IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 27, 2020, accepted June 21, 2020, date of publication June 30, 2020, date of current version July 9, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3005911*

# Structural Image De-Identification for Privacy-Preserving Deep Learning

**DONG-HYUN KO[1], SEOK-HWAN CHOI[1], JIN-MYEONG SHIN[1], PENG LIU[2], (Member, IEEE), AND YOON-HO CHOI[1], (Member, IEEE)**

[1]School of Computer Science and Engineering, Pusan National University, Busan 46241, South Korea
[2]College of Information Sciences and Technology, Pennsylvania State University, State College, PA 16802, USA

Corresponding author: Yoon-Ho Choi (yhchoi@pusan.ac.kr)

This work was supported by the Korea Health Technology Research and Development Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, South Korea under Grant HI19C0824.

**ABSTRACT** Due to the risk of data leakage while training deep learning models in a shared environment, we propose a new privacy-preserving deep learning (PPDL) method using a structural image de-identification approach for object classification. The proposed structural image de-identification approach is designed based on the fact that the degree of structural distortion of an image object has the greatest impact on human's perceptual system. Thus, by modifying only the structural parts of the original one using order preserving encryption(OPE), the proposed structural image de-identification approach decreases only the recognition rate by human. From the experimental results using different standard datasets, we show that the object classification accuracy of the proposed structural image de-identification method is almost the same as the deep learning performance for non-encrypted images, without revealing the original image contents including sensitive information. Also, by handling the trade-off between object classification accuracy and privacy protection for the de-identified image, we experimentally find the optimal size of input image for the proposed structural image de-identification approach.

**INDEX TERMS** Data privacy, deep learning, image encryption, structural similarity, vector graphics.

## I. INTRODUCTION

Recently, the performance of deep learning has become to exceed human ability in various application services such as language translation service, image recognition, self-driving car service and so on [1], [2]. To realize the wide deployment of deep learning, deep learning techniques such as improved deep neural network(DNN) models, optimization algorithms and data augmentation methods such as rotation, flip and shifting have rapidly developed [3]–[5]. Also, since we commonly require the large amount of computing power to make deep learning techniques work effectively [6], [7], many cloud service providers deployed cloud computing environments such as Microsoft Azure, Google Cloud AI [8], [9]. However, due to the risk of data leakage while transmitting data into the cloud server, serious privacy concerns can cause users not to use deep learning services on cloud computing environment.

For example, once an adversary penetrates cloud computing environment, all contents of user data for learning can be exposed to the adversary. Especially, it is known that compared to the text data, the exposure of the image data with dense information causes more critical privacy issues [10], [11]. To resolve the critical privacy issue, we expect cloud service providers to learn our data without exposing the private information. Let us consider an input image in Fig. 1. From the input image on client side, only a human face itself may not reveal any private information. However, an association between the human face and a context on the input image can be used to recognize a person's private information. For example, the association between the human face and a licence plate in the input image can reveal his private information such as name and age. Also, the association between the human face and a house location on background can reveal his residence or favorite travel places. Without informed consent, the associated data can be used by adversaries or cloud service providers when deriving value to cause some critical privacy issues.

As a representative solution, differential privacy technique such as the Oasis Lab's commercial platform allows the data to remain anonymous and obscured [12]. As another representative solution, privacy-preserving deep learning (PPDL)

The associate editor coordinating the review of this manuscript and approving it for publication was Yong Xiang.

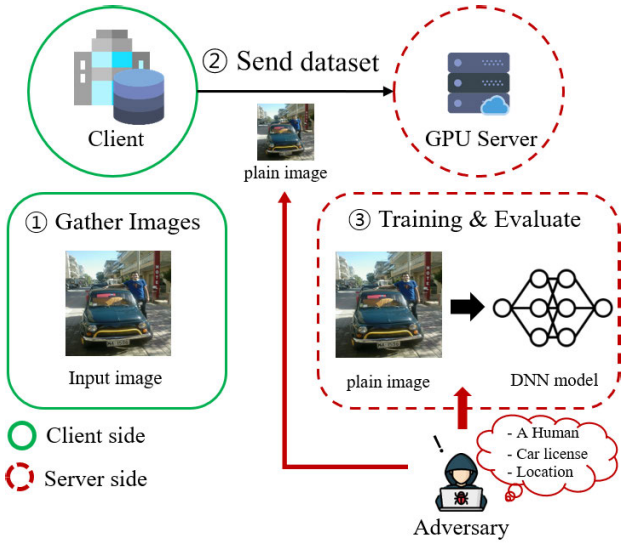



**FIGURE 1.** Example of privacy exposures in deep learning environments using cloud GPU server.

enables deep learning computation on input data without revealing the original content. When protecting the privacy of data transmitted to cloud servers, service developers using PPDL can use encrypted data in deep learning. The current state-of-the-art PPDL methods which use encrypted data can be categorized into two groups: (1) fully homomorphic encryption(FHE)-based methods [13]–[15]; (2) pixel-value-based encryption methods [16], [17]. Even though such PPDL methods were successfully deployed when training deep learning models using encrypted images directly, their usage is constrained due to the following limitations.

Even though FHE-based methods have strong encryption strength, their usage is mainly limited due to its slow evaluation(test) time, difficulty in applying the state-of-the-art DNN models and the fact that data augmentation techniques for performance improvement cannot be used in encrypted state [13]–[15]. As a representative approach to obtain good performance in the most recent deep learning models, pixel-value-based PPDL methods were proposed.

Pixel-value-based PPDL methods show the good-enough classification accuracy on color images such as CIFAR-10 [18] and ImageNet [19]. These methods have evolved because data augmentation techniques can be adopted in an encrypted state [17]. However, since pixel-value-based PPDL methods are designed using a probability-based approach, they have limitations in relying on the performance of random number generator. Also, since pixel-value-based PPDL methods require to change the RGB pixel values, they cannot directly analyze the gray-scale images.

In this paper, we propose a new type of privacy-preserving approach using structural image de-identification, which is a vector-driven approach of an image for PPDL on object classification. The proposed approach prevents private information existing within training image data from being exposed by unauthorized personnel when the data resides in a shared environment like cloud system. The proposed approach is designed to enable the use of state-of-the-art deep learning techniques for better performance, such as various DNN models and data augmentation, as well as to analyze gray-scale images directly for general purposes.

Note that even though an image object contains a huge amount of noise that does not affect the structural shape of an object, humans can still recognize the image object [20], [21]. Based on these previous observations, the proposed structural image de-identification method modifies only the structural features of image object to decrease the recognition rate of human. Thus, instead of increasing the noise level, privacy could be substantially enhanced by taking advantage of humans' sensitivity to structural change of images.

When modifying only the structural features of the image, we use the concept of vector graphics file [22]. After transforming the original input image into the vector graphics file, we shift the points(represented by $x$ and $y$) on image while maintaining their position order using order preserving encryption(OPE) [23]. OPE changes the distribution of the original data points($x$, $y$) into a targeted distribution. However, it doesn't change the order of the original data when being encrypted. As a result, the proposed structural image de-identification decreases the recognition rate for the private object by human. Also, since other attribute values on an image except the $x$- and $y$- coordinate values do not change, we can keep the high classification accuracy of deep learning.

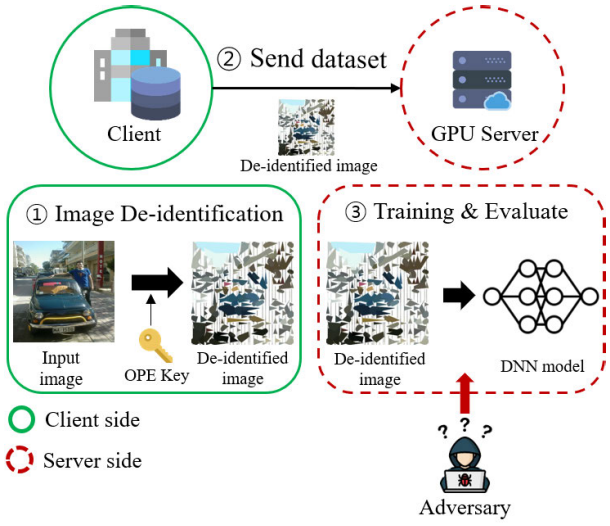



**FIGURE 2.** Operational overview of the proposed structural image de-identification PPDL approach.

In Fig. 2, let us overview how the private object in an input image can be protected by using the proposed structural image de-identification method. Even though the image data transmitted to the GPU server on clouding computing environment is exposed to an adversary, the private object in the

input image is protected because the structural shape of the image object is modified not to be identified by human and also, de-identified image cannot be restored to the original input image.

Main contributions of this paper can be summarized into three folds: (1) To the best of our knowledge, we design, implement, and evaluate a new structural image de-identification approach using OPE for the first time. The primary objective of the proposed structural image de-identification approach is to protect the privacy of the input data to DNN models for object classification while keeping the high accuracy; (2) We measure trade-off values between utility and privacy according to various parameter values of input image size. That is, we find the optimal size of input image in the context of the trade-off between object classification accuracy and privacy; (3) From the evaluation results under various parameters using different well-known standard datasets, we show the effectiveness of the proposed structural image de-identification method.

The rest of this paper is organized as follows. In section II, we describe the related works. In section III, we show some preliminary analysis results for designing the proposed structural image de-identification approach. After describing the details of the proposed structural de-identification approach in section IV, we show the evaluation results under various parameters using standard image datasets such as CIFAR-10 and ImageNet in section V. Finally, we conclude the paper in section VI.

## II. RELATED WORK

In this section, after categorizing PPDL approaches according to the usage of encryption methods, we overview the characteristics of the state-of-the-art PPDL approaches.

As a representative non-encryption method, Dwork *et al.* proposed differential privacy(DP) to provide privacy protection for individual data [24]. DP adds noise to original data and thus, generates fake dataset which have the same distribution as original data. Even though DP allows us to identify the features of the entire set of data, we cannot identify the individual sensitive data. As a practical DP-based PPDL approach, Phan *et al.* proposed the deep private auto-encoder(dPA) model for analyzing the data with added noise [25]. They perturbed the objective functions of deep auto-encoder to enforce the differential privacy. They have shown that the dPA model is very effective and efficient through theoretical analysis and experimental results. However, the dPA model cannot be used with a certain deep learning model because it is designed only for a specific model, i.e., deep auto-encoder. Abadi *et al.* also proposed a practical DP-based PPDL approach, called private stochastic gradient descent(pSGD) algorithm [26]. Since such approach applies directly to gradient computations, it can be applied to various deep learning models. However, experimental results of pSGD showed that accuracy significantly reduced into 73% for CIFAR-10 dataset compared to 94% for MNIST dataset. These DP-based PPDL methods showed the significant results in quantifying how much data are compromised and how much privacy can be protected. As a practical solution, Oasis lab's Chorus automatically applies differential privacy for general purpose data analysis. Chorus has been released as open-source for protecting individual privacy [12]. However, the DP-based PPDL models result in reducing the object classification accuracy because the large amount of added noise eventually distorts the data [27].

The state-of-the-art privacy-preserving methods which directly use encrypted data for evaluation are mainly categorized into two groups: (1) FHE-based PPDL [13]–[15], [28], [29]; (2) Pixel-value-based encryption [16], [17].

As a representative FHE-based method, Nathan Dowlin *et al.* proposed CryptoNet, which trained the deep learning models with FHE-encrypted data [13], for the first time. To compute FHE-encrypted data, CryptoNet transformed the activation and loss functions into polynomial functions. However, CryptoNet required very high computational complexity and could be trained only on CPU. To overcome the above issues from CryptoNet, Florian Bourse *et al.* proposed a new approach, called FHE-DiNN [28]. Although FHE-DiNN greatly improved the evaluation time and the size of network compared to CryptoNets, it showed low classification accuracy. Xiaoqian Jiang *et al.* proposed a new approach, called E2DM, which uses a new matrix computation mechanism to improve the evaluation time [29]. Le Trieu Phong *et al.* proposed a new approach using additively homomorphic encryption to protect the gradients over the cloud server while considering the trade-off between utility and privacy [14]. Very recently, Ahmad Al Badawi *et al.* proposed homomorphic convolutional neural networks(HCNN), which encrypts the training data using FHE, for classifying MNIST and CIFAR-10 datasets with graphics processing units(GPUs) [15]. These FHE-based PPDL approaches need to transform non-linear activation functions and loss functions into polynomial functions in order to cope with the linear operation of FHE. As a result, the FHE-based PPDL approaches cannot work with the state-of-the-art DNN models. Also, the FHE-based PPDL approaches need high computation complexity while evaluating the test data. FHE-based PPDL approaches have a limitation that they do not allow us to use data augmentation for the encrypted data.

As a representative pixel-value-based encryption approach, Masayuki Tanaka proposed block-wise pixel shuffling algorithm for 8-bit RGB image [16]. Because data augmentation must be done before encryption to improve the object classification accuracy, this method requires a lot of computing resources. Recently, Warit Sirichotedumrong *et al.* proposed a new pixel-based image encryption approach to generate Negative-Positive transformation and color shuffled image [17]. The new pixel-based image encryption approach improved the object classification accuracy by adding a simple layer, called an adaptation layer, in front of the existing DNN model. They also reduced computer resource

**TABLE 1. Characteristics of representative PPDL approaches.**

| PPDL Approach | Data Encryption | Deep Learning Model | Data Augmentation Stage | Gray-scale Image | Accuracy |
|---|---|---|---|---|---|
| DP-based Methods [12], [24]–[27] | No | Semi-Specific | After encryption | Applicable | Low |
| FHE-based Methods [13]–[15], [28], [29] | Yes | Specific | Before encryption | Applicable | High |
| Pixel-value-based encryption Methods [16], [17] | Yes | Any | After encryption | Inapplicable | High |
| Proposed Method | Yes | Any | After encryption | Applicable | High |

consumption by enabling data augmentation after encryption. However, their usage was limited to the RGB color images because the pixel-value-based encryption approach required to change the RGB pixel values.

Similar to the encryption-based PPDL approaches including FHE-based PPDL and pixel-value-based encryption approaches, the proposed structural image de-identifcation approach for PPDL uses OPE to generate encrypted input image. However, different from FHE-based PPDL approach, the proposed PPDL approach can be used with any deep learning model. The evaluation time is much faster than the other encryption-based approaches. Also, data augmentation can be applied after encryption. Different from the pixel-value-based encryption approach, the proposed PPDL approach can classify the gray-scale image well. Compared to DP-based PPDL, the proposed PPDL approach also shows the higher accuracy. We summarize the characteristics of representative PPDL approaches in Table. 1 for comparison.

## III. PRELIMINARIES

In this section, we show why we use vector graphics files instead of raster graphics images. We describe how to combine different measurement indices for the proposed structural image de-identification approach. Also, we introduce a privacy threat scenario where data privacy can be infringed when training deep learning model.

### A. RASTER GRAPHICS IMAGE V.S. VECTOR GRAPHICS IMAGE

A raster graphics image has an RGB color for each pixel [30]. Well-known image formats, e.g., JPG, PNG and BMP, belong to raster graphics image formats. According to the arrangement of RGB color channels, raster graphics images represent objects and the combination of pixel color values allows us to recognize and classify objects. However, if each pixel value changes carelessly, objects in the raster graphics image cannot be easily identified by deep learning. When analyzing the input images, whose pixel values are encrypted, using convolutional neural network(CNN) models, we actually observed the poor classification accuracy. This is because each pixel in an image has a value from 0 to 255 following modular operation and thus, a mixing and an overlap can be observed from the converted image as shown in Fig. 3. Thus, information of objects in the raster graphics image can be easily damaged.

On the other hand, vector graphics file uses geometric information such as curves and polygons to represent images.

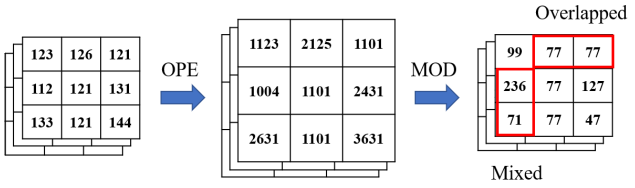



**FIGURE 3. Conversion on pixel image. Here, an overlap occurs where the original pixel values are different before conversion but, the pixel values are the same as after conversion. A mixing can be observed at the pixels where small original pixel values are converted into larger ones.**

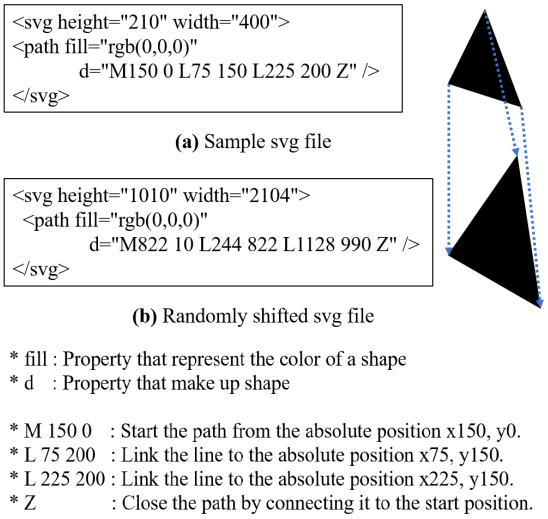



**FIGURE 4. Conversion on vector graphics file. While the color of shape is unchanged, only the structural shape slightly changes.**

Even though the size of vector graphics file varies, mathematical operations can help to render it without compromising image quality [22]. Fig. 4 (a) shows the content of a vector format file together with the corresponding vector graphics file. Note that the $<$ *path* $>$ tag in the vector format file consists of properties such as positions, line thickness, colors, and so on and thus, describes a shape of the corresponding vector graphics file. As a result of a random shift for each pixel, we can observe that the $<$ *path* $>$ tag in Fig. 4 (b) has the different position($d$) but, the same color(*fill*). As a result, even though we can observe some variation in structural shape of the original image, object in the vector graphics file can be easily maintained into the same shape.

The relationship between vector graphics and raster graphics can be expressed as shown in equation 1.

$$\begin{aligned} Vectorize(RI) &:= VI \\ Rasterize(VI) &:= RI \end{aligned} \tag{1}$$

where as $RI$ is raster graphics image and $VI$ is vector graphics image.

In optimal cases, vector images produced by vectorizing the original raster image are not different from the original raster image when viewed by the human recognition system. Likewise, raster images produced by rasterizing the original vector image are not different from the original vector image when viewed by the human perception system. In other words, as shown in equation 2, the value of dissimilarity is close to zero, and the value of similarity is close to one.

$$\begin{aligned} Dissimilarity(RI, VI) &\simeq 0; \\ Similarity(RI, VI) &\simeq 1. \end{aligned} \tag{2}$$

### B. PERFORMANCE MEASUREMENT INDICES

Most deep learning studies have used $L_p norm$ to measure the degree of modification in the original image due to adversarial perturbation. However, as Sabour *et al.* showed in [21], $L_p norm$ has limitation when measuring human's recognition rate. As shown in Fig. 5, let us consider three images, whose original images are the same. We can observe that the normalized $L_2$ distance between the original image and either the messed-up image or the darkened image is almost the same. However, humans can recognize the difference between two of them. This observation shows that only $L_p norm$ cannot measure the object classification accuracy of human's visual system.


**FIGURE 5.** $L_2 norm$ and $SSIM$ values between the original image and a perturbed image, i.e., messed up image and darkened image.

To measure the object classification accuracy of human's visual system, we use structural similarity($SSIM$) index, which was introduced by Z. Wang *et al.* [31]. The key assumption underlying $SSIM$ is that the degree of distortion of structural information has the greatest impact on perceptual quality since human visual systems are specialized in deriving structural information from images. Values of $SSIM$ range from 0 to 1. Here, the value closer to 1 indicates that human perception system identifies no difference between the original image and the converted image. As shown in Fig. 5, the $SSIM$ index between the original image and either the messed-up image or the darkened image is slightly different. From the $SSIM$ indices, we can recognize that the darkened image is more similar to original image than the messed-up image.

Based on these observations, we suggest a new measurement index, called Structural based De-Identification Measurement ($SDIM$). Since $SDIM$ is designed by combining $L_2 norm$ with $SSIM$, $SDIM$ is used to handle the trade-off between object classification accuracy and privacy under image de-identification.

### C. THREAT MODEL

By analyzing of context information between objects in the image, personal information such as personal residence and location information can be exposed unintentionally. As shown in Fig. 1, let us consider an example where the input data for deep learning are transmitted into deep learning server on cloud computing environment. Here, the input data in the cloud computing server for deep learning can be accessed without authority by someone in the deep learning environment. In the same way as MITM(man in the middle) attack, data privacy exposure can be occurred during transmission to cloud computing sever. Also, due to deep learning characteristics that require lots of training data, there is also a risk of privacy exposure if related agencies share their data. Thus, when the input data for deep learning are stored or shared externally, privacy infringement can be occurred due to unintended data exposures.

## IV. STRUCTURAL IMAGE DE-IDENTIFICATION

In this section, we describe in detail the operation of the proposed structural image de-identification for PPDL not to disclose the original image. Also, we describe how to find the optimal size of the input image in the context of the trade-off between object classification accuracy and privacy under image de-identification.

### A. OVERALL OPERATION

To understand the operation of the proposed structural image de-identification approach, we first overview the overall operation. As shown in Fig. 6, the proposed structural image de-identification approach follows three steps: (1) **Vectorization**, which transforms the original raster graphics image into vector graphics file; (2) **Position OPE**, which partially applies OPE to positions on vector graphics file; (3) **Rasterization**, which transforms the position-shifted vector graphics file into de-identified raster graphics image.

In Vectorization step, the original raster graphics image is converted to the vector graphics file as shown in Fig. 6. Only the image format changes from raster to vector but, the other features in the original image do not change. Thus, the vector graphics file has the same size as the original raster graphics image.

In Position OPE step, position values in the vector graphics file are encrypted using OPE. As shown in Fig. 6, the image

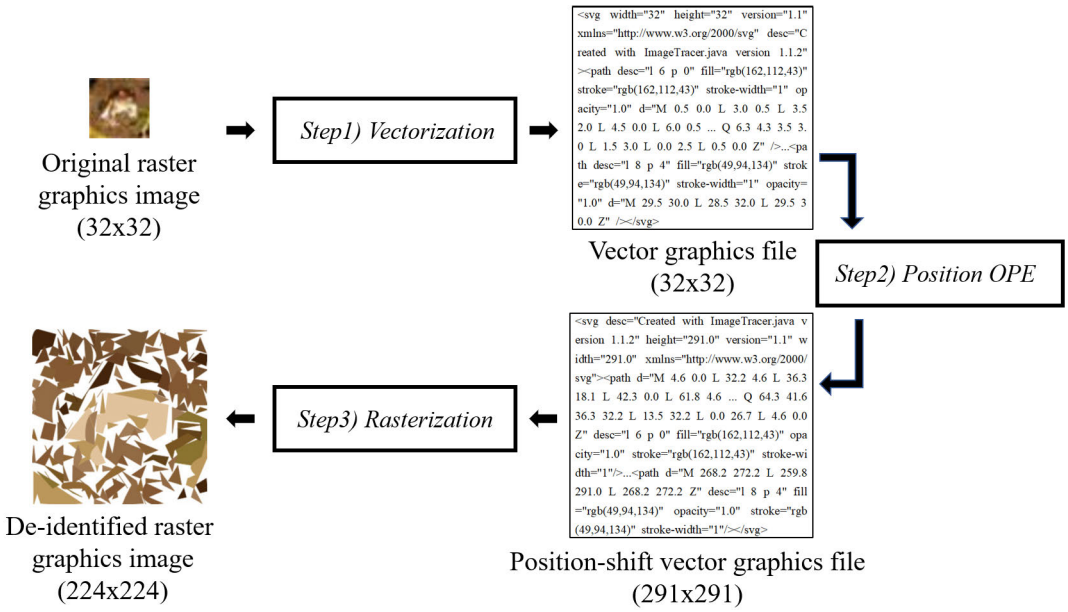



**FIGURE 6.** Overall operation of the proposed structural image de-identification approach.

size changes from 32 × 32 to 291 × 291 due to OPE-based encryption. While the position value($d$) in the $< path >$ tag changes, all other attributes remain the same as the vector graphics file before encryption.

Unfortunately, the OPE-encrypted vector graphics file cannot directly be used as an input to CNN models. Thus, in the Rasterization step, the vector graphics file transformed by position OPE is converted back to the raster graphics image. As shown in Fig. 6, the rasterization step generates a de-identified raster graphics image of the 224 × 224 size, which is a reduced size of the position-shifted vector image.

Overall operation of the proposed method can be expressed into:

$$Rasterize(PosOPE(Vectorize(OI))) := DeI, \quad (3)$$

where $OI$ is original raster graphics image and $DeI$ is de-identified raster graphics image.

### B. DETAILED OPERATION

We describe detailed operation of each step of the proposed structural image de-identification approach for PPDL.

#### 1) STEP1: VECTORIZATION

In Algorithm 1, we show how to convert the original graphics image into vector graphics file. After reading the original file, we resize the image to fit the $Param_{size}$ value(Lines 2 to 5). Results from Vectorization also vary following the optional parameters of the *Vectorize* function, i.e., $Param_{vec\_opt}$. Here, $Param_{vec\_opt}$ is a set of hyper-parameters such as the blur-radius, number of color, path-omit and so on. Thus, the value of the *vector* variable is set according to the parameter value of the target quality of vectorization such as the 'default', 'gray-scale', 'detail' and so on(Line 6). Following the configuration value stored in the *vector* variable, the original raster image is converted into the vector image(Line 7). The converted vector image is most affected by the quality, size of the original raster image. Finally, the converted vector image is stored into a vector graphics file(Line 8).

**Algorithm 1** Vectorization Procedure

1: **procedure** Vectorization($Param_{size}$, $Param_{vec\_opt}$)
2: $image$ = Read(original_path);
3: **if** $image$.size != $Param_{size}$ **then**
4: $image$ = resize($image$, $Param_{size}$);
5: **end if**
6: $vector$ = Vectorize($Param_{vec\_opt}$);
7: $vector\_image$ = $vector$.convert($image$);
8: $file$.write($vector\_image$);
9: **end procedure**

Let us consider the vectorization function changes only the graphics file format. However, after converting the original graphics image into vector graphics file, the *SSIM* value, i.e., a similarity indicator based on a person's perception system, is not close to one. Also, if the quality of the original image is low, two critical problems are observed. For example, as shown in Fig. 7, some features which are important in deep learning can disappear. In Fig. 7 (a), we observe that a bird object itself disappears after vectorization. In Fig. 7 (b), we also observe that dog's eye and nose are missing after vectorization. To address such vanishing problems in vector graphics images, we use the image size larger than 32px.

In Fig. 8, we show that as the size of the original image increases from 32×32 to 224×224, the visibility of converted


**FIGURE 7.** Object and feature vanishing examples in the Vectorization step: (a) a bird image; (b) a dog image.

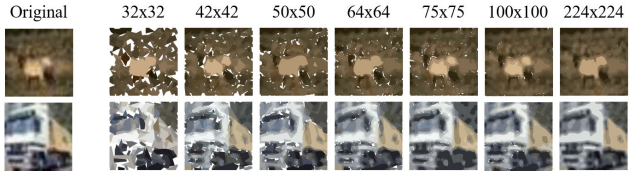



**FIGURE 8.** Vector graphics images under various resizing options after applying vectorization. Here, the above original image is the 32 × 32 size of a deer image and the below one is the 32 × 32 size of a truck image. Images on the right side of each original image show vector graphics images obtained from vectorization when gradually increasing the size of the original image from 32 × 32 to 224 × 224.

vector graphics image becomes the same as the original image. That is, from the two images in Fig. 8, we observe that a deer and the wheel of truck are clearly visible. From this observation, we optionally resize the input size before vectorization.

### 2) STEP2: POSITION OPE (POS-OPE)

In position OPE step, we encrypt the height and width values, and $d$ attribute value in the $<$ *path* $>$ tag of the vector graphics image file. As a result, even though the $x$- and $y$-coordinate values of pixels change, the order of pixel remains intact. The attributes of *stroke-width*, *color*, and *fill* in the $<$ *path* $>$ tag are also kept intact.

In Algorithm 2, we show how to convert a vector graphics file into an position-shifted vector image file. First, we set the *cipher* value to encrypt the position of each point using OPE key($Param_{key}$) and OPE range($Param_{range}$) (Line 2). Using the variable *cipher*, we set the *offset* value to prevent the loss of data when the encrypted value is negative (Line 3). After reading the vector graphics file from vectorization, we obtain a $<$ *path* $>$ tag array (Lines 4 to 5), and then obtain an $d$ attribute in each $<$ *path* $>$ tag (Lines 6 to 7). We encrypt each numerical value in *ds*, which are data segments from $d$ (Lines 8 to 14). After encrypting all $x$- and $y$-axis values, we also encrypt the height and width values of the image (Lines 16 to 22). Finally, we generate the position-shifted vector graphics file (Line 23).

To distribute the OPE-converted pixel values randomly, we set the conversion range of pixels about 100 times larger than the maximum value of the input value. That is, we set the range of OPE-converted pixel value into −99999 to 99999, which is much larger than the pixel value in the input image

**Algorithm 2** Position OPE Procedure

```
1:  procedure Pos_OPE(Param_key, Param_range)
2:      cipher = OPE(Param_key, Param_range);
3:      offset = cipher.encrypt(0);
4:      doc = minidom.parse(vector_path);
5:      paths = doc.getElementsByTag('path');
6:      for pe in paths do
7:          d = pe.getAttribute('d');
8:          ds = d.split();
9:          for p in ds do
10:             if is_number(p) then
11:                 c = cipher.encrypt(p) − offset;
12:                 ds = re.sub(p, c, ds);
13:             end if
14:         end for
15:     end for
16:     vec = doc.getElementsByTag('vec');
17:     height = vec.getAttribute('height');
18:     width = vec.getAttribute('width');
19:     enc_height = cipher.encrypt(height) − offset;
20:     enc_width = cipher.encrypt(width) − offset;
21:     vec = re.sub(height, enc_height, vec);
22:     vec = re.sub(width, enc_width, vec);
23:     file.write(vec);
24: end procedure
```

of the 32 × 32 size. This allows us to distribute enough the previous vector values, but we have to consider data loss. Note that the position-related numeric values inside the vector graphics file from vectorization are all positive. However, since the OPE-converted values can be negative depending on the OPE key, data loss can happen. Thus, after setting an *offset* value into the encrypted zero-value, we subtract each converted numeric variable from the *offset* value.

### 3) STEP3: RASTERIZATION

In Rasterization step, the position-shifted vector graphics file is converted back to the raster graphics image. We note that the vector image file encrypted using OPE cannot directly be used as an input to CNN models. Also, let us note that if the position-shifted vector file is exposed to an adversary, the private object in the original input image can be exposed because OPE is vulnerable to the chosen plaintext attack [32].

To address such two problems, the rasterization step consists of "resize" to reduce height and width and "rasterize" to convert back to raster graphic image from vector graphic image. The process of "Resize" prevents the encrypted values by OPE from being expose. Also, regardless of the size of the original image, the size of resultant de-identified image can be maintained constant like 224 × 224 and the training time can be kept low. Since the CIFAR-10 data is a three-dimensional array data of 32 × 32 × 3 and the de-identified raster image is also a three-dimensional array data, proposed PPDL approach enables the generated new

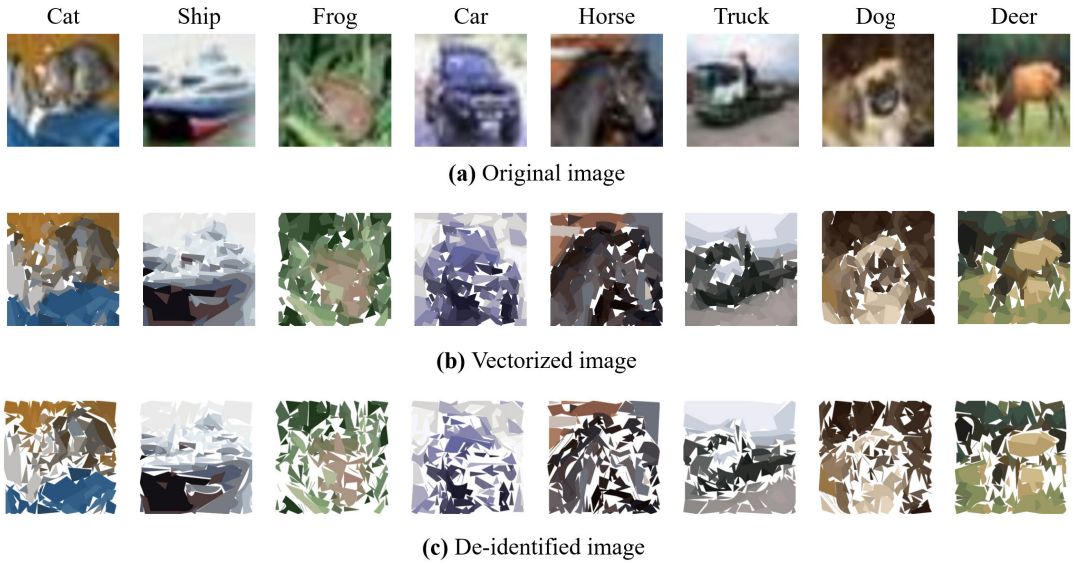
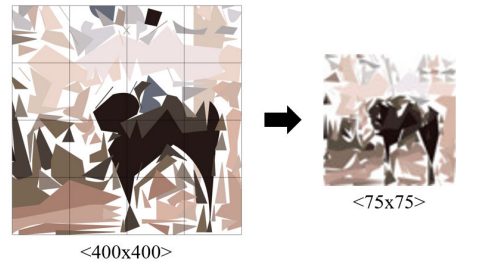
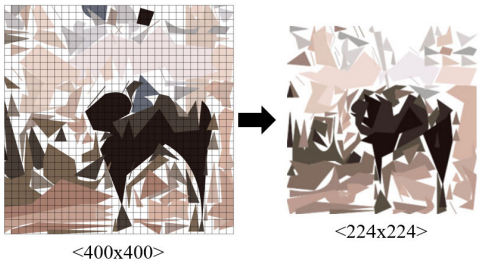



**FIGURE 9.** Structural de-identification procedure of 8 CIFAR-10 images following three steps of the proposed structural image de-identification approach. From step 1, original images in (a) are converted into vector graphics images in (b). From steps 2 and 3, de-identified images in (c) are generated.




**(a)** Rasterization with 400px(Vector) to 75px(Raster)




**(b)** Rasterization with 400px(Vector) to 224px(Raster)

**FIGURE 10.** Rasterization problem. Vector images of the same size were converted to raster images of 75 × 75 and 224 × 224.

image to be trained without changing the existing CNN models. In addition, since the objects in de-identified image have been modified at the same level as the objects in original three-dimensional image, we can apply data augmentation in the cloud server.

The size of one pixel in the raster graphics image depends on pixel density (dpi), but is fixed at 25.4mm in 1dpi. Extremely, while a single object can be represented by multiple pixels in large image, multiple objects in small image can be represented by one pixel. As shown in Fig. 10, depending on the degree of resizing, vector graphics image may not all be represented in raster graphics image. Based on our experimental results, we reduce the size of image by at least 30%.

### C. STRUCTURAL IMAGE DE-IDENTIFICATION EXAMPLE

After randomly selecting eight images with the different class labels from CIFAR-10 dataset, we showed how original raster graphics images changed into de-identified images following the proposed three steps in Fig. 9. While images in Fig. 9 (a) represent the original raster graphics images, images in Figs. 9 (b) and (c) show vector graphics images obtained from the vectorization step and de-identified images obtained from the position OPE step and the rasterization step, respectively.

By comparing the original images in Fig. 9 (a) with the de-identified images in Fig. 9 (c), we observe that the white margin created by OPE lowers the human recognition rate. However, since the overall layouts or colors in object are well kept, CNN model for deep learning can correctly classify each image into one of ten classes.

### D. MEASUREMENT INDICES

As being described in [14], there is a trade-off between the object classification accuracy of deep learning models and the object privacy in the privacy-preserved image. Indeed, since protecting only specific privacy-sensitive objects within the image is more challenging, most existing PPDL methods encrypt the entire image for preserving private information. That is, object classification accuracy decreases after encrypting private objects in the image.

From the observation that human visual system is slightly different from deep learning system, we use the $L_2norm$ index with the structural similarity ($SSIM$) index

**TABLE 2. Structural based de-identification measurement (*SDIM*). Here, each value is the average value measured for 10,000 test images in CIFAR-10.**

| Original image size | (a) $SSIM_{Vec}$ | (b) $SSIM_{Dei}$ | (c) $SSIM_{Pos}$ | (d) $L_2norm_{Dei}$ | (e) $DI$ | (f) $SDIM$ |
|---|---|---|---|---|---|---|
| 32x32 | 0.6673 | 0.3477 | -0.3196 | 0.2949 | 0.32130 | 0.00170 |
| 36x36 | 0.7034 | 0.3589 | -0.3445 | 0.2729 | 0.31590 | -0.02860 |
| 42x42 | 0.7368 | 0.3584 | -0.3784 | 0.2577 | 0.30805 | -0.07035 |
| 46x46 | 0.7545 | 0.3434 | -0.4111 | 0.2527 | 0.29805 | -0.11305 |
| 48x48 | 0.7630 | 0.3388 | -0.4242 | 0.2505 | 0.29465 | -0.12955 |
| 50x50 | 0.7705 | **0.3355** | **-0.4350** | 0.2510 | **0.29325** | **-0.14175** |
| 52x52 | 0.7783 | 0.3484 | -0.4299 | 0.2387 | 0.29355 | -0.13635 |
| 54x54 | 0.7844 | 0.3606 | -0.4238 | 0.2290 | 0.29480 | -0.12900 |
| 58x58 | 0.7943 | 0.3616 | -0.4327 | 0.2253 | 0.29345 | -0.13925 |
| 62x62 | 0.8036 | 0.3947 | -0.4089 | 0.2028 | 0.29875 | -0.11015 |
| 64x64 | 0.8086 | 0.4033 | -0.4053 | 0.1974 | 0.30035 | -0.10495 |
| 75x75 | 0.8245 | 0.4077 | -0.4168 | 0.1821 | 0.29490 | -0.12190 |
| 100x100 | **0.8438** | 0.4823 | -0.3615 | **0.1408** | 0.31155 | -0.04995 |

for estimating the trade-off between the object classification accuracy and the object privacy in the de-identified image. By combing the $L_2norm$ index with the *SSIM* index, we use a performance measurement index, called Structural De-Identification Measurement(*SDIM*). Note that while $L_2norm$ implicitly measures the degree of modification in adversarial images [33], *SSIM* measures a human recognition rate of an object in an image [31]. Also, with $L_2norm$, deep learning model can calculate the dissimilarity between the modified image and the original image. With *SSIM*, we can measure the structural similarity between the original image and the modified image. Thus, *SDIM* is used to measure the trade-off between the object classification accuracy of deep learning models and the object privacy within the input image in the proposed PPDL method.

To obtain the *SDIM* value, let us consider two terms: (1) Structural Similarity of Position OPE($SSIM_{Pos}$); (2) De-identification Indicator($DI$);

### 1) STRUCTURAL SIMILARITY OF POSITION OPE ($SSIM_{Pos}$) INDEX

To measure the performance of the proposed method, it is required to determine how much structural similarity has decreased through position OPE. The structural similarity (*SSIM*) values through position OPE can be measured from the fact that structural image de-identification is a combined procedure of vectorization and position OPE. That is, structural similarity of position OPE ($SSIM_{Pos}$) can be formulated into the following equation:

$$SSIM_{Pos} := SSIM_{Dei} - SSIM_{Vec}. \tag{4}$$

From equation 4, the greater the gap between the *SSIM* value of de-identified image and the vector image is, the better the de-identification through position OPE works. Since the value of *SSIM* has a range from 0 to 1, the value of $SSIM_{Pos}$ closer to $-1$, when $SSIM_{Dei}$ is 0 and $SSIM_{Vec}$ is 1, indicates that human perception system cannot identify the objects in the de-identified image.

In Table. 2, from the original CIFAR-10 image of the $32 \times 32$ size, we observed that the $SSIM_{Vec}$ value is 0.6673 for vectorized images and the $SSIM_{Dei}$ value is 0.3477 for de-identified images. From equation 4, the value of $SSIM_{Pos}$ is $-0.3196$.

In the vectorization step, we note that object vanishing and feature vanishing can be observed. This means that unintended information can be lost in the vectorization step. Bartolini *et al.* introduced that the value of *SSIM* larger than 0.94 indicates that human can clearly recognize objects even though the image is converted into the compressed image [34]. Since the $SSIM_{Vec}$ value of 0.6673 is lower than the least recommended value(: 0.94), the original image of the $32 \times 32$ size is not suitable for de-identification.

To solve the above problem, we perform de-identification while increasing the size of an original CIFAR-10 image from the $32 \times 32$ size. As a result, we measure the values of $SSIM_{Vec}$ for a set of images obtained from vectorization and $SSIM_{Dei}$ for a set of images obtained from de-identification, respectively. From values for $SSIM_{Vec}$ under various size of image in Table. 2(a), we observe that as the size of the original image increases, values of *SSIM* after vectorization approaches to 1. On the other hand, from values for $SSIM_{Dei}$ under various sizes of an image in Table. 2(b), we observe that as the size of the original image exceeds the $50 \times 50$ size, values of $SSIM_{Dei}$ gradually increase. These observations imply that we need to find the optimal point which considers both vectorization and image de-identification. In Table. 2(c), we observe that while $SSIM_{Pos}$ values decrease when the size of image is smaller than $50 \times 50$. We also observe that $SSIM_{Pos}$ values increases when the size of image is larger than $50 \times 50$. From the experimental results, the $50 \times 50$ size is determined into the optimal size for CIFAR-10 images since the value of $SSIM_{Pos}$ was closest to $-1$.

### 2) DE-IDENTIFICATION INDICATOR ($DI$)

Successful privacy-preserving deep learning requires to consider the trade-off between object classification accuracy

and privacy. While deep learning performance can be predicted through the value of $L_2$ distance in general, privacy can be measured through the value of *SSIM*. That is, the smaller the $L_2Norm$ is, the smaller the difference from the original image is. Also, the smaller the *SSIM* value is, the larger the difference from the original though human perception is. In Table. 2(d), we show the normalized values of $L_2norm_{Dei}$ under various sizes of a de-identified image. We can observe that as the size of image increases, the value of $L_2norm_{Dei}$ decreases. In Table. 2(b), we can observe that as the size of the image exceeds the $50 \times 50$ size, the value of $SSIM_{Dei}$ increases.

Since the smaller *SSIM* value and $L_2norm$ distance value are the better indicators in de-identification, we determine the sum of both values into an indicator of the trade-off between classification accuracy and privacy, i.e., De-identification Indicator(DI):

$$DI(\alpha) := SSIM_{Dei} * \alpha + L_2norm_{Dei} * (1 - \alpha), \tag{5}$$

where $\alpha (0 \leq \alpha \leq 1)$ is a ratio which represents the trade-off coefficient between privacy and performance. At the column (e) in Table. 2, we show values of *DI* under various sizes of an image when $\alpha = 0.5$. Here, the value $\alpha$ set into 0.5 means that privacy and performance weights are equal. From the lowest *DI* value, we also observe that the $50 \times 50$ size is the optimal size of the image for de-identification.

#### 3) STRUCTURAL DE-IDENTIFICATION MEASUREMENT (*SDIM*) INDEX

While $SSIM_{pos}$ shows how de-identification influences on the performance of the proposed PPDL approach, *DI* shows the trade-off between object classification accuracy and privacy. To consider the trade-off between object classification accuracy and privacy under image de-identification, we define a new metric, called Structural based De-Identification Measurement (SDIM) as follows:

$$SDIM(\alpha) := SSIM_{Pos} + DI(\alpha), \tag{6}$$

where as the value of *SDIM* decreases, the performance of the proposed PPDL increases in practice. At the column (f) in Table. 2, we show how the values of *SDIM* change under various sizes of an image. We observe that the value of *SDIM* is minimized at the $50 \times 50$ size of image. This indicates that the $50 \times 50$ size of image in CIFAR-10 is the optimal input image for structural image de-identification when considering the trade-off between object classification accuracy and privacy under image de-identification.

In order to employ an user's image data as input to the proposed PPDL method, it is recommended to choose the optimal image size to protect the privacy of the object within image while maintaining the performance of object classification. A image size with the smallest value of measuring the *SDIM* value by resizing the user's image data to various sizes becomes the optimal image size. On CIFAR-10 used in this paper, likewise the $50 \times 50$ with the smallest *SDIM* value is the optimal size, the optimum size for user's image data can be predicted through *SDIM* values without training the DNN model.

### E. SECURITY ANALYSIS

In this section, we show security analysis results of the proposed method. Under the ciphertext-only attack and the order-revealing attacks on OPE including chosen plaintext attack (CPA) [35] and the Leakage-Abuse attack [36], we analyze the security strength of the proposed method.

#### 1) CIPHERTEXT-ONLY ATTACK TOLERANCE

Let us assume that an adversary acquires a de-identified image without authority during transmission to cloud computing server or within a cloud system. Also, let us consider that the brute-force attack is used as an attack type of ciphertext-only attack.

If an image with $X \times Y$ pixels is divided into pixels, the number of pixels, $n$, is given by

$$n = X \times Y. \tag{7}$$

Unlike negative-positive transformation and color shuffling method [17] that is the most recent pixel-value-based image encryption method, the proposed method shifts only the position of the pixel in the vectored graphics image. Because color values do not change, the color value of each pixel in de-identified image is mapped to one of color values in the original image.

Note that after applying the Position OPE into each pixel, the pixel values can also be changed into white color in a white background. Thus, if we assume that all $n$ pixels have different colors, the pixel value of the generated de-identified image can be mapped to one of $n + 1$ colors. That is, the featured spaces of the proposed method can be represented into:

$$N(n) = (n + 1)^n. \tag{8}$$

On the other hand, in negative-positive transformation and color shuffling method, each pixel value is replaced into the original value $p$ or $p \oplus 255$ value with a $P(r) = 0.5$ probability. Also, the order of RGB is shuffled with a $P(r) = 0.167$ probability. That is, the featured spaces of negative-positive transformation and color shuffling method [17] are given into:

$$N_{NP}(n) = 2^{3n} \times 6^n = 8^n \times 6^n. \tag{9}$$

Since $N(n) > 64^n \gg N_{NP}(n)$, where $n > 64$, the proposed method has the larger featured space than the latest pixel-value-based method. That is, when the image size is larger than $8 \times 8$, security of the proposed method is much stronger than the latest pixel-value-based method against the brute-force attack.

#### 2) ORDER-PRESERVING ENCRYPTION TOLERANCE

OPE is known to be vulnerable to CPA attacks [35] and Leakage-Abuse attacks [36] because there is a possibility of deducing the original values based on the sequence information between the original data and the encrypted one.

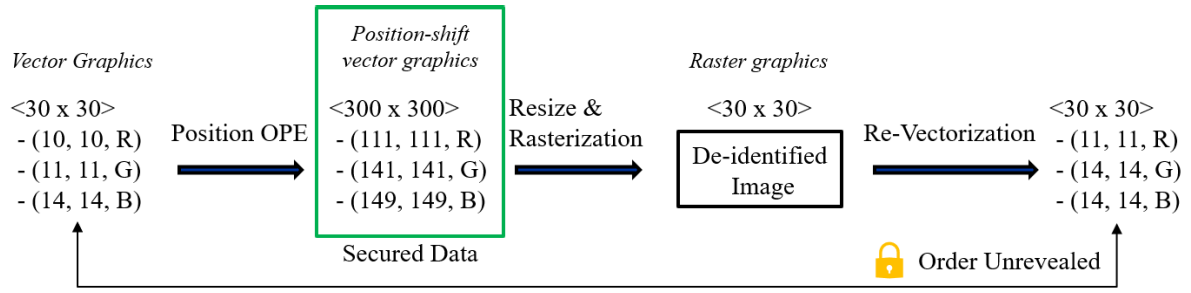



**FIGURE 11.** Simple example of structural image de-identification approach. Here, values between the position values of original vector graphics and the position values generated by re-vectorization are dissimilar. Also, the sequence information encrypted by OPE in position-shift vector graphics is not directly exposed.

From equation 1, the relationship between vectorization and rasterization can be expressed into:

$$Rasterize(Vectorize(PI)) := PI, \tag{10}$$

where as $PI$ is Plain Image.

In the proposed method, position OPE and resize operations are located between vectorization and rasterization in equation 10, Thus, de-identified image cannot to be restored into plain original image. For example, in Fig. 11, the position-shift vector graphics image size, 300, is known only to OPE users. Thus, the proposed method cannot be performed in reverse because the target size value, 300, is unrevealed. Even though we know the OPE key, we cannot restore the original image from de-identified image due to the resize operation in the rasterization. Since the values between the original vector graphics image and the images generated by re-vectorization are dissimilar as shown in Fig. 11, the proposed method is not vulnerable to the leakage-abuse attack. Also, as shown in Fig. 11, the proposed method does not use the encrypted value using OPE and does not expose the sequence of encrypted data. Thus, the proposed method is not vulnerable to the chosen plaintext attack.

## V. EVALUATION RESULTS

To show how effective the proposed structural image de-identification approach is, we measured the object classification accuracy using CIFAR-10 [18] dataset which is a major dataset used as benchmarks in the deep-learning literature. The CIFAR-10 dataset consists of 60,000 32 × 32 colour images in 10 classes, which are 50,000 training images and 10,000 test images and each of which has a single object in. Specifically, we evaluated the performance of the proposed approach by answering the following questions for de-identification:

- How does the various sizes of an input images being vectorized or de-identified influence on the performance of the CNN model?
- How does the size of the final de-identified image influence on the object classification accuracy of the CNN model?
- Is the proposed structural image de-identification approach adequate to process gray-scale images?
- Does the proposed structural image de-identification approach show the good-enough object classification accuracy compared with existing method?
- Does the proposed structural image de-identification approach show the good-enough object classification accuracy under various datasets?

### A. EXPERIMENTAL ENVIRONMENT

To measure the performance of the proposed structural image de-identification approach, we used the GPU server, which consists of Intel(R) Xeon(R) CPU E5-2630v3 @2.40GHz with 8 cores, 62 GB RAM and an NVIDIA(R) GeForce RTX 2080 Ti.

As a set of input images, we used CIFAR-10 [18], which is a standard dataset to evaluate the performance of the deep learning approaches. Since the proposed structural image de-identification approach can work with any DNN models, we used the Inception ResNet V2 model [3], which adds the advantage of the ResNet model using the skip connection [37] to the Inception V3 model which has the various size of multi convolution layer [38]. This state-of-the-art DNN model was trained using a RMSprop algorithm, where the learning rate was initially set to 0.001 for 250 epochs. Also, the learning rate was set to 0.0005 for 100 epochs, 0.0003 for 125 epoches and 0.0001 for 200 epochs. We used a weight decay of $1e-6$, and a batch size of 64. We also used data augmentation(rotation, shifting and flip) for preprocessing the input images to the DNN model.

To implement our method, we used *imagetracerjs* [39] in the vectorization step, *pyope* [40] in the position OPE step and *ImageMagick* [41] in the rasterization step. Specifically, in the vectorization step, we used SVG vector file format with the options 'default'. In the position OPE step, we set the OPE key to "jX4ZXXmM3qTgAjez/1j/EVz5tCcmQ711bz8hqIMJap0=". OPE output values were bounded between −99999 and 99999. Also, the offset value is set to −796 to prevent loss of data. In the rasterization step, we used '-density = 288' and '-resize = 224 × 224!' options to generate de-identified images.

In general, when evaluating the performance of deep learning models, different performance metrics such as accuracy,

precision, recall, and F1-score[1] can be used. We noted that since images in CIFAR-10 are uniformly distributed for each class, accuracy is better than F1-score [42] when evaluating the deep learning model. Thus, the object classification accuracy was used as a metric for images in CIFAR-10. Also, since images in ImageNet are non-uniformly distributed(imbalanced) for each class, we measured F1-score for evaluating the deep learning model and accuracy for comparison with the analysis results for CIFAR-10.

### B. EXPERIMENTAL RESULTS

In this section, we show the experimental results to answer the above five questions. Also, we compare the object classification accuracy of the proposed approach with a recent pixel-based-encryption approach.

#### 1) HOW DOES THE VARIOUS SIZES OF AN INPUT IMAGE BEING VECTORIZED OR DE-IDENTIFIED INFLUENCE ON THE PERFORMANCE OF THE CNN MODEL?

To observe the influence of the various size of input images being vectorized or de-identified on the performance of the CNN model, we measured the values of *SDIM* and the object classification accuracy of the proposed approach for four different sizes: $32 \times 32$, $42 \times 42$, $50 \times 50$, and $64 \times 64$.

**TABLE 3. Image classification accuracy under various sizes of an image.**

| Original image size | $SDIM$ | Accuracy(%) | |
|---|---|---|---|
| | | Vectorized image | De-identified image |
| 32x32 | 0.0017 | 82.17 | 81.26 |
| 42x42 | -0.0703 | 85.55 | 85.88 |
| 50x50 | -0.1417 | 86.76 | 87.10 |
| 64x64 | -0.1049 | 87.75 | 87.48 |

Table. 3 show our measurement results. First, we observed that the larger the size of the original raster graphics image is, the more accurate the classification of the vector graphics image is. This is because the size of the input image to the CNN model increases, vanishing problems in the vectorization step are being solved. Second, we observed that the object classification accuracy for de-identified images was almost the same as the object classification accuracy of vector graphics image. This observation implies that the vanishing problems is important. Third, we observed that as the size of the original image increases, the object classification accuracy of the CNN model under the various sizes of an input image being vectorized or de-identified increases. Fourth, we observed that when the size of input images are set to $50 \times 50$, the value of *SDIM* was lowest while the highest classification accuracies of being vectorized or de-identified images were observed at the $64 \times 64$ size of input images. This is because as the size of the input image to the CNN model increases, it becomes easy for human to identify the input image. This observation implies that in the context of the trade-off between object classification accuracy and privacy, we need to adjust the weight between classification accuracy and privacy according to the deep learning applications.

#### 2) HOW DOES THE SIZE OF THE FINAL DE-IDENTIFICATION IMAGE INFLUENCE ON THE OBJECT CLASSIFICATION ACCURACY OF THE CNN MODEL?

To evaluate the performance of the proposed structural de-identification approach under various sizes of de-identified images, we generated multiple datasets with the different sizes from an image with the $50 \times 50$ size and then, measured the performance of the proposed PPDL approach. As shown in Table. 4, the object classification accuracy of the CNN model gradually increased as the size of de-identified image increases. This is because if the size of de-identified image is too small, the objects in the image might be represented using small number of pixels. In other words, even though decreasing the size of de-identified image is good for avoiding the exposure of the OPE-encrypted data while reducing the evaluation time, the object classification accuracy becomes worse.

**TABLE 4. Classification accuracy under various sizes of a rasterization image whose original size is 50 × 50.**

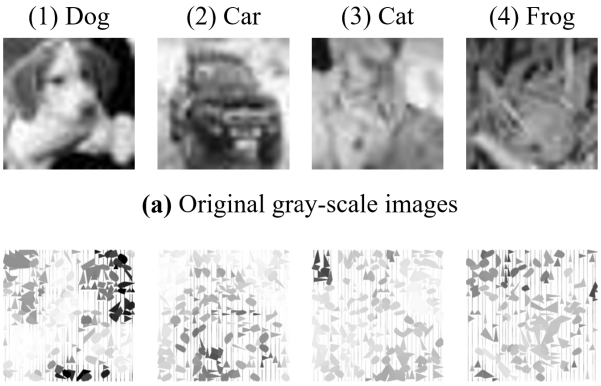
| De-identified image size | Accuracy(%) | Evaluation time(ms) (per image) |
|---|---|---|
| 75x75 | 79.16 | 0.4000 |
| 100x100 | 81.29 | 0.6000 |
| 150x150 | 85.32 | 1.2001 |
| 200x200 | 85.77 | 2.1014 |
| 224x224 | 87.10 | 3.0003 |

#### 3) IS THE PROPOSED STRUCTURAL IMAGE DE-IDENTIFICATION APPROACH ADEQUATE TO PROCESS GRAY-SCALE IMAGES?

To show that the proposed structural image de-identification approach is even adequate to analyze gray-scale images, we measured the performance of the proposed structural image de-identification approach after converting the CIFAR-10 images into gray-scale images. For examples, we showed gray-scale images converted from CIFAR-10 color images and the corresponding de-identified gray-scale images in Figs. 12 (a) and (b), respectively.

From Table. 5, we compared the values of $SSIM_{Pos}$ and *SDIM* for color images with those for the converted gray-scale images with the same size. As a result, we found that the value of $SSIM_{Vec}$ was close to 1 and then, the value of $SSIM_{Dei}$ was close to 0. Also, from the result that the $SSIM_{Pos}$ value was close to $-1$, we found that gray-scale images and color

[1]Accuracy represents the ratio of the number of correct predictions over the total number of predictions made. Precision represents the ratio of the number of samples that belong to the actual positive class over the number of samples predicted to be positive, and recall represents the ratio of the number of samples predicted to be positive over the actual number of positive class samples. F1-score is determined into the harmonic mean of the precision and recall.




**(a)** Original gray-scale images

**(b)** De-identified gray-scale images

**FIGURE 12.** Examples of de-identified gray-scale image.

**TABLE 5.** Comparison under different color channels on an image: RGB color image *v.s.* Gray-scale image.

| Index | RGB color image | Gray-scale image |
|---|---|---|
| $SSIM_{Vec}$ | 0.7705 | **0.8112** |
| $SSIM_{Dei}$ | 0.3355 | **0.2363** |
| $SSIM_{Pos}$ | -0.4350 | **-0.5749** |
| $SDIM$ | -0.1417 | **-0.2452** |

images showed the same characteristics when using the proposed PPDL. Also, the value of *SDIM* of a gray-scale image was close to $-1$ than the RGB color image. This observation verifies that since the proposed structural de-identification approach transforms only the structural features of image, it can effectively process images with one channel such as gray-scale image as well as images with multiple channels such as RGB color image.

The CNN model in experimental environment described in subsection V(A) achieved an accuracy of 83.56%. Unlike the pixel-value-based encryption [16], [17] approach which uses pixel values of three RGB channels, we confirmed that our approach works well for gray-scale image just as color image through the accuracy and *SDIM* values.

#### 4) DOES THE PROPOSED STRUCTURAL IMAGE DE-IDENTIFICATION APPROACH SHOW THE GOOD-ENOUGH OBJECT CLASSIFICATION ACCURACY COMPARED WITH EXISTING METHOD?

To show how the proposed structural image de-identification approach predicts the class labels even by using encrypted images, we compared the object classification accuracy of the proposed approach with those of the CNN model using plain images, which are the non-encrypted images. Note that among the most recent PPDL approaches, the pixel-value-based encryption approach obtained good performance in the most recent deep learning models. Thus, we also compared the proposed structural image de-identification approach with the state-of-the-art pixel-value-based encryption approach, i.e., the negative-positive and color shuffling method [17]. We implemented the negative-positive and color shuffling method and produced dataset by stretching the original image into the $224 \times 224$ size [17] when training the CNN model described in section V(A).

**TABLE 6.** Accuracy and brute-force attack tolerance comparison of different encryption-based PPDL methods under CIFAR-10 dataset.

| Method | Accuracy(%) | Brute-force attack tolerance $(n = \#\ of\ pixels)$ |
|---|---|---|
| Using plain image | 91.68 | - |
| Negative-Positive and Color Shuffling method [17] | 82.16 | $(8)^n \times (6)^n$ |
| Proposed method | 87.10 | $(n+1)^n$ |

From the experimental results in Table. 6, the proposed structural image de-identification approach showed the object classification accuracy of 87.10%, while the negative-positive and color shuffling method [17] showed the object classification accuracy of 82.16%. That is, the proposed de-identification approach showed the better classification accuracy than the negative-positive and color shuffling method. In addition, where $n > 64$, the proposed method has the larger featured space than the negative-positive and color shuffling method. That is, when the image size is larger than $8 \times 8$, security of the proposed method is much stronger than the negative-positive and color shuffling method against the brute-force attack.

#### 5) DOES THE PROPOSED STRUCTURAL IMAGE DE-IDENTIFICATION APPROACH SHOW THE GOOD-ENOUGH OBJECT CLASSIFICATION ACCURACY UNDER VARIOUS DATASETS?

To show that the proposed structural image de-identification approach presents the good-enough object classification accuracy under various datasets, we measured the performance with ImageNet [19]. As a well-known standard dataset for image classification, ImageNet contains 14,197,122 images with 20,000 classes. We selected ImageNet because different from CIFAR-10, where image objects are uniformly distributed for each class, ImageNet consists of images whose objects are non-uniformly distributed (imbalanced) for each class. For performance comparison with CIFAR-10, we randomly selected 68,904 images, which consist of 58,492 images for training and 10,412 images for test, with 139 classes.

Since ImageNet has the various sizes of images, we resized each image into $32 \times 32$, which is the size of image in CIFAR-10, to make an input image for deep learning. Note that we focus on measuring the performance of the proposed structural image de-identification approach under various datasets. After using the ResNet50 model with Adam Optimizer for simplicity, we measured the object classification

accuracy for the plain image and the de-identified image with the same parameter values in Section V(A).

From the experimental results for ImageNet, we observed that the object classification accuracy and F1-score for the de-identified image were less than those for the plain image by as much as 6.39% and 0.06, respectively. However, since *SDIM* of the de-identified image was -0.4313, the experimental results showed that human perception system could not identify the objects in the de-identified image due to the increase of privacy. From the experimental results for ImageNet and the others for CIFAR-10, we also observed that the proposed structural image de-identification approach shows the good object classification accuracy even though any image dataset with uniform or non-uniform distribution of images for each class is used for learning.

## VI. CONCLUSION

Due to the risk of data leakage while training deep learning models involving an enormous amount of data including sensitive information, general deep learning approach can expose the private information in training data. To address such privacy concerns, we proposed a new PPDL method, called structural image de-identification approach, which trains encrypted data itself without decryption. Based on the intuition that the human visual system is sensitive to structural change, the proposed structural image de-identification approach converts input images into vectors for modifying only the structural parts of the original one. Different from FHE-based methods whose usage is limited due to difficulty in applying the state-of-the-art DNN models directly and so on, the proposed structural image de-identification approach can employ any DNN models using encrypted data. Different from pixel-value-based encryption methods which cannot analyze the gray-scale images directly, the proposed structural image de-identification approach showed the good performance even for the gray-scale images. From the evaluation results using CIFAR-10, we showed that the object classification accuracy of the proposed structural image de-identification approach was higher than the other encryption-based PPDL approaches. From the evaluation results using CIFAR-10 and ImageNet with the different distributions of images for each class, we also showed that the object classification accuracy of the proposed structural image de-identification approach was the same as the performance of CNN models for the non-encrypted image.

## REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[4] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*. [Online]. Available: http://arxiv.org/abs/1712.04621

[5] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2113–2122.

[6] G. Lacey, G. W. Taylor, and S. Areibi, "Deep learning on FPGAs: Past, present, and future," 2016, *arXiv:1602.04283*. [Online]. Available: http://arxiv.org/abs/1602.04283

[7] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 1–5.

[8] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen, "Multi-key privacy-preserving deep learning in cloud computing," *Future Gener. Comput. Syst.*, vol. 74, pp. 76–85, Sep. 2017.

[9] M.-P. Hosseini, H. Soltanian-Zadeh, K. Elisevich, and D. Pompili, "Cloud-based deep learning of big EEG data for epileptic seizure prediction," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2016, pp. 1151–1155.

[10] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "IPrivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1005–1016, May 2017.

[11] A. K. Tonge and C. Caragea, "Image privacy prediction using deep features," in *Proc. 13th AAAI Conf Artif. Intell.*, 2016, pp. 4266–4267.

[12] *Oasis Labs' Dawn Song on a Safer Way to Protect Your Data*. Accessed: Jan. 3, 2020. [Online]. Available: https://www.wired.com/story/dawn-song-oasis-labs-data-privacy-wired25/

[13] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 201–210.

[14] 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.

[15] A. Al Badawi, J. Chao, J. Lin, C. F. Mun, J. J. Sim, B. H. M. Tan, X. Nan, K. M. M. Aung, and V. R. Chandrasekhar, "The AlexNet moment for homomorphic encryption: HCNN, the first homomorphic CNN on encrypted data with GPUs," 2018, *arXiv:1811.00778*. [Online]. Available: http://arxiv.org/abs/1811.00778

[16] M. Tanaka, "Learnable image encryption," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2018, pp. 1–2.

[17] W. Sirichotedumrong, T. Maekawa, Y. Kinoshita, and H. Kiya, "Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 674–678.

[18] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Proc. Citeseer*, 2009, p. 7.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[20] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2016, pp. 25–32.

[21] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," 2015, *arXiv:1511.05122*. [Online]. Available: http://arxiv.org/abs/1511.05122

[22] N. Chapman and J. Chapman, *Digital Multimedia*. Hoboken, NJ, USA: Wiley, 2009.

[23] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2004, pp. 563–574.

[24] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[25] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 1309–1316.

[26] M. Abadi, A. Chu, I. Goodfellow, H. B. Mcmahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.

[27] Y.-T. Tsou, H.-L. Chen, and Y.-H. Chang, "RoD: Evaluating the risk of data disclosure using noise estimation for differential privacy," *IEEE Trans. Big Data*, early access, May 14, 2019, doi: 10.1109/TBDATA.2019.2916108.

[28] F. Bourse, M. Minelli, M. Minihold, and P. Paillier, "Fast homomorphic evaluation of deep discretized neural networks," in *Proc. Annu. Int. Cryptol. Conf.* Springer, 2018, pp. 483–512.

[29] X. Jiang, M. Kim, K. Lauter, and Y. Song, "Secure outsourced matrix computation and application to neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Jan. 2018, pp. 1209–1222. [Online]. Available: https://dl.acm.org/doi/proceedings/10.1145/3243734?tocHeading=heading7
[30] (Mar. 2017). *Types of Bitmaps*. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/types-of-bitmaps
[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[32] A. Boldyreva, N. Chenette, and A. O'Neill, "Order-preserving encryption revisited: Improved security analysis and alternative solutions," in *Proc. Int. Assoc. Cryptologic Res. (IACR)*. Springer, 2011, pp. 578–595. [Online]. Available: https://link.springer.com/book/10.1007/978-3-642-22792-9
[33] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: http://arxiv.org/abs/1412.6572
[34] A. Bartolini, M. Ruggiero, and L. Benini, "Visual quality analysis for dynamic backlight scaling in LCD systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib.* Nice, France: European Design and Automation Association, Apr. 2009, pp. 1428–1433.
[35] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in *Proc. Int. Assoc. Cryptologic Res. (IACR)*. Springer, 2009, pp. 224–241. [Online]. Available: https://link.springer.com/book/10.1007/978-3-642-01001-9
[36] P. Grubbs, K. Sekniqi, V. Bindschaedler, M. Naveed, and T. Ristenpart, "Leakage-abuse attacks against order-revealing encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 655–672.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[39] *Jankovicsandras, Imagetracerjs*. Accessed: Feb. 15, 2019. [Online]. Available: https://github.com/jankovicsandras/imagetracerjs
[40] *Tonyo, Pyope*. Accessed: Feb. 22, 2019. [Online]. Available: https://github.com/tonyo/pyope/
[41] ImageMagick Studio LLC. *The Official Imagemagick Blog*. Accessed: Feb. 28, 2019. [Online]. Available: https://imagemagick.org/index.php
[42] P. Huilgol. *Accuracy vs. F1-Score*. Accessed: Dec. 9, 2019. [Online]. Available: https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2

**DONG-HYUN KO** received the B.E. degree from Pusan National University, Busan, South Korea, in 2018, where he is currently pursuing the M.E. degree in computer science and engineering. His research interests include privacy, blockchain, network security, and software security.

**SEOK-HWAN CHOI** received the B.E. degree from Pusan National University, Busan, South Korea, in 2016, where he is currently pursuing the Ph.D. degree in computer science and engineering. His research interests include intrusion detection, network security, and adversarial deep learning.

**JIN-MYEONG SHIN** received the B.E. degree from Pusan National University, Busan, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in computer science and engineering. His research interests include network security, IDS, the IoT security, and fully homomorphic encryption.

**PENG LIU** (Member, IEEE) received the B.S. and M.S. degrees from the University of Science and Technology of China, and the Ph.D. degree from George Mason University, in 1999. He is currently a Professor of information sciences and technology, the Founding Director of the Center for Cyber-Security, Information Privacy, and Trust, and the Founding Director of the Cyber Security Lab, Penn State University. His research interests include the areas of computer and network security. He has published a monograph and more than 260 refereed technical articles. His research has been sponsored by the U.S. National Science Foundation, ARO, AFOSR, DARPA, DHS, DOE, AFRL, NSA, TTC, CISCO, and HP. He has served on more than 100 program committees and reviewed articles for numerous journals. He received the DOE Early Career Principle Investigator Award. He has co-led the effort to make Penn State an NSA-certified National Center of Excellence in Information Assurance Education and Research. He has advised or co-advised more than 30 Ph.D. dissertations to completion.

**YOON-HO CHOI** (Member, IEEE) received the M.S. and Ph.D. degrees from the School of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea, in 2004 and 2008, respectively. He was a Postdoctoral Scholar at Seoul National University, in 2008, and at Pennsylvania State University, University Park, PA, USA, from 2009 to 2009. From 2010 to 2012, he was a Senior Engineer with Samsung Electronics. From 2012 to 2014, he was an Assistant Professor with the Department of Convergence Security, Kyonggi University, Suwon, South Korea. He is currently an Associate Professor at the School of Computer Science and Engineering, Pusan National University, Busan, South Korea. His research interests include privacy-preserving deep learning, adversarial examples, anomaly detection, deep packet inspection for high-speed intrusion prevention, and the IoT security for realizing secure computer systems and networks. He has served as a TPC member and an editor in various international conferences and journals.