning. $m$, batch size. $M$, total number of
       training data in each discriminator iteration. $T_d$,
       the number of discriminator iterations per
       generator iteration. $T_g$, generator iteration. $\sigma$,
       noise scale. $c_g$, the bound on the gradient of
       Wasserstein distance with respect to weights.
Output: Differentially private generator $G$
Initialize discriminator parameters $w_0$, generator
  parameters $\lambda$;
for $t_1 = 1, 2, \cdots, T_g$ do
    for $t_2 = 1, 2, \cdots, T_d$ do
        Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim p\left(z\right)$ a batch of prior
          samples;
        Sample $\left\{x^{(i)}\right\}_{i=1}^{m} \sim p\left(x\right)$ a batch of real data;
        for $i = 1, 2, \cdots, m$ do
            // computing generator's gradients
            $g_w\left(x^{(i)}, z^{(i)}\right) \leftarrow$
              $\nabla_w \left[f_w\left(x^{(i)}\right) - f_w\left(g\left(z^{(i)}\right)\right)\right]$;
        // pruning and perturbation
        $\tilde{g}_w \leftarrow \frac{1}{m}\left(\sum_{i=1}^{m} g_w\left(x^{(i)}, z^{(i)}\right) + N\left(0, \sigma_n^2 c_g^2 I\right)\right)$;
        // updating the discriminator
        $w^{(t_2+1)} \leftarrow w^{(t_2)} + \alpha_d \cdot AdaptRate\left(w^{(t_2)}, \tilde{g}_w\right)$;
        // updating the parameters
          $w^{(t_2+1)} \leftarrow prune\left(w^{(t_2+1)}, -c_p, c_p\right)$;
    // updating privacy accountant
    Update $\mathcal{L}$ with $(\sigma, m, T_g)$;
    Sample $\left\{z^{(i)}\right\}_{i=1}^{m} \sim p\left(z\right)$, another batch of prior
      samples;
    $g_w \leftarrow -\nabla_G \frac{1}{m}\sum_{i=1}^{m} f_w\left(g\left(z^{(i)}\right)\right)$;
    // updating generator
    $G^{(t_1+1)} \leftarrow G^{(t_1)} - a_g \cdot AdaptRate\left(G^{(t_1)}, g\right)$;
Return $G$;
```

on w we can automatically guarantee a bound of the norm of $g_w\left(x^{(i)}, z^{(i)}\right)$.

Lemma 1. *Under the condition of Alg. 1, assume that the activation function of the discriminator has a bounded range and bounded derivatives everywhere, i.e.,* $\sigma\left(\cdot\right) \leq B_\sigma$ *and* $\sigma'\left(\cdot\right) \leq B_{\sigma'}$*, and every data* x *satisfies* $\|x\| \leq B_x$*, then* $g_w\left(x^{(i)}, z^{(i)}\right) \leq c_g$ *for some constants.*

Proof: Without loss of generality, we assume f_w is implemented using a fully connected network. Let H be the number of layers except input layer. Let $W^{\{l\}}$ be the l-th weight matrix $(l = 1, ..., H)$ whose element $W_{ij}^{(l)}$ is the weight connecting jth node in layer $l-1$ to ith node in layer l. Let $D^{(l)}$ be the diagonal Jacobian of nonlinearities of l-th layer. Thus, we have

$$D_{ij}^{(l)} = \begin{cases} \sigma'\left(w_{i,:}^{(l)}\sigma\left(z^{(l-1)}\right)\right) & if \quad i = j \\ 0 & if \quad i \neq j \end{cases} \tag{7}$$

where $w_{i,:}^{(l)}$ is the ith row of $W^{(l)}$ and $\sigma\left(z^{(l-1)}\right)$ is the output of the $(l-1)$th layer. The following fact is well known from the back-propagation algorithm on a fully connected network, i.e.,

$$\delta^{(H)} = \nabla_\alpha \mathcal{L} \odot \sigma'\left(z^{(H)}\right), \tag{8}$$

$$\delta^{(l)} = \left(\left(W^{(l+1)}\right)^T \delta^{(l+1)}\right) \odot \sigma'\left(z^{(l)}\right), \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial W_{jk}^{(l)}} = \alpha_k^{(l-1)} \delta_j^{(l)}, \tag{10}$$

where $\mathcal{L}$ is the privacy loss function, $z^{(l)}, \alpha^{(l)}$ and $\delta^{(l)}$ are the input, output and error vector of layer l, respectively. For equation (10), when $l = 2, \cdots, H$:, we have

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \delta^{(l)}\left(\alpha^{(l+1)}\right)^T \\
&= \left(D^{(l)}(W)^{(l+1)}\right)^T \delta^{(l+1)}\left(\alpha^{(l-1)}\right)^T \\
&= \left(D^{(l)}\left((W)^{(l+1)}\right)^T \ldots D^{(H-1)}\left(W^{(H)}\right)^T \delta^{(H)}\right) * \left(\alpha^{(l+1)}\right)^T \\
&= \left(D^{(l)}\left((W)^{(l+1)}\right)^T \ldots D^{(H-1)}\left(W^{(H)}\right)^T\right) \\
&* \left(\alpha^{(l-1)}\right)^T \sigma'\left(z^{(H)}\right).
\end{aligned} \tag{11}$$

Take $\frac{\partial \mathcal{L}}{\partial W^{(l)}}$ as an example:

$$\left[D^{(l)}\left(W^{(l+1)}\right)^T\right]_{ij} \leq c_p B_{\sigma'}$$

$$\left[D^{(l)}\left(W^{(l+1)}\right)^T D^{(l+1)}\left(W^{(l+2)}\right)^T\right]_{ij} \leq \left(c_p B_{\sigma'}\right)^2 m_{l+1}$$

where we assume that $c_p \leq \frac{1}{m_{l+1} B_{\sigma'}}$. Here m_{l+1} is the number of nodes in the $l+1$th layer. Thus, we have

$$\left[\prod_{l=l_0}^{H-1} D^{(l)}\left(W^{(l+1)}\right)^T\right]_{ij} \leq \left(c_p B_{\sigma'}\right)^{H-l_0} \prod_{l=l_0}^{H-1} m_{l+1}. \tag{12}$$

Because of the assumption that $\sigma(\cdot) \leq B_\sigma$, we have $\sigma(\cdot) \leq B_\sigma$. Combining it with equation (11), we have $\left[\frac{\partial \mathcal{L}}{\partial W^{(l)}}\right]_{ij} \leq c_p B_\sigma B_{\sigma'}^2$ and therefore we have

$$\begin{aligned}
\left\|g_w\left(x^{(i)}, z^{(i)}\right)\right\| &= \left\|\nabla_w\left(f_w\left(x^{(i)}\right) - f_w\left(g_\theta\left(z^{(i)}\right)\right)\right)\right\| \\
&\leq 2\left\|\nabla_w f_w\left(x^{(i)}\right)\right\| = 2\sum_l \sum_{ij}\left[\frac{\partial C}{\partial W^{(l)}}\right]_{ij} \\
&\leq 2 c_p B_\sigma B_{\sigma'}^2 \sum_{k=1}^{H-1} m_k m_{k+1} = c_g,
\end{aligned}$$

where the boundness of $g_w\left(z^{(i)}\right)$ comes from the choice of sigmoid activation in the last layer of the generator. Note that, when computing c_g, we need to take into consideration the dropout rate, weight sparsity, connection percentage of convolutional nets, and other factors. ■

Note that activation functions like ReLU (and its variants) and Softplus have unbounded B_σ. This will not affect our result because both the data and weights are bounded, which guarantees that the output of each node in each layer is bounded. The boundness of data comes from a common fact that each data element has a bounded range.

Then, we have the following lemma 2 which guarantees DP for discriminator training procedure.

Lemma 2. *Given the sampling probability $q = \frac{m}{M}$, the number of discriminator iterations in each inner loop T_d and privacy violation δ, for any positive ϵ, the parameters of discriminator guarantee (ϵ, δ) differential privacy with respect to all the data used in that outer loop if it satisfies:*

$$\sigma_n = \frac{2q\sqrt{T_d \log\left(\frac{1}{\delta}\right)}}{\epsilon} \tag{13}$$

Proof: The DP guarantee for the discriminator training procedure follows the intermediate result [11]. We need to find an explicit relation between σ_n and ϵ, i.e., how much noise standard deviation σ_n we need to impose on the gradient so that we can guarantee a privacy level ϵ, with small violation. Combining $T_d q^2 \lambda^2/\sigma^2 \leq \lambda\epsilon/2$ and $e^{-\lambda\epsilon/2} \leq \delta$ in Theorem 2, we can get the result by letting the equality hold. ■

Lemma 2 quantifies the relation between noise level σ_n and privacy level ϵ. It shows that for fixed perturbation σ_n on gradient, a larger q leads to less privacy guarantee (i.e., a larger ϵq). This is indeed true since when more data are involved in computing discriminator w, less privacy is assigned on each of them. Also, more iterations (T_d) lead to less privacy because the observer gives more information (specifically, more accurate gradient) for data. This requires us to choose the parameters carefully in order to have a reasonable privacy level. Finally we have the following theorem as the privacy guarantee of the parameters of the generator.

Using moments accounting [11], we can have the following theorem.

Theorem 2. *The output of generator learned in Algorithm 1 guarantees $\left(\mathcal{O}\left(q\epsilon\sqrt{T_d}\right), \delta\right)$-differential privacy.*

Proof: The privacy guarantee is a direct consequence from Lemma 2 followed by the post-processing property of differential privacy [7]. According to the key idea of moments accounting [11] and the composition theorem of privacy loss, each step of training typically requires gradients at multiple layers, and the accountant accumulates the privacy loss that corresponds to all of them. For the Gaussian noise that we use, if we choose $\sigma = \sqrt{2\log\frac{1.25}{\delta}}/\epsilon$, then by standard arguments each step is (ϵ, δ)-differentially private with respect to the lot. Since the lot itself is a random sample from the database, each step is $\left(\mathcal{O}\left(q\epsilon\sqrt{T_d}\right), \delta\right)$-differential privacy with respect to the full database where $q = \frac{m}{M}$ is the sampling ratio per lot and T_d is the number of discriminator iterations. In the i-th iteration of Algorithm 1, the gradient $g^{(i)}$ is first pruned by a global bound c_g and the random noise $\eta \sim \mathcal{N}\left(0, c_g{\sigma_i}^2\right)$ is applied to $g^{(i)}$ to ensure $\left(\mathcal{O}\left(q\epsilon\sqrt{T_d}\right), \delta\right)$-differential privacy, where $\sigma_i = \sqrt{2\log(1.25/\delta)}/\epsilon$. ■

D. Optimizing GANobfuscator with Pruning

The GAN formulation is known for its training stability issue [22]. This issue is even more evident in the GANobfuscator framework, as random noise is injected in each training step. In our empirical study, it is observed that the basic GANobfuscator suffers a set of drawbacks: i) the synthesized data is often of low quality, e.g., unrealistic looking images;

and ii) it converges slower than its regular GAN counterpart, leading to excessive privacy loss, and sometimes even diverges.

To tackle these drawbacks, we propose a optimization scheme that significantly improves GANobfuscator's training stability and convergence rate. Specifically, we enhance the GANobfuscator by adaptive pruning to monitor the change of gradient magnitudes, and dynamically adjust the pruning bounds to achieve faster convergence and stronger privacy.

In Algorithm 1, the gradient pruning bound c_g is a hyperparameter that needs careful tuning. An extremely small c_g amounts to excessive truncation of the gradients, while an extremely large c_g is equivalent to overestimating the sensitivity. Both of them result in slow convergence and low utility. However, within the improved WGAN framework, it is challenging to find a near-optimal setting of c_g, due to the magnitudes of the weights and biases as well as their gradients vary greatly across different layers during the training.

To overcome these challenges, we propose to constantly monitor the magnitudes of the gradients before and during the training, and adaptively set the pruning bounds based on the average magnitudes. As shown in Algorithm 1, the DP constraint essentially influences the training in two key operations: (i) pruning: the norm of gradients is truncated by an upper bound, and (ii) obfuscation: random noise is added to the gradients. We propose to explore the opportunities to optimize these two critical operations. The gradients of all the parameters are grouped together to compute the norm. This global clipping scheme minimizes the privacy budget spent in each iteration, but introduces excessive random noise for some parameters, causing slow convergence. At the other end of the spectrum, one may clip the gradient of each parameter with a parameter-specific clipping bound, which may reduce the overall amount of random noise, but at the cost of privacy budget. Here we propose two alternative grouping strategies that strike a balance between convergence rate and privacy loss per iteration. Specifically, we assume that besides the private data $\mathcal{D}_{pri}$ to train the model, we have access to a small amount of public data $\mathcal{D}_{pub}$ which is available in many settings. During each training step, we randomly sample a batch of examples from $\mathcal{D}_{pub}$, and set the pruning bound of each parameter as the average gradient norm with respect to this batch. In our empirical study (Section V), we find that this adaptive pruning strategy leads to much faster training convergence and higher data utility.

Algorithm 2: Optimizing GANobfuscator with Pruning

```
Input: $\alpha_d$, learning rate of discriminator. $\alpha_g$, learning
    rate of generator. $c_p$, constant for parameter
    pruning. $m$, batch size. $M$, total number of
    training data in each discriminator iteration. $T_d$,
    the number of discriminator iterations per
    generator iteration. $T_g$, the number of generator
    iteration. $\sigma$, noise scale. $c_g$, the bound on the
    gradient of Wasserstein distance with respect to
    weights, $\mathcal{D}_{pub}$, public data, $\mathcal{D}_{pri}$, private data.
Output: Differentially private generator $G$
Initialize discriminator parameters $w_0$, generator
  parameters $\lambda$;
for $t = 1, 2, \cdots, T_g$ do
    // computing gradients of public data
    Sample $\{\tilde{x}_i\}_{i=1}^{m_{pub}} \sim \mathcal{D}_{pub}$;
    $\{g^{(i)}\}_{i=1}^{m_{pub}} \leftarrow$ Improved WGAN-Gradient
      $(\{x_i\}_{i=1}^{m_{pub}}, m_{pub})$;
    // computing gradients of real data
    Sample $\{x_i\}_{i=1}^{m_{pub}} \sim \mathcal{D}_{pri}$;
    $\{g^{(i)}\}_{i=1}^{m} \leftarrow$ Improved WGAN-Gradient
      $(\{x_i\}_{i=1}^{m}, m)$;
    for $j = 1, 2, \cdots, m$ do
        //pruning and perturbation
        $\epsilon \sim N\left(0, \quad (c_j\sigma_j)^2 \quad\right)$;
        $g_j^{(i)} \leftarrow g_j^{(i)} \max\left(1, \left\|g_j^{(i)}\right\|_2 / c_j\right) + \epsilon$;
        // updating privacy accountant
        Update $\mathcal{L}$ with $(\sigma, m, T_g)$;
        $w \leftarrow \{w_j\}_{j=1}^{m}$;
    // updating generator
    Sample $\{z_i\}_{i=1}^{m} \sim p_\theta$;
    // computing cumulative privacy loss
    $\delta \leftarrow$ query $\mathcal{L}$ with $\epsilon_0$;
Return G;
Procedure Improved WGAN-Gradient $\left(\{x_j\}_{j=1}^{m}, m\right)$;
for $j = 1, \cdots, m$ do
    sample $z \sim p_\theta$, $\lambda \sim \Phi[0, 1]$;
    $\tilde{x} \leftarrow \lambda x_j + (1 - \lambda) G(z)$;
    $\ell^{(j)} \leftarrow D(G(z)) - D(x_j) + c_g(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2$;
    $g^{(j)} \leftarrow \nabla_w \ell^{(j)}$;
Return $\{g^{(j)}\}_{j=1}^{m}$;
```

V. Experimental Evaluation

In this section, we conduct extensive experiments to evaluate the performance of GANobfuscator on three benchmark datasets (MNIST[1], LSUN[2] and CelebA [3]) in terms of the quality of generated data, privacy level and data utility.

A. Experimental Setting

In our experiments, we use three benchmark datasets:

- MNIST, which consists of 70K handwritten digit images of size 28 × 28, split into 60K training and 10K test samples.
- LSUN, which contains around one million labelled images of size 64×64, for each of the 10 scene categories.
- CelebA, which comprises 200K celebrity face images of size 48×48, each with 40 attribute annotations.

In this experiment, we set the learning rate of discriminator α_d and generator α_g to be 5.0×10^{-5}. The parameter prune constant c_p is 1.0×10^{-2} such that the weights of discriminator will be pruned back to $[-c_p, +c_p]$. Hence, the sample

[1] MNIST : http://yann.lecun.com/exdb/mnist/

[2] LSUN: http://lsun.cs.princeton.edu/2017/

[3] CelebA: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

probability q is $\frac{m}{M} = \frac{64}{6\times 10^4} \approx 1.1 \times 10^{-3}$. The noise scale δ is 10^{-5}, and the number of iterations on discriminator (T_d) and generator (T_g) are 5 and 5×10^5, respectively. Since we use leaky ReLU as the activation function on discriminator network and ReLU on generative network, we have $B_{\sigma'} \leq 1$ where $B_{\sigma'}$ is the bound on the derivative of the activation function. Dimension of z is 100 and every coordinate is within $[-1, 1]$. We adopt a similar network structure of DCGAN [23] with noise generation and inference parts to protect data privacy. Its effectiveness has been verified in [6]. To impose a certain level of noise on the network, we choose Gaussian noise with zero mean (hence no bias) and multiple values of standard deviation. Gaussian distribution is widely used in privacy-preserving algorithm (see Gaussian mechanism and its variants in [7]) and usually results in (ϵ, δ)-differential privacy. We add L_2-regularization on the weights of generator and discriminator, which has little impact on our bound in Lemma 1. For each dataset, we split the training data (which is the entire dataset if no labeling information is considered) using the ratio of $2 : 98$ as publicly available data $\mathcal{D}_{pub}$ and private data $\mathcal{D}_{pri}$, respectively. We train GANobfuscator on $\mathcal{D}_{pri}$ under the DP constraint. All the experiments are conducted on TensorFlow.

B. Relationship between Privacy Level and Generation Performance

We conduct experiments on MNIST and CelebA datasets to illustrate the relationship between the privacy level and the quality of output images from the GANobfuscator.

In these experiments, we investigate how the change of privacy level affects the image quality. According to the selection of privacy budgets in dp-GAN [24] and DPGAN [25], we select some relatively large (ranged from 0.3 to 11) in our experiments to evaluate the performance of GANobfuscator. The generated images are shown in Figure 2 and Figure 3, corresponding to three different ϵ values. The results demonstrate that the distortions of images are caused by noise instead of bad training images. Comparing the generated images with their nearest neighbors, it is clear to see that our model is not simply to memorize the training data but is capable of generating samples with unique details. As mentioned in [22], these images indeed come from actual samples of the model distributions, rather than the conditional means given samples of hidden units. More importantly, the generated images in Figure 2 and Figure 3 show that, the larger the variance of noise is, the blurrier the synthetic samples would be, when all other conditions are the same. In our proposed GANobfuscator, any observer who gets the synthetic samples can hardly know whether a data point is involved in the training procedure or not, as elaborated in Theorem 2 and illustrated by the synthetic samples in Figure 2 and Figure 3. The observer has no way to reconstruct the training images in such case and hence the privacy of data is protected. Despite the fact that smaller noise makes the accuracy higher (better generated quality), the variance of plot also decreases generally. The generation quality is little affected below specific thresholds. Thus, it is recommended to choose an ϵ larger than that threshold (add less noise) so that the generated data will not be affected much. This also demonstrates that our model successfully addresses the privacy issue mentioned previously. The privacy level (ϵ) is recommended to be tuned in a large range to guarantee good quality of synthetic samples. In addition, it can be seen from the results that our method does not suffer from mode collapse or gradient vanishing, which is an advantage inherited from the WGAN network structure.

Next, we conduct quantitative evaluation on GANobfuscator's performance. Specifically, we first compare the synthetic data against the real data in terms of the statistical properties including Inception scores [22] and Jensen-Shannon divergence [3].

Salimans et al. [22] propose Inception score to measure the quality of data generated by GAN. Formally, the Inception score of a generator G is defined as

$$S\left(G\right) = \exp\left(\mathbb{E}_{x\sim G(z)} KL\left(Pr\left(y|x\right)||Pr\left(y\right)\right)\right), \qquad (14)$$

where x is a sample generated by G and $Pr(y|x)$ is the conditional distribution imposed by a pre-trained classifier to predict x's label y. If x is similar to a real sample, we expect the entropy of $Pr(y|x)$ to be small. $Pr\left(y\right) = \int_x Pr\left(y|x = G\left(z\right)\right)dz$ is the marginal distribution of y. If G is able to generate a diverse set of samples, we expect the entropy of $Pr(y)$ to be large. Thus, by measuring the KL divergence of the two distributions, $S(G)$ captures both the quality and diversity of the synthetic data. For the MNIST and LSUN-L datasets, we use the entire training set to train baseline classifiers to estimate $Pr(y|x)$. The classifiers are tuned to achieve reasonable performance on the validation sets.

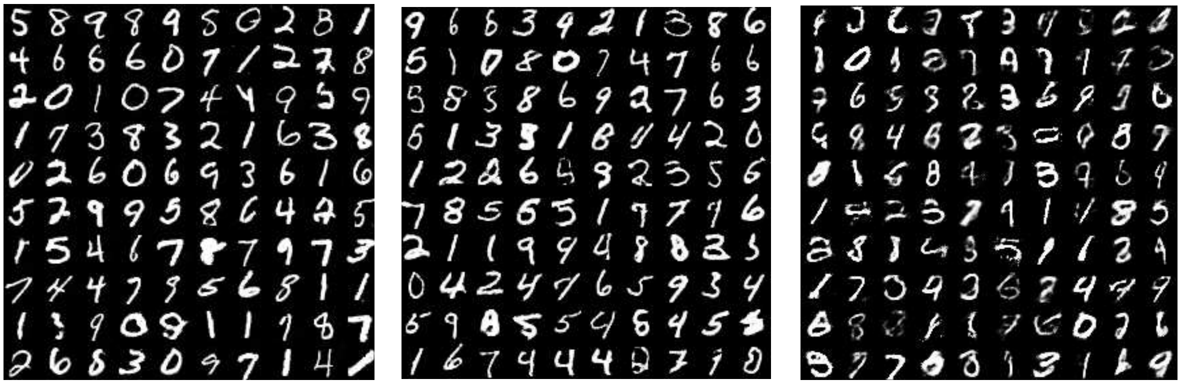
Figure 4(a) compares the Inception scores of synthetic data (generated by GANobfuscator) and real data for the MNIST and LSUN-L datasets with different privacy budgets. Intuitively, a larger value of inception score indicates better quality and diversity of data generated by generator. It also shows that GANobfuscator can synthesize data with Inception scores fairly close to the real data generated by regular WGANs (without privacy constraints, i.e., $\epsilon = \infty$). For example, in the case of MNIST, the difference between the real data and the synthetic data by GANobfuscator is less than 1.28.

To measure GANobfuscator's performance on unlabelled data (e.g., CelebA and LSUN-U datasets), we train another discriminator $D^{'}$ using the label data and test whether $D^{'}$ can discriminate the synthetic data. We consider two distributions: (i) $Pr(y|x)$ is the conditional distribution imposed by a pre-trained classifier to predict x's label y and (ii) B_p is a Bernoulli distribution with $p = 0.5$. We use the Jensen-Shannon divergence [3] of the two distributions to measure the quality of the synthetic data, i.e.,

$$S^{'}\left(G\right) = \frac{1}{2} KL\left(Pr\left(y|x\right)||B_p\right) + \frac{1}{2} KL\left(Pr\left(B_p||y|x\right)\right) \qquad (15)$$

Intuitively, a smaller value of $S^{'}(G)$ indicates that $D^{'}$ has more difficulty to discriminate the synthetic data, i.e., better quality of the data generated by G.

The Jensen-Shannon scores of the real and synthetic data (regular WGAN and GANobfuscator) on the CelebA and

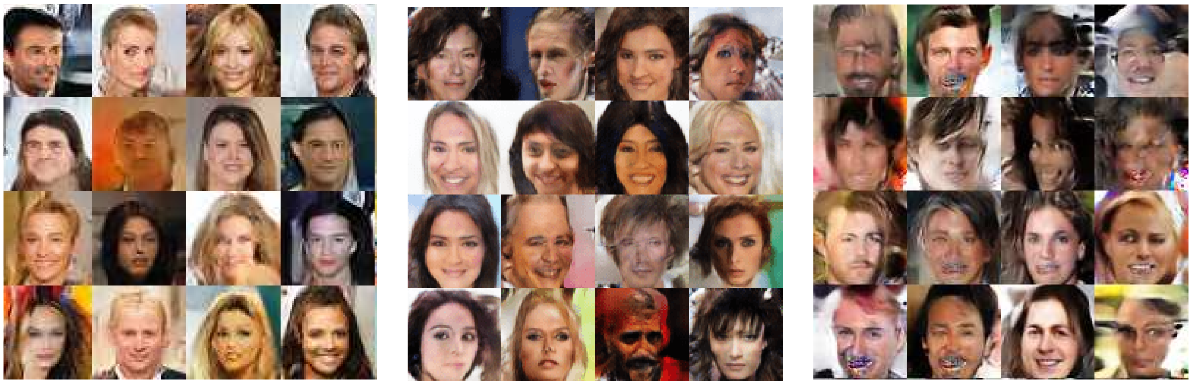


(a) $\epsilon = 8$, $\delta = 10^{-5}$

(b) $\epsilon = 4$, $\delta = 10^{-5}$

(c) $\epsilon = 2$, $\delta = 10^{-5}$

Fig. 2: Synthetic samples with three different ϵ on MNIST dataset



(a) $\epsilon = 8$, $\delta = 10^{-5}$

(b) $\epsilon = 4$, $\delta = 10^{-5}$

(c) $\epsilon = 2$, $\delta = 10^{-5}$

Fig. 3: Synthetic samples with three different ϵ on CelebA dataset

LSUN-U datasets are shown in Figure 4(b). It can be observed that the quality of the data generated by GANobfuscator is close to that generated by regular WGAN (without privacy constraints), especially in the case of LSUN-U, i.e., 0.20 versus 0.12. This is because that compared to CelebA, LSUN-U is a relatively larger dataset, enabling GANobfuscator to better capture the underlying data distribution.

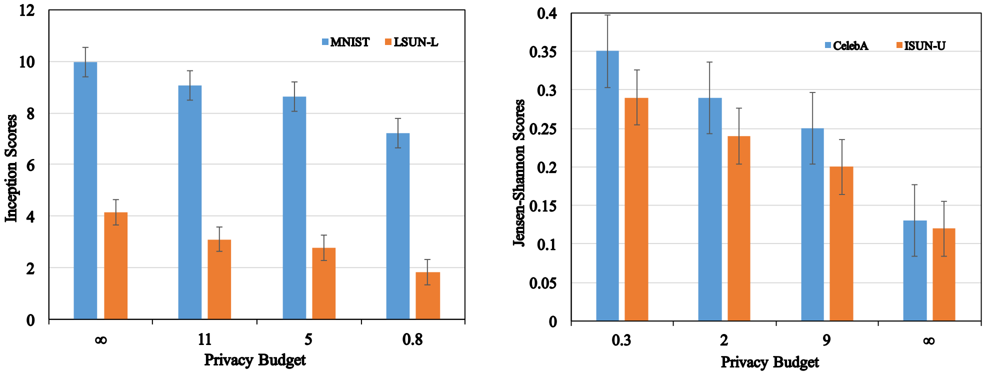
To evaluate the generation performance of GANobfuscator, we conduct the experiment by comparing with three existing solutions, namely dp-GAN [24], DPGAN [25] and GAN (no-privacy) in terms of the quality of data generated. Fig. 5(a) and Fig. 5(b) show the Inception scores and Jensen-Shannon scores of synthetic data with different solutions respectively. According to the statistical property of Inception scores and Jensen-Shannon scores, it can be observed from Fig. 5 that the quality of the data generated by GANobfuscator is superior to dp-GAN and DPGAN.

To investigate the privacy property of GANobfuscator, we adopt membership inference [26] to measure the membership risk that a person incurs if they allow their data to be used to train a model. When a record is fully known to the adversary, learning that it was used to train a particular model is an indication of information leakage through the model. In our attack experiments, we use different fractions data of CelebA dataset to show the effect of privacy budget on the accuracy of the attack. Figure 6 shows the precision of the attacks trained under GAN with different privacy budgets and different dataset sizes. It can be seen from the figure that the precision increases as privacy budget ϵ increases. This demonstrates that the attacker cannot accurately infer the distribution of the target models, which verifies that the GANobfuscator can mitigate information leakage under GAN.

Moreover, we further evaluate the privacy property of GANobfuscator by using membership inference attack and comparing with three existing solutions, namely dp-GAN [24], DPGAN [25] and GAN (no-privacy). Fig. 7 shows the precision of the inference attack for the CelebA dataset under different solutions. We can also observe from the figure that GANobfuscator is superior to dp-GAN, DPGAN and GAN (no-privacy) in terms of the ability of defending against membership privacy attacks.

C. *Effectiveness of Optimized GANobfuscator*

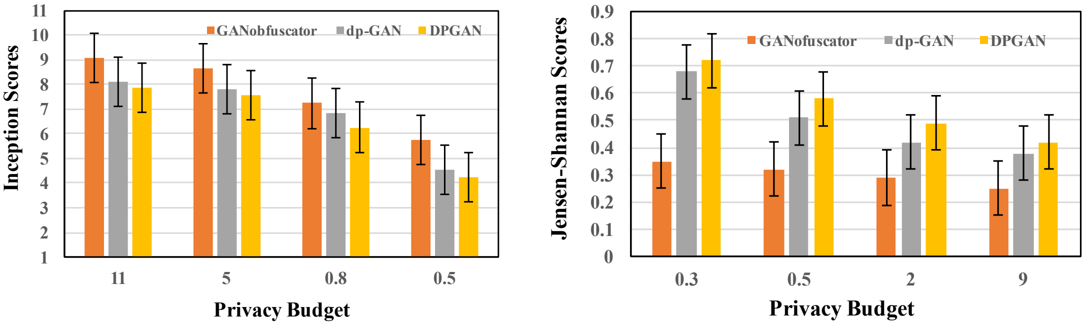
To evaluate the effectiveness of pruning optimization in GANofuscator, we plot the Wasserstein distance with different privacy budgets when applying GANobfuscator on MINST,



(a) Inception scores of synthetic data on the MNIST and LSUN-L

(b) Jensen-Shannon scores of synthetic data on the CelebA and LSUN-U

Fig. 4: The Inception scores and Jensen-Shannon scores of synthetic data



(a) Inception scores of synthetic data on the MNIST

(b) Jensen-Shannon scores of synthetic data on the CelebA

Fig. 5: The Inception scores and Jensen-Shannon scores of synthetic data with different solutions

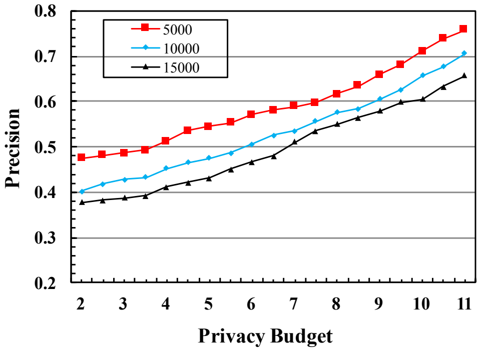


Fig. 6: The precision of the inception attack for the CelebA dataset with different sizes of datasets

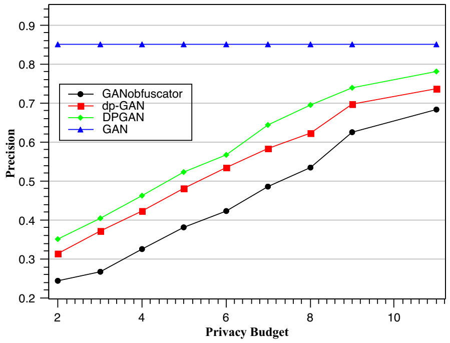


Fig. 7: The precision of the inference attack for the CelebA dataset under different solutions

which also correlates well with the visual quality of the generated samples [6]. The corresponding results are shown in Figure 8. As expected, the Wasserstein distance decreases as the training procedure goes on and converges. Despite the fluctuation caused by the min-max training itself, we can also observe that, a smaller ϵ (hence larger noise) leads to more frequent fluctuation and larger variance. This conforms to the common intuition that more noise will result in a more blurry image, which is also consistent with the results of the previous experiments. One interesting phenomena is that the peaks often appear after the convergence of Wasserstein distance. More evidences show that this might be caused by pruning the weights. The reason is that pruning weights is equivalent to adjusting the gradient g_i in directions where the corresponding gradient w_i magnitude is too large ($|w_i| > c_p$). Different from the gradient descent step (even with noise) which always changes the weight towards the optimal solution, the effect of such adjustment is hard to predict and hence might cause instability. This is especially clear when network converges. However, these peaks can be quickly eliminated during the training procedure and the network may maintain a numerical stability. This is due to the fact that the generator is in convergence stage, which is one of the advantages of adversarial networks. Therefore, it demonstrates that our system does not suffer from divergence problem, and also consequently verifies the effectiveness of pruning optimization in GANobfuscator.

D. Utility Evaluation of GANobfuscator

In this subsection, we further compare the GANobfuscator's performance in concrete analysis tasks to existing models (GAN and WGAN) on the MNIST dataset. For fixed $\epsilon = 2$ and $\delta = 10^{-5}$, we consider to use the synthetic data in a semi-supervised classification task. In such a task, we possess a small amount of public, labelled data and a large amount of synthetic, unlabelled data (generated by GANobfuscator). The goal is to leverage both the labelled and unlabelled data to train a better classifier than that trained only using the limited labelled data. We randomly select different amounts of samples from the generated data, build classifiers on them and test on MNIST's testing set. Then we repeat this for 100 times and show the accuracy (in Figure 9(a)) on testing set with classifiers built from training data and generated ones with different standard deviations. It is clear that across all the datasets, this strategy significantly increases the number of samples, thereby improving the retained utility in the generative models. This experiment use classification task to demonstrate the trade-off between learning performance and privacy level. Additionally, we evaluate GANobfuscator'performance in such a task on the LSUN-5 Cat dataset. As shown in Figure 9(b), it can be observed that the accuracy of classification with GANobfuscator steadily outperforms the result with GAN. The difference is especially evident when the number of the samples ranges from 1.5×10^4 to 2.5×10^4. Therefore, we can conclude that GANobfuscator supplies valuable synthetic data for such semi-supervised classification tasks.

VI. Related Work

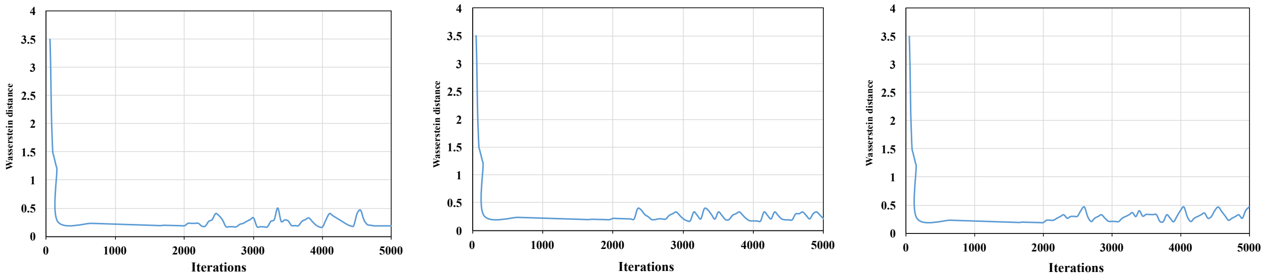
In this section, we provide a brief literature review of relevant topics: generative adversarial network and differentially private learning in neural networks.

A. Generative Adversarial Network

GAN [3] and its variants are developed in recent years with important advances from the theoretical perspective. Instead of pruning the weights, Gulrajani et al. [22] improve the training stability and performance of WGAN by penalizing the norm of the critical gradients with respect to its input.

Zhao et al. [27] introduce energy-based GAN (EBGAN), which views the discriminator as an energy function that attributes low energies to the regions near the data manifold and higher energies to other regions. The instantiation of EBGAN framework uses an auto-encoder architecture, with the energy being the reconstruction error. The behavior of EBGAN has shown to be more stable than regular GANs during training. Berthelot et al. [28] also adopt an autoencoder as a discriminator and developed an equilibrium enforcing method, paired with a loss derived from the Wasserstein distance. It improves over WGAN by balancing the power of the discriminator and the generator so as to control the trade-off between image diversity and visual quality. Qi [29] proposes a loss-sensitive GAN with Lipschitz assumptions on data distribution and loss function. It improves WGAN by allowing the generator to focus on improving poor data that is far apart from real examples rather than wasting efforts on those samples that have already been well generated, and thus improving the overall quality of generated samples. Jones et al. [30] use a differentially private version of Auxiliary Classifier GAN (AC-GAN) to simulate participants based on the population of the SPRINT clinical trial. Zheng et al. [31] leverage the GAN-generated samples by the label smoothing regularizer. Some recent research has also employed GANs to cross-modal retrieval. For example, Wang et al. [32] propose an adversarial framework for cross-modal retrieval, in which a feature projector tries to generate a modality-invariant representation to confuse the modality classifier, and the modality classifier tries to discriminate between different modalities based on the generated representation. While GANs are widely employed in various visual tasks, little effort has been made in GANs for hashing. Choi et al. [33] proposed medGAN, which is a generative adversarial framework that can successfully generate EHR. However, the approach may have privacy concerns as we discussed earlier.

Generative models have recently received a lot of attention in the machine learning community [3], [15]. Ultimately, deep generative models hold the promise of discovering and efficiently internalizing the statistics of the target signal to be generated. State-of-the-art generative models are trained in an adversarial fashion [3], [15]: the generated signal is fed into a discriminator which attempts to distinguish whether the data is real (i.e., sampled from the true underlying distribution) or synthetic (i.e., generated from a low dimensional noise sequence). Training generative models in an adversarial manner has been proved to be successful in computer vision and enabled several exciting applications. In [34], the authors use autoencoders to remove sensitive information, which do not have an obvious generative interpretation. Instead, we use a GANs-like approach to learn privatization schemes that prevent an adversary from inferring the private features. Our work differs from [35] in that we are focusing on a privacy problem rather than secrecy. Furthermore, we go beyond these works by studying a game-theoretic setting and comparing the performance of the privatization schemes learned in an adversarial fashion with the game-theoretically optimal ones. Besides, some researchers take an approach in considering an adversarial formulation to sharing images between consumers and data curators in [36]. Their frameworks are not precisely GANs-like solutions but more analogous to [37] in that they take a specific learning function for the attacker (adversary), which in turn is the loss function for the obfuscator and considers a Lagrangian formulation for the utility-privacy

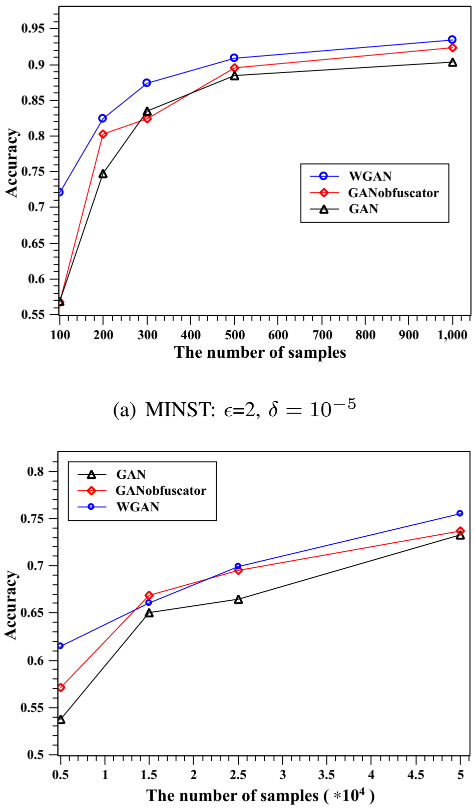


(a) $\epsilon = 8$, $\delta = 10^{-5}$

(b) $\epsilon = 4$, $\delta = 10^{-5}$

(c) $\epsilon = 2$, $\delta = 10^{-5}$

Fig. 8: Wasserstein distance with different privacy budget ϵ when applying GANobfuscator on MINST



(a) MINST: ϵ=2, $\delta = 10^{-5}$

(b) LSUN-5 Cat: ϵ=2, $\delta = 10^{-5}$

Fig. 9: The accuracy of classification on two benchmark datasets

tradeoff that the obfuscator computes. To address this issue, Huang et al. [38] propose generative adversarial privacy (GAP) that leverages recent advancements in generative adversarial networks (GANs) to allow the data holder to learn privatization schemes from the dataset itself.

Therefore, the most successful application for such generative models so far has been realistic data generation, perhaps due to the abundance of training data and inherent geometric structure.

B. Differentially Private Learning in Neural Network

Differential privacy (DP) [39], [40] and related algorithms have been widely studied in the literatures. Examples include the sensitivity-based algorithm proposed by Dwork et al. [9], which is among the most popular methods that protect privacy by adding noise to obfuscate the maximum change of data related functions. The applications of DP in deep learning have also been studied recently. Abadi et al. [35] study a gradient clipping method that imposes privacy during the training procedure. Shokri and Shmatikov [12] design a multi-party privacy preserving neural network with a parallelized and asynchronous training procedure. Papernot et al. [25] combine Laplacian mechanism with machine teaching framework. Phan *et al.* [41] enforce ε-differential privacy by injecting noise into the objective functions of the deep autoencoders at every layer and training step. After that, Phan et al. [42] further develop a “adaptive Laplace Mechanism” that could be applied in a variety of different deep neural networks while the privacy budget consumption is independent of the number of training step. A private convolutional deep belief network is present in [43] by leveraging the functional mechanism to perturb the energy-based objective functions of traditional CDBNs. Our work advances this line of research by enforcing differential privacy in the setting of training generative adversarial networks that are a new class of deep learning models.

Moreover, some researchers protect privacy of training data under GAN by injecting Gaussian noise to achieve approximate DP. Triastcyn et al. [44] employ GANs to produce artificial privacy-preserving datasets to ensure scalability and derive an empirical method to assess the risk of information disclosure in a differential privacy way. However, the approach lacks strict differential privacy guarantees and the privacy evaluation framework itself could be improved in many ways. Xie et al. [25] propose a privacy preserving generative adversarial network (DPGAN) that preserves privacy of the training data in a differentially private manner. However, DPGAN does not adopt the optimization strategy to improve the training stability and convergence speed. To address these problems, Zhang et al. [24] present a generic framework (dp-GAN) of publishing semantic-rich data in a privacy-preserving manner, and develop multi-fold system optimization strategies to improve the stability and scalability of dp-GAN, but the relationship between privacy budget and information disclosure is not quantitatively evaluated in dp-GAN.

Different from these solutions, our work focuses on preserving privacy by training a differentially private GAN that can generate infinite number of data without violating the privacy of training data. Our framework can achieve differential privacy by adding noise within the training procedure instead of adding noise on both energy functions and an extra softmax layer. Also, we adopt membership inference to measure the relationship between privacy budget and information disclosure quantitatively, and develop a Wasserstein gradient pruning strategy to improve the scalability and stability of data training.

VII. Conclusion

In this paper, we have proposed GANobfuscator, a differential privacy generative adversarial network that can mitigate information leakage by adding carefully designed noise to gradients during the learning procedure. We have theoretically proved that GANobfuscator can rigorously guarantee (ϵ, δ)-differential privacy. Moreover, we have conducted comprehensive experiments to demonstrate that GANobfuscator can generate data with good quality under reasonable privacy budgets and remain desired utility. In addition, our experimental results validates that GANobfuscator does not suffer from mode collapse or gradient vanishing during the training procedure, and hence can keep an excellent stability and scalability for model training.

For future work, we will consider reducing the privacy budget while maximizing the utility by trying different ways of pruning. In addition, GANobfuscator is formulated as an unsupervised framework, while its extensions to supervised and semi-supervised learning are attractive for the data with label information.

Acknowledgment

This work is supported by Young and Middle-Aged Scientific Research Project in the Department of Education of Fujian (No. JAT160469/B201618), Scientific Research Fund of Hunan Education Department (No. 17C0866), the Natural Science Foundation of China (Nos. 61702562 and 61702561), and the Innovation-Driven Project of Central South University (No. 2016CXS013). Kui's research is supported in part by US National Science Foundation under grant CNS-1262277.

References

[1] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[2] L. Mescheder, S. Nowozin, and A. Geiger, "Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks," *arXiv preprint arXiv:1701.04722*, 2017.

[3] 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*, 2014, pp. 2672–2680.

[4] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[5] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International Conference on Machine Learning*, 2017, pp. 214–223.

[7] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[8] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 603–618.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Third Conference on Theory of Cryptography*, 2006, pp. 265–284.

[10] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, "Privbayes: Private data release via bayesian networks," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2014, pp. 1423–1434.

[11] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[12] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[13] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *the 5th International Conference on Learning Representations*, 2017.

[14] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2006, pp. 486–503.

[15] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[16] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proceedings of 51st Annual IEEE Symposium on Foundations of Computer Science*, 2010, pp. 51–60.

[17] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2013, pp. 245–248.

[18] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning-lecture 6a-overview of mini-batch gradient descent," 2012.

[21] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 19–30.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[24] X. Zhang, S. Ji, and T. Wang, "Differentially private releasing via deep generative model," *arXiv preprint arXiv:1801.01594*, 2018.

[25] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[26] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 3–18.

[27] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[28] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[29] G.-J. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," *arXiv preprint arXiv:1701.06264*, 2017.

[30] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, and C. S. Greene, "Privacy-preserving generative deep neural networks support clinical data sharing," *bioRxiv*, p. 159756, 2017.

[31] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," *arXiv preprint arXiv:1701.07717*, vol. 3, 2017.

[32] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 154–162.

[33] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun, "Generating multi-label discrete electronic health records using generative adversarial networks," *arXiv preprint arXiv:1703.06490*, 2017.

[34] H. Edwards and A. Storkey, "Censoring representations with an adversary," *arXiv preprint arXiv:1511.05897*, 2015.

[35] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[36] A. Machanavajjhala, N. P. Landon Cox *et al.*, "Protecting visual secrets using adversarial nets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 25–28.

[37] J. Hamm, "Minimax filter: Learning to preserve privacy from inference attacks," *arXiv preprint arXiv:1610.03577*, 2016.

[38] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, p. 656, 2017.

[39] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, 2008, pp. 1–19.

[40] C. Xu, J. Ren, Y. Zhang, Z. Qin, and K. Ren, "Dppro: Differentially private high-dimensional data release via random projection," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 3081–3093, 2017.

[41] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: an application of human behavior prediction." in *AAAI*, 2016, pp. 1309–1316.

[42] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive laplace mechanism: Differential privacy preservation in deep learning," *arXiv preprint arXiv:1709.05750*, 2017.

[43] N. Phan, X. Wu, and D. Dou, "Preserving differential privacy in convolutional deep belief networks," *Machine Learning*, vol. 106, no. 9-10, pp. 1681–1704, 2017.

[44] A. Triastcyn and B. Faltings, "Generating artificial data for private deep learning," *arXiv preprint arXiv:1803.03148v2*, 2018.

Chugui Xu [S'16] (chuguixu@csu.edu.cn) received the Ph.D degree in Computer Science (2018) from Central South University, China. He has received the B.S. degree in Physics (2005) from Hunan Normal University, China. He also holds the M.S. degree in Computer Science (2010) from Hunan University of Technology, China. His research interests include Internet of Things, network computing, data privacy and security.

Ju Ren [S'13, M'16] (renju@csu.edu.cn) received the B.Sc. (2009), M.Sc. (2012), Ph.D. (2016) degrees all in computer science, from Central South University, China. During 2013-2015, he was a visiting Ph.D. student in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Currently, he is a professor with the School of Computer Science and Engineering, Central South University, China. His research interests include Internet-of-Things, wireless communication, network computing and cloud computing. He is a co-recipient of the best paper award of IEEE IoP 2018 and the most popular paper award (2015-2018) of Chinese Journal of Electronics. He currently serves/served as an associate editor for IEEE Transactions on Vehicular Technology and Peer-to-Peer Networking and Applications, and a TPC member of many international conferences including IEEE INFOCOM^^e2^^80^^9919/18, Globecom^^e2^^80^^9917, WCNC^^e2^^80^^9917, WCSP^^e2^^80^^9916, etc. He also served as a poster co-chair of IEEE MASS^^e2^^80^^9918, a track co-chair for IEEE VTC^^e2^^80^^9917 Fall and IEEE I-SPAN^^e2^^80^^9918, and an active reviewer for over 20 international journals. He is a member of IEEE and ACM.

Deyu Zhang [S'14, M'17] (zdy876@csu.edu.cn) received the B.Sc. degree (2005) in communication engineering from PLA Information Engineering University, China, and the M.Sc. degree (2012) from Central South University, China, also in communication engineering. He received his Ph.D. degree in computer science from Central South University, China, in 2016. He is now an assistant professor with the School of Software and a postdoc fellow with the Transparent Computing Lab in the School of Information Science and Engineering, Central South University, China. He was a visiting scholar with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, from 2014 to 2016. He has served as the co-chair of workshop SECIoT'1, and the guest editor for special issue on IEEE Internet of Things Journal. His research interests include stochastic resource allocation transparent computing, edge computing and IoT. He is a member of IEEE and CCF.

Yaoxue Zhang [M'17, SM'18](zyx@csu.edu.cn) received his B.Sc. degree from Northwest Institute of Telecommunication Engineering, China, in 1982, and his Ph.D. degree in computer networking from Tohoku University, Japan, in 1989. Currently, he is a professor with the School of Computer Science and Engineering, Central South University, China, and also a professor with the Department of Computer Science and Technology, Tsinghua University, China. His research interests include computer networking, operating systems, ubiquitous/pervasive computing, transparent computing, and big data. He has published over 200 technical papers in international journals and conferences, as well as 9 monographs and text-books. Currently, he is serving as the Editor-in-Chief of Chinese Journal of Electronics. He is a fellow of the Chinese Academy of Engineering.

Zhan Qin (zhanqin@buffalo.edu) is a professor in Institute of Cyberspace Research of Zhejiang University. He received his Ph.D. degree at the Ubiquitous Security and Privacy Research Laboratory (UbiSeC) in the Computer Science and Engineering Department of the State University of New York at Buffalo, NY, USA. His research interests focus on Data Privacy, Crowdsourcing Security and Smart Grid.

Kui Ren [F'16] (kuiren@buffalo.edu) is a professor in Institute of Cyberspace Research of Zhejiang University and the director of UbiSeC Lab at State University of New York at Buffalo (UB). He received his PhD degree from Worcester Polytechnic Institute. Kui's current research interest spans Cloud & Outsourcing Security, Wireless & Wearable Systems Security, and Mobile Sensing & Crowdsourcing. Kui has published 200 papers in peer-reviewed journals and conferences and received several Best Paper Awards, including IEEE ICDCS 2017, IWQoS 2017, and ICNP 2011. He received IEEE CISTC Technical Recognition Award in 2017, UB Exceptional Scholar Award for Sustained Achievement in 2016, UB SEAS Senior Researcher of the Year Award in 2015, Sigma Xi/IIT Research Excellence Award in 2012, and NSF CAREER Award in 2011. He currently serves on the editorial boards of IEEE Trans. on Dependable and Secure Computing, IEEE Trans. on Service Computing, IEEE Trans. on Mobile Computing, IEEE Wireless Communications, IEEE Internet of Things Journal, and SpingerBriefs on Cyber Security Systems and Networks. Kui is a Fellow of IEEE, a Distinguished Lecturer of IEEE, a member of ACM, and a past board member of Internet Privacy Task Force, State of Illinois.