IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 7, 2019, accepted December 7, 2019, date of publication December 12, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959017

Pixel-Based Image Encryption Without Key Management for Privacy-Preserving Deep Neural Networks

WARIT SIRICHOTEDUMRONG, (Student Member, IEEE), YUMA KINOSHITA, (Student Member, IEEE), AND HITOSHI KIYA, (Fellow, IEEE)
Department of Computer Science, Tokyo Metropolitan University, Tokyo 191-0065, Japan

Corresponding author: Hitoshi Kiya (kiya@tmu.ac.jp)

This work was supported in part by the Grant-in-Aid for Scientific Research (B) through the Japan Society for the Promotion Science under Grant 17H03267.

ABSTRACT We present a novel privacy-preserving scheme for deep neural networks (DNNs) that enables us not to only apply images without visual information to DNNs but to also consider the use of independent encryption keys for both training and testing images for the first time. In this paper, a novel pixel-based image encryption method that maintains important features of original images is proposed for privacy-preserving DNNs. For training, a DNN model is trained with images encrypted by using the proposed method with independent encryption keys. For testing, the model enables us to apply both encrypted images and plain images for image classification. Therefore, there is no need to manage keys. In addition, the proposed method allows us to perform data augmentation in the encrypted domain. In an experiment, the proposed method is applied to well-known networks, that is, deep residual networks and densely connected convolutional networks, for image classification. The experimental results demonstrate that the proposed method, under the use of independent encryption keys, can maintain a high classification performance, and it is robust against ciphertext-only attacks (COAs). Moreover, the results confirm that the proposed scheme is able to classify plain images as well as encrypted images, even when data augmentation is carried out in the encrypted domain.

INDEX TERMS Deep learning, deep neural network, image encryption, privacy-preserving.

I. INTRODUCTION

The spread of deep neural networks (DNNs) has greatly contributed to solving complex tasks for many applications [1]–[3], such as for computer vision, biomedical systems, and information technology. Deep learning utilizes a large amount of data to extract representations of relevant features, so the performance is significantly improved [4], [5]. However, DNNs have been deployed in security-critical applications, such as facial recognition, biometric authentication, and medical image analysis.

Recently, it is very popular for data owners to utilize cloud servers to compute and process a large amount of data instead of using local servers. This is because cloud environment provides the flexibility and cost saving computation. However, since cloud servers are semi-trusted, data privacy, such as personal information and medical records, may be revealed in cloud computing. Therefore, it is necessary to protect data privacy in the cloud environment, and privacy-preserving DNNs have become an urgent challenge. In this paper, we focus on protecting data privacy by encrypting data before uploading to the cloud environment.

Various perceptual encryption methods have been proposed that generate images without visual information [6]–[20], although information theory-based encryption (like RSA and AES) generates a ciphertext. In contrast to information theory-based encryption, images encrypted by the perceptual encryption methods can be directly applied to some image processing algorithms.

Even though some perceptual encryption methods [11]–[15] can be applied to traditional machine learning (ML) algorithms, such as support vector machine (SVM), k-nearest

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

neighbors (KNN), and random forest, even under the use of the kernel trick [17], these methods have never been applied to DNNs. There are two methods [18], [19] that use encrypted images for both training and testing DNN models. One is the first perceptual encryption for privacy-preserving DNNs, that is, Tanaka's scheme [18], which applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to DNNs. The other is a pixel-based encryption method that was proposed to directly apply encrypted images to DNNs [19].

However, Tanaka's scheme [18] cannot avoid the influence of data augmentation in the encrypted domain. Although the pixel-based encryption method [19] carries out data augmentation in the encrypted domain, training and testing images are encrypted by using only one common security key, so it is necessary to safely manage keys, and the method is not very robust against ciphertext-only attacks (COAs).

Thus, in this paper, we propose a novel privacy-preserving method for DNNs that enables us to not only apply images without visual information to DNNs for both training and testing but to also consider the use of independent encryption keys, which means that all images are encrypted by using different security keys, for the first time. In addition, the proposed method allows us to carry out data augmentation in the encrypted domain without any performance degradation. Moreover, it makes it possible for clients to classify plain images even though DNNs are trained by encrypted images.

In an experiment, we compare the proposed method with conventional perceptual encryption-based methods. The experimental results illustrate that the proposed method outperforms the conventional methods in terms of classification accuracy. In addition, images encrypted by the proposed method with independent keys are robust against COAs, namely, the reconstructed images have almost no visual information.

The rest of the paper is organized as follows. Section II is a review of related work. Section III presents the novel perceptual image encryption method for DNNs, and its robustness against COAs is discussed. The classification performance and robustness are evaluated in Section IV. Concluding remarks are in Section V.

II. RELATED WORK

A. VISUAL INFORMATION PROTECTION

Security mostly refers to protection from adversarial forces. This paper focuses on protecting visual information that allows us to identify an individual, the time, and the location of the taken photograph. Semi-trusted cloud providers and unauthorized users are assumed to be adversaries.

A lot of perceptual encryption methods have been proposed for protecting the visual information of images [6]–[20]. Perceptual image encryption generates visually protected images, which have pixel values, like Fig. 1, but information theory-based encryption (like RSA and AES) generates a ciphertext. Therefore, the encrypted images can be directly applied to some image processing algorithms.

For example, encryption methods [6], [7] have been proposed not only for visually protecting privacy and security but also for matching and searching images in the encrypted domain.

Compressible encryption methods have been also proposed that consider both security and efficient compression so that they can be adapted to cloud storage and network sharing [8]–[16]. Some of them [11]–[15] can be applied to traditional ML algorithms, such as support vector machine (SVM), k-nearest neighbors (KNN), and random forest, even under the use of the kernel trick [17]. However, these methods have never been applied to DNNs.

There are two encryption methods [18], [19] that use encrypted images for both training and testing DNN models. One, the first perceptual encryption for privacy-preserving DNNs, is Tanaka's scheme [18]. Tanaka's scheme applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to DNNs. The other is a pixel-based encryption method that directly applies encrypted images to DNNs [19].

However, Tanaka's scheme cannot avoid the influence of data augmentation in the encrypted domain, and an encrypted image has some visual information from the original image, as shown in Fig. 1(c), compared with the conventional pixel-based image encryption [20] in Fig. 1(d). Accordingly, the pixel-based encryption method [19] carries out data augmentation in the encrypted domain. However, training and testing images are encrypted by using only one common security key, so it is necessary to safely manage the keys, and the pixel-based method is not very robust against COAs.

In this paper, we propose a novel perceptual image encryption for privacy-preserving DNNs that overcomes these issues that the conventional methods have.

B. SECURITY PROBLEMS WITH MACHINE LEARNING

The security issues with ML are classified into three classes in terms of the goals of an attack [21]–[28]: reliability of results, model protection, and data protection. For reliability of results, some adversaries aim to confuse users by misclassifying the results of ML, such as imperceptible adversarial perturbation, called ''adversarial examples'' [26]–[28]. Adversarial examples cause DNNs misclassify with high confidence or force them to classify a targeted label. Although various methods have been proposed to defend against adversarial examples [29]–[31], there is no robust model yet that has the same classification accuracy as the conventional models.

Model protection means to protect model parameters including hyper-ones and learned-ones [21], [22]. Model extraction attacks [21], [22], which aim to extract an equivalent or near-equivalent ML model, are vulnerable to ML models.

Alternatively, data protection means to avoid a threat to training and testing data from adversaries. Namely, the data

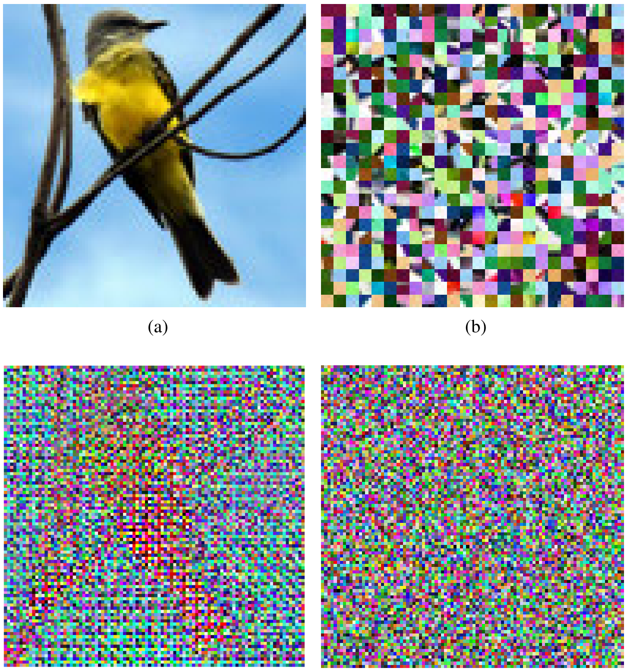

(a) (b) (c) (d)

FIGURE 1. Examples of images. (a) Original image ($U \times V = 96 \times 96$). (b) Image encrypted by block-based encryption [11], [12] (Block size $= 4 \times 4$). (c) Image encrypted by block-based encryption [18] (Block size $= 4 \times 4$). (d) Image encrypted by pixel-based image encryption [20].

that contains confidential information, such as personal information and medical records, has to be protected. For data protection, there are three issues that have to be considered. One is membership inference [23], which has been proposed to identify whether a data record was trained by a model. The second is model inversion attacks [24], [25] that aim to obtain the trained data from the trained model. The other is untrusted cloud environments which are also vulnerable for data privacy because cloud servers are generally semi-trusted. Thus, data privacy may be revealed during computation carried out in the cloud server.

Various encryption methods, such as homomorphic encryption (HE) [32]–[40] and perceptual encryption [8]–[19], have been proposed not only to protect privacy of data but also to be available for cloud computing.

In this paper, we focus on data protection for training and testing DNN models. Moreover, even if membership inference attacks can identify whether or not an image encrypted by perceptual encryption methods was utilized to train the model because the proposed method protects the visual information of images, as described in Section II-A, an individual, the time, and location of a taken photograph are not exposed. Hence, protecting visual information allows us to provide robustness against membership inference attacks and model inversion attacks.

C. PRIVACY-PRESERVING DNNs

Privacy-preserving machine learning methods with homomorphic encryption (HE) [32], [36]–[40] have been studied. One is CryptoNet [39], which can apply HE to the influence stage of DNNs. CryptoNet has very high computational complexity, so a dedicated low computer convolution core architecture for CryptoNet was proposed and implemented with CMOS technology [40]. In CryptoNet, all activation functions and the loss function must be polynomial functions. Therefore, it cannot be applied to state-of-the-art DNNs. Moreover, CryptoNet assumes that the weights in a neural network have been trained beforehand; therefore, CryptoNet is not robust against model inversion attacks [24], [25].

In comparison, an approach with HE was proposed for privacy-preserving weight transmission for multiple owners who wish to apply a machine learning method over combined data sets [32], [36]–[38]. In this approach, since the gradients are encrypted by using HE, model information is not leaked. Privacy-preserving weight transmission can provide robustness against model extraction attacks. However, this approach cannot be applied to network training in the encrypted domain.

Alternatively, as described in Section II-A, two types of perceptual image encryption [18], [19] have applied to privacy-preserving DNNs for image classification, but there are several issues that need to be overcome. In this paper, we aim to directly apply images encrypted by novel perceptual encryption for training and testing DNN models.

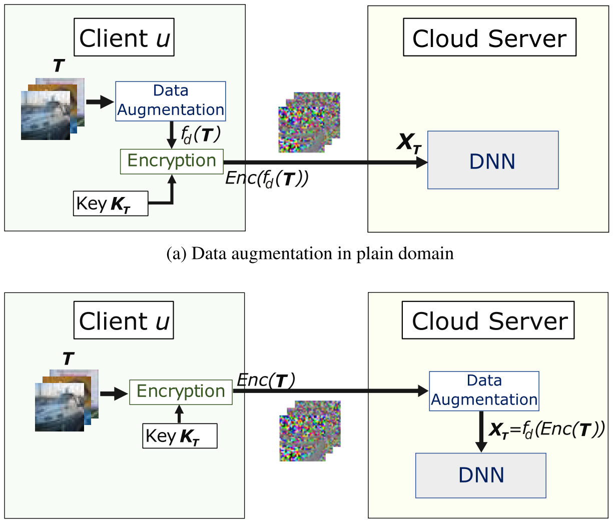
III. PROPOSED METHOD

In this section, an overview of privacy-preserving DNNs is provided. Then, the proposed image encryption and properties of encrypted images are described. In addition, data augmentation and the security of the proposed method are discussed.

A. NOTATION

The following notations are used throughout this paper:

- U and V are used to denote the width and height of an image.
- A full color image is denoted by I and is composed of red (I_R), green (I_G), and blue (I_B) color channels.
- A pixel of I_R, I_G, or I_B is denoted by p.
- p_R, p_G, and p_B denote pixel values of I_R, I_G, and I_B, respectively.
- n denotes the number of pixels of I.
- An image encryption algorithm is represented by $Enc(.)$.
- An encrypted image I_e is written as $I_e = Enc(I)$.
- $\boldsymbol{T}$ denotes a set of plain images for training, which consists of g images, where $\boldsymbol{T} = \{I_{t_1}, I_{t_2}, \ldots, I_{t_g}\}$.
- A set of plain images for testing $\boldsymbol{Q}$ includes h testing images so that $\boldsymbol{Q} = \{I_{q_1}, I_{q_2}, \ldots, I_{q_h}\}$.
- $\boldsymbol{X_T}$ and $\boldsymbol{X_Q}$ denote sets of input images used to train and test a model, respectively.
- A secret key $\boldsymbol{K}$ denotes a set of keys used for image encryption.



(a) Data augmentation in plain domain

(b) Data augmentation in encrypted domain

FIGURE 2. Frameworks of model training for image classification. (a) A DNN model is trained by $Enc(f_d(T))$. (b) $f_d(Enc(T))$ is used for training a model.

- K_T denotes a secret key set used for encrypting T, namely, $K_T = \{K_{t_1}, K_{t_2}, \ldots, K_{t_g}\}$. For example, $Enc(I_{t_1})$ is obtained by encrypting I_{t_1} with K_{t_1}.
- A set of secret key K_Q is utilized for encrypting Q where $K_Q = \{K_{q_1}, K_{q_2}, \ldots, K_{q_h}\}$. For example, $Enc(I_{q_h})$ is obtained by encrypting I_{q_h} with K_{q_h}.
- $f_d(.)$ denotes a data augmentation process.

B. OVERVIEW OF PRIVACY-PRESERVING DNNs

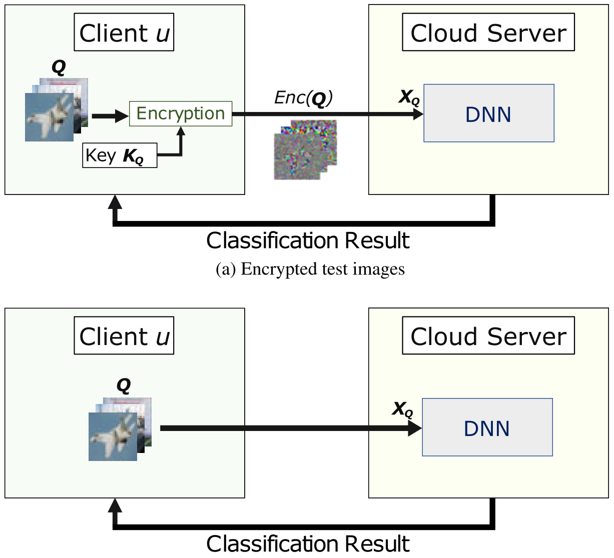
Figure 2 illustrates the training frameworks used for image classification used in this paper.

- ***Data augmentation in plain domain:*** As shown in Fig. 2(a), data augmentation is first carried out to T with labels, so $f_d(T)$ is obtained. Then, a client u encrypts $f_d(T)$ by using K_T to protect the visual information. $Enc(f_d(T))$ with labels is sent to a cloud server to train DNNs. As a result, $X_T = Enc(f_d(T))$.
- ***Data augmentation in encrypted domain:*** A client u encrypts T with labels to protect the visual information by using K_T, as shown in Fig. 2(b). Then, $Enc(T)$ with labels is uploaded to a cloud server. Eventually, data augmentation is carried out to $Enc(T)$; therefore, $X_T = f_d(Enc(T))$.

After a DNN model is trained by using encrypted images, the classification results can be returned by using two testing frameworks as follows.

- ***Test with encrypted images:*** The client u encrypts Q by using K_Q and sends $Enc(Q)$ to a server, as shown in Fig. 3(a). Hence, $X_Q = Enc(Q)$.
- ***Test with plain images:*** The client u sends Q to a server, as shown in Fig. 3(b). As a result, $X_Q = Q$.

Then, the server solves a classification problem with an image classification model trained in advance, and then returns the classification results to the client.



(a) Encrypted test images

(b) Plain test images

FIGURE 3. Frameworks of model testing for image classification. (a) $X_Q = Enc(Q)$ is applied to trained model. (b) Trained model is tested by $X_Q = Q$.

Note that the server has no secret key, so clients are able to control the privacy of images by themselves even when the classification process is done in the server. In addition, as demonstrated later, the proposed scheme allows us to carry out data augmentation in the encrypted domain.

The conventional privacy-preserving methods [18], [19] encrypt training and testing images by using a common security key. In this paper, the use of different encryption keys is discussed as follows.

- ***Same encryption key:*** Like the conventional methods [18], [19], all training and testing images are encrypted by using only one secret key, i.e. $K_{t_1} = K_{t_2} = \ldots = K_{t_g} = K_{q_1} = \ldots = K_{q_h} = K$.
- ***Different encryption keys:*** The different secret keys are independently assigned to training and testing images, i.e. $K_{t_1} \neq K_{t_2} \neq \ldots \neq K_{t_g} \neq K_{q_1} \neq \ldots \neq K_{q_h}$.

In this paper, we propose encrypting images by using different encryption keys for the first time. All clients are able to utilize independent keys for training and testing a model. Hence, there is no need to manage the keys.

In addition, since the proposed method and DNN models are independent, the proposed method is expected to be applicable to any DNNs.

C. PROPOSED IMAGE ENCRYPTION

In this section, we present our novel perceptual image encryption method that allows us not only to use different encryption keys but to also carry out data augmentation in the encrypted domain.

To generate an encrypted training image $Enc(I_{t_j})$ by using K_{t_j} from I_{t_j}, $j \in \{1, 2, \ldots, g\}$, three steps are carried out, as shown in Fig. 4. Note that $K_{t_j} = \{K_{NP}^{t_j}, K_{CS}^{t_j}\}$ consists of a set of secret keys for negative-positive transformation (NP)

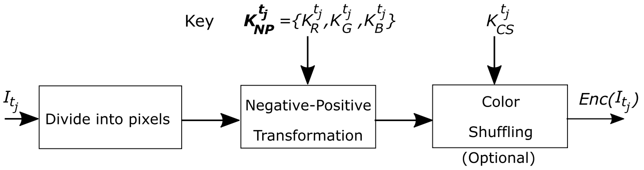


FIGURE 4. **Proposed image encryption.**

TABLE 1. **Permutation of color components with random integers. For example, if random integer is equal to 2, p_R is replaced by p_G, and p_G is replaced by p_R, while p_B is not replaced.**

Random Integer	Three Color Channels		
	p_R	p_G	p_B
0	p_R	p_G	p_B
1	p_R	p_B	p_G
2	p_G	p_R	p_B
3	p_G	p_B	p_R
4	p_B	p_R	p_G
5	p_B	p_G	p_R

and a key for color shuffling (CS), which are denoted by $\boldsymbol{K}_{NP}^{t_j}$ and $K_{CS}^{t_j}$, respectively.

1) Divide a color image I_{t_j} with $U \times V$ pixels into pixels.
2) Individually apply NP to each pixel of the three RGB color channels, $I_R^{t_j}$, $I_G^{t_j}$, and $I_B^{t_j}$, by using a random binary integer generated by $\boldsymbol{K}_{NP}^{t_j} = \{K_R^{t_j}, K_G^{t_j}, K_B^{t_j}\}$, which consists of $K_R^{t_j}$, $K_G^{t_j}$, and $K_B^{t_j}$ used for encrypting $I_R^{t_j}$, $I_G^{t_j}$, and $I_B^{t_j}$, respectively. In this step, a transformed pixel value of the i-th pixel, p_c', is calculated using
$$p_c' = \begin{cases} p_c & (r(i) = 0) \\ p_c \oplus (2^L - 1) & (r(i) = 1), \end{cases} \tag{1}$$
where $r(i)$ is a random binary integer generated by $\boldsymbol{K}_{NP}^{t_j}$, $c \in \{R, G, B\}$, and p_c is the pixel value of I_{t_j} with L bits per pixel. The value of the occurrence probability $P(r(i)) = 0.5$ is used to invert bits randomly [14].
3) (Optional) Shuffle three color components of each pixel by using an integer randomly selected from six integers generated by a key $K_{CS}^{t_j}$ as shown in Table 1.

For generating encrypted test images $Enc(I_{q_l})$, $l \in \{1, 2, \ldots, h\}$, the same encryption steps are carried out as for training images by using $\boldsymbol{K}_{q_l}$.

Images encrypted by using the proposed method are illustrated in Fig. 5, where Fig. 1(a) is the original image. It is proved that the visual information of the images was protected as well as in Fig. 1(d). Moreover, the proposed encryption and a DNN model are independent; therefore, encrypted images can be applied to any DNNs.

D. DATA AUGMENTATION IN ENCRYPTED DOMAIN

To solve complex tasks, a large amount of data is necessary to train DNNs. Data augmentation aims to enlarge the number of data points used for training and enables us to avoid the overfitting of DNNs. Many data augmentation techniques have already been proposed, e.g., horizontal/vertical flip, random crop, random rotation, cutout, and random erasing [41]. Data augmentation is required to be done in both clients and servers when training DNNs due to the following reasons. First, it is necessary for the servers to enlarge the number of training data if there is not enough training data for model training. The second reason is that data augmentation in servers can reduce the communication cost. However, in general, conventional privacy-preserving methods have to perform data augmentation in clients, namely, data augmentation in servers is not available [18].

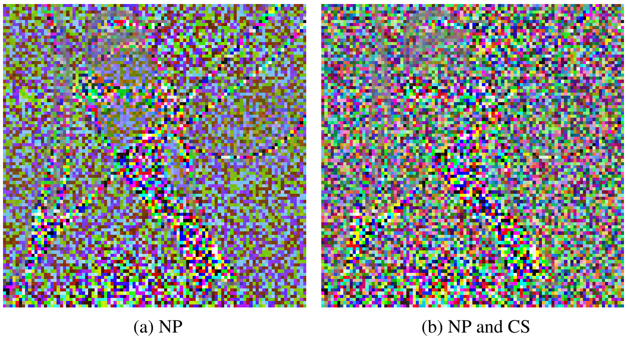

(a) NP
(b) NP and CS

FIGURE 5. **Examples of images encrypted by proposed encryption method, where Fig. 1(a) is original. (a) Image encrypted by NP. (b) Image encrypted by NP and CS.**

In this paper, some data augmentation techniques are demonstrated to be applied to images encrypted by the proposed method, as shown in Fig 2(b). Here, the following well-known techniques are utilized for data augmentation in the encrypted domain:

- *Horizontal flip*: flips original images horizontally. Therefore, the number of original images is doubled by horizontal flipping.
- *Shifting:* shifts the pixel locations of original images by four pixels on both the horizontal and vertical axes. Hence, the number of original images is increased fourfold.

In general, data augmentation is randomly performed every batch generation during model training. In this paper, to avoid the randomness of data augmentation, every possible data augmentation technique is carried out. As a result, the number of images is increased eightfold by horizontal flipping and shifting. For example, if $\boldsymbol{T}$ consists of 50K images, the total number of images of $f_d(\boldsymbol{T})$ is 400K images.

E. REQUIREMENTS OF ENCRYPTED IMAGES

Image encryption methods for privacy-preserving DNNs should meet the following requirements.

- ***Visual information protection:*** to protect an individual, the time, and the location of a taken photograph.
- ***Lightweight computation:*** to train and test privacy-preserving DNNs with the same computational cost as with plain images.
- ***Low damage to DNNs:*** to maintain the performance of DNNs as with plain images.

- ***Data augmentation in encrypted domain:*** to carry out data augmentation on encrypted images.
- ***Security:*** to provide robustness against COA.

In this paper, the proposed method considers all requirements of encrypted images mentioned above.

F. SECURITY EVALUATION

Security mostly refers to protection from adversarial forces. The visual information of encrypted images has to be difficult to reconstruct. In this paper, we assume that a cloud server is semi-trusted, so a client encrypts images to protect the visual information before sending the images to the server. Hence, we focus on robustness against COAs, such as brute-force attacks.

1) KEY MANAGEMENT

In conventional privacy-preserving DNNs, training and testing images are encrypted by using one common security key. This means that all images used for training and testing DNNs have to be encrypted by using the same encryption key. Therefore, clients are required to manage the secret keys used for encrypting training and testing images. Namely, the clients have to share a secret key to other clients in order to use a trained model, and all encrypted images are vulnerable if the key is leaked. As a result, the clients have to not only confidentially store the key in trusted environments but also transmit the key through a trusted channel.

In comparison, we propose encrypting training and testing images by using independent security keys, namely, images encrypted by different encryption keys can be utilized for training and testing a DNN model. Hence, there is no need to manage the keys for the proposed privacy-preserving DNNs. In addition, under the use of different keys, it is more difficult for adversaries to carry out collusion attacks as well as known-plaintext attacks (KPAs).

2) BRUTE-FORCE ATTACK

If I with $U \times V$ pixels is divided into pixels, the number of pixels n is given by

$$n = U \times V. \tag{2}$$

The key spaces of negative-positive transformation (N_{NP}) and color component shuffling (N_{CS}) are represented by

$$N_{NP}(n) = 2^{3n}, N_{CS}(n) = \left({}_3P_3\right)^n = 6^n. \tag{3}$$

Consequently, the key space of images encrypted by using the proposed encryption scheme, $N(n)$, is represented by the following.

$$\begin{aligned} N(n) &= N_{NP}(n) \cdot N_{CS}(n) \\ &= 2^{3n} \cdot 6^n \end{aligned} \tag{4}$$

In contrast, in Tanaka's method [18], I with $U \times V$ pixels is divided into blocks each with 4×4 pixels, and each block is split into upper 4-bit and lower 4-bit images to generate 6-channel image blocks. Then, the intensities of randomly selected pixels are reversed. Eventually, the pixels in each block are shuffled with the same pattern.

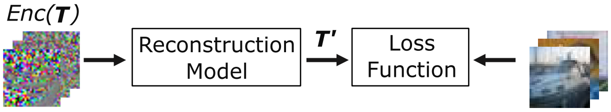


FIGURE 6. Training framework of DNN-based COA model.

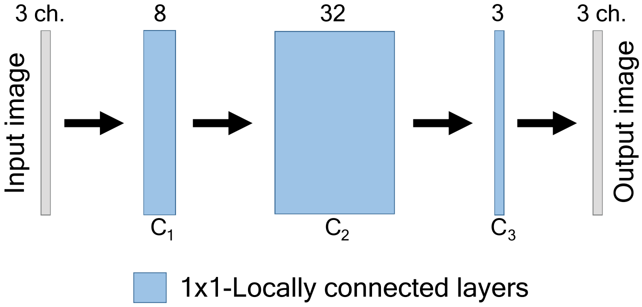


FIGURE 7. Network architecture of reconstruction model. Each box denotes multi-channel feature map produced by each layer. Number of channels is denoted above each box. Feature map resolutions are $U \times V$. Kernel size and stride of locally connected layers are (1,1).

The key space of Tanaka's method [18], N_{tanaka}, is given by

$$N_{tanaka} = 96! \cdot 2^{96}. \tag{5}$$

$N(n)$ is equal to N_{tanaka} when n is approximately equal to 106.4. Therefore, the proposed encryption has a larger key space than Tanaka's method if $U \times V$ is more than 11×11 pixels.

3) DNN-BASED CIPHERTEXT-ONLY ATTACK

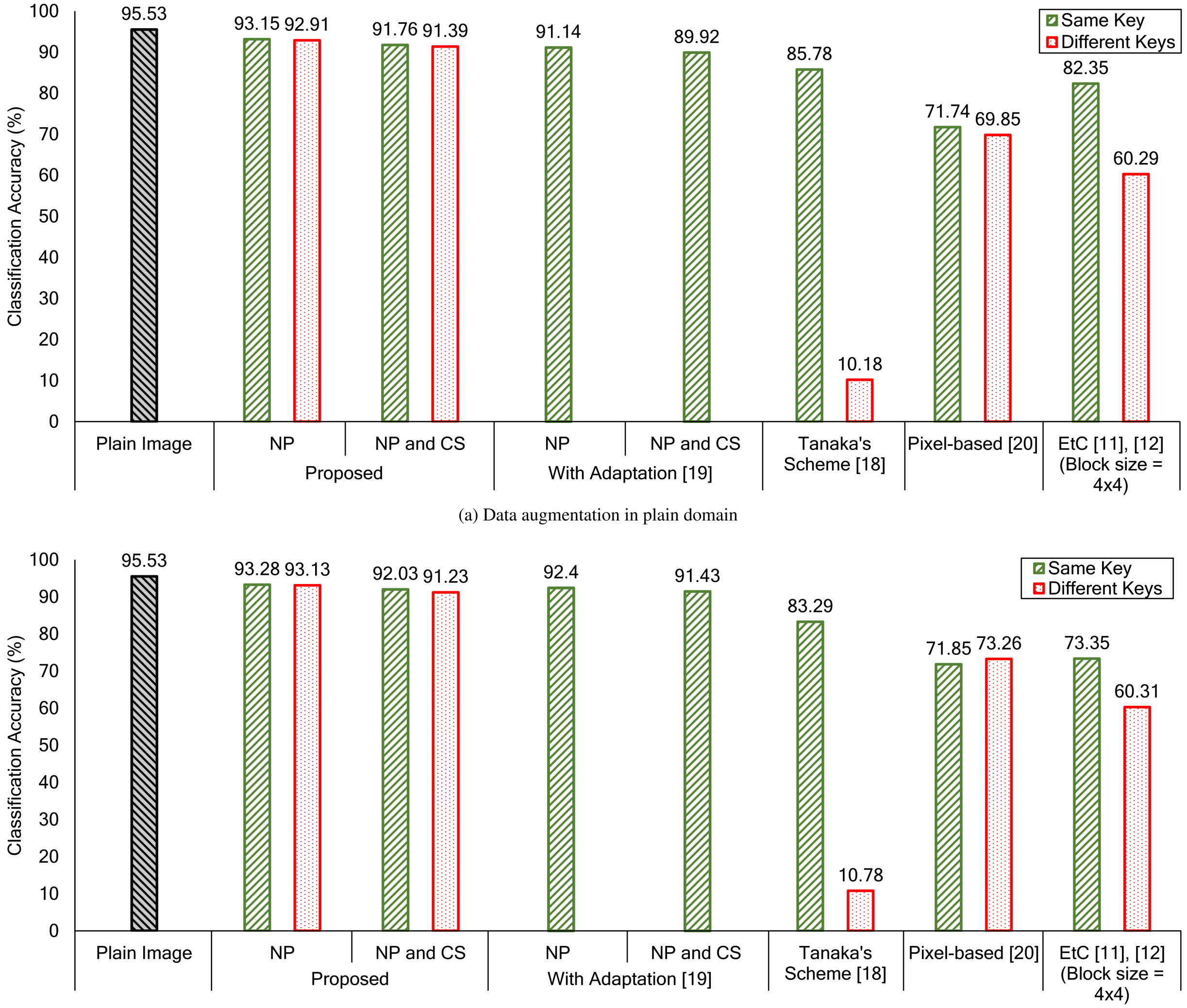
A DNN-based COA may be able to reconstruct the visual information of I from $I_e = Enc(I)$. Therefore, robustness against this attack has to be evaluated.

As shown in Fig. 6, a reconstruction model is trained by using $Enc(\boldsymbol{T})$, and then the training loss is calculated from a set of reconstructed images ($\boldsymbol{T'}$) and $\boldsymbol{T}$. The DNN-based COA model consists of three 1×1-locally connected layers (C_1, C_2, and C_3) each with both a kernel size and a stride of (1,1), as shown in Fig. 7. A locally connected layer similarly works as a 1×1-convolution layer, but weights are unshared. As shown in Fig. 7, the numbers of filters of C_1, C_2, and C_3 are 8, 32, and 3, respectively. In the testing process, the reconstruction model, which is trained by $Enc(\boldsymbol{T})$, is utilized to recover encrypted test images ($Enc(\boldsymbol{Q})$) to obtain the reconstructed test images ($\boldsymbol{Q'}$).

Images encrypted under the use of different keys will be demonstrated to be robust against this attack later.

IV. EXPERIMENTS

To confirm that the proposed scheme is effective, we evaluated the performance of image classification accuracy and robustness against the DNN-based COA attack under various conditions.



(a) Data augmentation in plain domain

(b) Data augmentation in encrypted domain

FIGURE 8. Image classification accuracy when testing DNN models (ResNet-18) with $Enc(\boldsymbol{Q})$, where models were trained with following training data: (a) $Enc(f_d(\boldsymbol{T}))$ and (b) $f_d(Enc(\boldsymbol{T}))$. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.

A. IMAGE CLASSIFICATION

1) EXPERIMENTAL CONDITIONS

We employed the image database CIFAR10, which contains 32 × 32 pixel color images and consists of 50K training images and 10K test images in 10 classes [42]. Two data augmentation techniques (shifting and horizontal flip) were used to enlarge the number of training images for all cases, i.e. both plain images and encrypted ones. Hence, the number of training images was 400K, as described in Section III-D. In addition, we conducted the experiments under the use of two encryption key conditions: same encryption key and different encryption keys, as described in Section III-B.

We evaluated the image classification accuracy of encrypted images under the use of deep residual networks (ResNet-18) [43], [44], which consist of 18 layers, and densely connected convolutional networks (DenseNet) [45].

The models with ResNet-18 were trained by using stochastic gradient descent (SGD) with momentum for 200 epochs. The learning rate was initially set to 0.1 and was decreased by a factor of 5 at 60, 120, and 160 epochs. We used a weight decay of 0.0005, a momentum of 0.9, and a batch size of 128. According to [45], the models with DenseNet were trained by using SGD for 300 epochs. The initial learning rate was set to 0.1, and was lowered by 10 times at 150 and 225 epochs. Moreover, we used a weight decay of 0.0001, a momentum of 0.9, and a batch size of 64.

2) DATA AUGMENTATION IN PLAIN DOMAIN (ResNet-18)

Figure 8(a) shows the classification accuracy when testing DNN models (ResNet-18) with $Enc(\boldsymbol{Q})$, where data augmentation was carried out in the plain domain, as shown in Fig. 2(a). We evaluated the performance under two key conditions: same encryption key, and different encryption

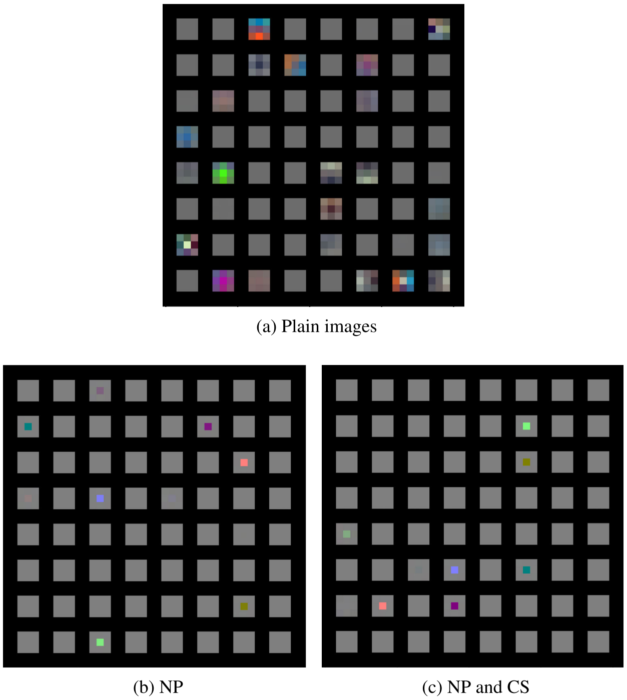


FIGURE 9. 64 filters with size of 3×3 in first convolution layer of ResNet-18. Data augmentation was carried out in plain domain. (a) Model trained by using plain images. (b) Model trained by using encrypted images with NP under the use of different keys. (c) Model trained by using encrypted images with NP and CS under the use of different keys.

keys. The performance of the proposed method was compared with four conventional methods [11], [12], [18]–[20].

The proposed method was able to maintain a high classification performance and provided the highest accuracy in the encryption methods even when the training and testing images were encrypted under the use of the same key. In addition, under the use of different keys, the performances of the conventional methods [11], [12], [18], [20] degraded heavily, although the proposed method was able to maintain almost the same accuracy as under the use of the same key. Note that the conventional pixel-based method considers only using the same encryption key [19]. The accuracy of the conventional method under the use of different encryption key was confirmed to be almost the same as that with the same key.

Figure 9 shows 64 filters with a size of 3 × 3 in the first convolution layer of ResNet-18. From this figure, we can see that each model has filters different from those other models, although the classification accuracies of the models are almost the same.

3) DATA AUGMENTATION IN ENCRYPTED DOMAIN (ResNet-18)

In Fig. 8(b), the performance of the proposed method is compared with the conventional methods [11], [12], [18]–[20] after data augmentation was carried out in the encrypted domain, as shown in Fig. 2(b).

The conventional methods [11], [12], [18], [20] were heavily damaged by data augmentation carried out in the encrypted domain even when images were encrypted under the use of the same key. The proposed method provided the highest accuracy in all encryption methods as well as for data augmentation in the plain domain. In contrast, it was proved that the proposed method was able to maintain the classification performance under the use of both key conditions.

4) USE OF PLAIN TEST IMAGES (ResNet-18)

Figure 10 shows that the classification accuracy when plain images $\boldsymbol{Q}$ were used for testing DNNs, where the models were trained by using encrypted images.

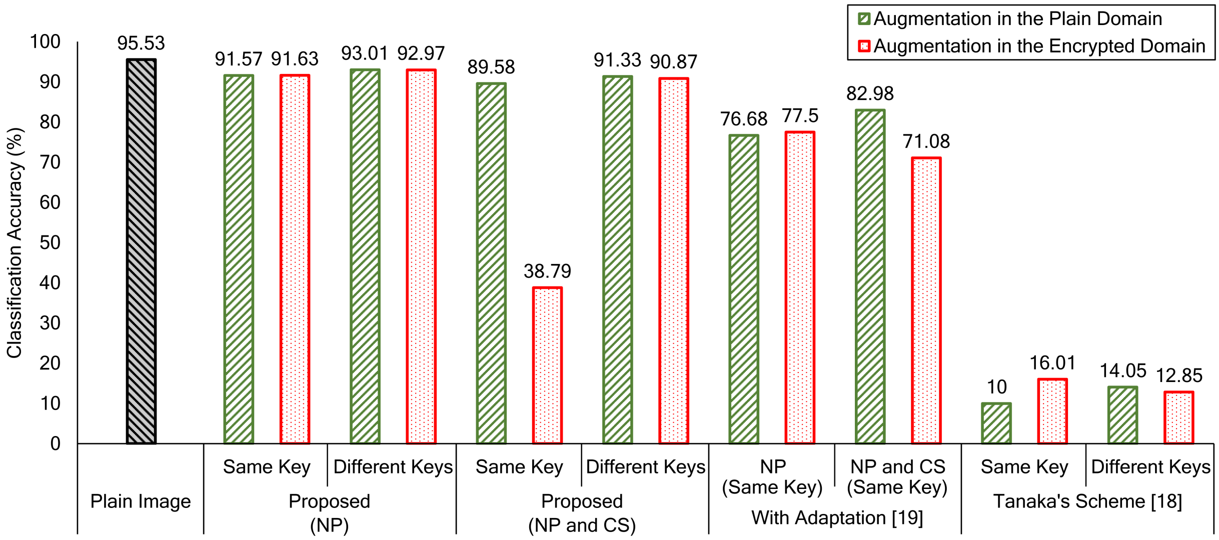
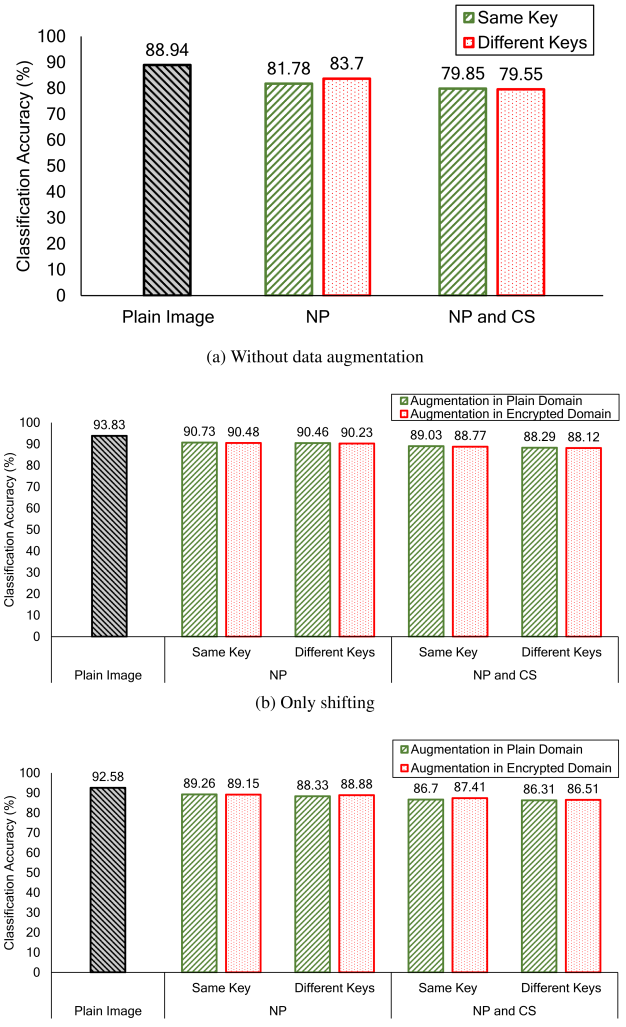


FIGURE 10. Image classification accuracy when testing DNN models (ResNet-18) with plain images Q, where models were trained with encrypted images. Horizontal axis corresponds to conditions of images used for training DNNs. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.



(a) Without data augmentation

(b) Only shifting

(c) Only flipping

FIGURE 11. Image classification accuracy when testing DNN models (ResNet-18) with $Enc(\boldsymbol{Q})$, where models were trained with following data augmentation conditions: (a) Without data augmentation, (b) Shifting, and (c) Horizontal flipping. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.

The models trained by Tanaka's scheme were not able to classify $\boldsymbol{Q}$ due to the damages caused by the encryption methods. In comparison, it was confirmed that the proposed method and the conventional one [19] under the use of different keys enabled us to test the DNNs with plain images. In addition, the performance of the proposed method was maintained under the use of two data augmentation conditions.

5) INFLUENCE OF DATA AUGMENTATION

To show the influence of data augmentation in more detail, we trained ResNet-18 by using plain images and encrypted ones under the following data augmentation conditions.

- *Without data augmentation:* Data augmentation was not carried out. The number of training images was 50k.

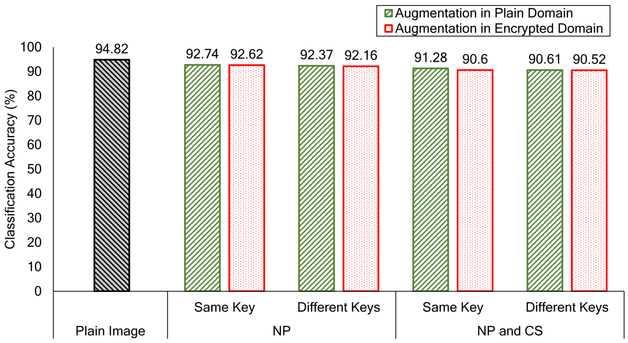


FIGURE 12. Image classification accuracy when testing DNN models (DenseNet) with $Enc(\boldsymbol{Q})$, where depth and growth rate of DenseNet were 40 and 12, respectively. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.

- *Shifting:* Only shifting was utilized for generating training images. As a result, the number of training images was 200k.
- *Horizontal flipping:* Only horizontal flipping was performed. Hence, the number of training images was 100k.

Figure 11 illustrates the influence of data augmentation on classification performance. Compared with the results in Fig. 11(a), two augmentation techniques in the encrypted domain were confirmed to improve the performance of the proposed method, respectively, and using both techniques provided better results than in Fig. 11, as shown in Fig. 8. In addition, the difference of the accuracy between the proposed method and plain images was shown to be reduced by using data augmentation techniques.

6) CLASSIFICATION ACCURACY (DenseNet)

To confirm that the proposed method and models are independent, we utilized DenseNet as a DNN model for training and testing. The performance was evaluated under the use of two data augmentation conditions: data augmentation in the plain domain, and data augmentation in the encrypted domain.

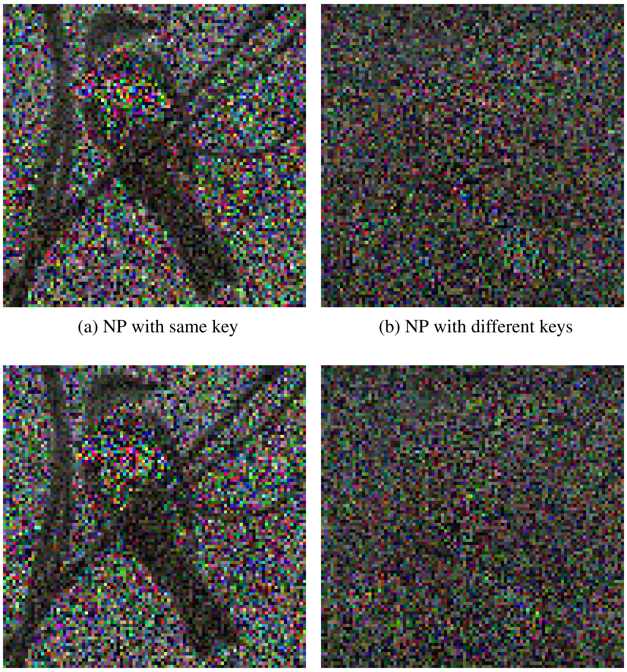
As shown in Fig. 12, the classification accuracy of the proposed method had almost the same tendency as when using ResNet-18 as a model. In addition, the proposed method was confirmed to maintain the performance even when data augmentation was carried out in the encrypted domain. Therefore, it was proved that the proposed method and models are independent.

B. ROBUSTNESS AGAINST DNN-BASED ATTACK

As described in Section III-F, a DNN-based COA may be able to reconstruct the visual information of I from I_e. Therefore, robustness against a DNN-based COA in Fig. 6 is discussed here.

1) EXPERIMENTAL CONDITIONS

We employed the STL-10 dataset, which consists of 5K training images and 8K testing images [46], and each image has



(a) NP with same key

(b) NP with different keys

(c) NP and CS with same key

(d) NP and CS with different keys

FIGURE 13. Examples of reconstructed images. (a) T and Q were encrypted by using NP with same key. (b) T) and Q were encrypted by using NP under use of different keys. (c) T and Q were encrypted by using NP and CS with same key. (d) T) and Q were encrypted by using NP and CS under use of different keys.

96 × 96 pixels. Note that data augmentation was not carried out in the experiment.

The network in Fig. 7 was trained by using SGD with momentum for 70 epochs, and the mean squared error (MSE), which compared the differences between $\boldsymbol{T'}$ and $\boldsymbol{T}$, was used as a loss function. The learning rate was initially set to 0.1 and decreased by a factor of 10 at 40 and 60 epochs. We used a weight decay of 0.0005, a momentum of 0.9, and a batch size of 128.

The robustness against the DNN-based COA was evaluated in terms of the visibility of reconstructed images.

2) RESULTS

Examples of reconstructed images under the use of same and different encryption keys are shown in Fig. 13, where Fig. 1(a) is the original image.

The visual information was slightly recovered by the DNN-based COA when the model was trained by using encrypted images with the same key, and the test image was encrypted with the same key, as shown in Fig. 13(a), and 13(c). In comparison, when the model was trained by using encrypted images with different keys, the reconstructed images had almost no visual information, as shown in Fig. 13(b), and 13(d).

Table 2 shows the average structural similarity (SSIM) values and average peak signal-to-noise ratio (PSNR) ones of 8K testing images, where lower values mean lower visual information. In other words, lower scores indicate that encrypted images are more robust against DNN-based attacks. From this table, it was confirmed that the use of different keys enhances robustness against DNN-based attacks.

TABLE 2. Average SSIM and PSNR of 8K reconstructed images.

Key Conditions	Encryption	SSIM	PSNR (dB)
Same encryption key	NP	**0.1732**	**10.73**
	NP and CS	0.1715	10.72
Different encryption keys	NP	0.0424	9.50
	NP and CS	0.0425	9.50

V. CONCLUSION

We presented a novel privacy-preserving scheme for deep neural networks (DNNs) that enables us not to only apply images without visual information to DNNs but to also consider the use of independent encryption keys for both training and testing images for the first time. In addition, the proposed privacy-preserving scheme for DNNs allows us to train a DNN model with encrypted images and then test it with plain images. Therefore, there is no need to manage keys. Moreover, we proposed performing data augmentation in the encrypted domain. In an experiment, we evaluated the performance of the proposed method in terms of image classification accuracy and robustness against a DNN-based attack. The experimental results demonstrated that the proposed method, under the use of independent encryption keys, was confirmed to maintain a high classification performance, and it was robust against the DNN-based attack. Moreover, the results confirmed that the proposed scheme was able to classify plain images as well as encrypted images, even when data augmentation was carried out in the encrypted domain.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[2] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Mach. Learn.*, Bejing, China, Jun. 2014, vol. 32, no. 1, pp. 647–655.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, vol. 1, 2012, pp. 1097–1105.

[4] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, May 2015, pp. 1–5.

[5] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning," in *Proc. ICLR*, May 2018, pp. 1–27.

[6] I. Ito and H. Kiya, "One-time key based phase scrambling for phase-only correlation between visually protected images," *EURASIP J. Inf. Secur.*, vol. 2009, no. 841045, pp. 1–11, 2010.

[7] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," in *Proc. 34th IEEE Symp. Reliable Distrib. Syst.*, Sep. 2015, pp. 11–20.

[8] J. Zhou, X. Liu, O. C. Au, and Y. Y. Tang, "Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 39–50, Jan. 2014.

[9] Y. Zhang, B. Xu, and N. Zhou, "A novel image compression–encryption hybrid algorithm based on the analysis sparse representation," *Opt. Commun.*, vol. 392, pp. 223–233, Jun. 2017.

[10] T. Y. Liu, K. J. Lin, and H. C. Wu, "ECG data encryption then compression using singular value decomposition," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 3, pp. 707–713, May 2018.

[11] K. Kurihara, S. Shiota, and H. Kiya, "An encryption-then-compression system for jpeg standard," in *Proc. Picture Coding Symp. (PCS)*, 2015, pp. 119–123.

[12] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for JPEG/Motion JPEG standard," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 98, no. 11, pp. 2238–2245, Nov. 2015.

[13] W. Sirichotedumrong and H. Kiya, "Grayscale-based block scrambling image encryption using ycbcr color space for encryption-then-compression systems," *APSIPA Trans. Signal Inf. Process.*, vol. 8, p. e7, Feb. 2019.

[14] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 6, pp. 1515–1525, Jun. 2019.

[15] T. Chuman, K. Kurihara, and H. Kiya, "On the security of block scrambling-based etc systems against jigsaw puzzle solver attacks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2157–2161.

[16] V. Itier, P. Puteaux, and W. Puech, "Recompression of jpeg crypto-compressed images without a key," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.

[17] T. Maekawa, A. Kawamura, Y. Kinoshita, and H. Kiya, "Privacy-preserving svm computing in the encrypted domain," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2018, pp. 897–902.

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

[19] 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)*, 2019, pp. 674–678.

[20] M. T. Gaata and F. F. Hantoosh, "An efficient image encryption technique using chaotic logistic map and rc4 stream cipher," *Int. J. Modern Trends Eng. Res.*, vol. 3, no. 9, pp. 213–218, 2016.

[21] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proc. 25th USENIX Secur. Symp. (USENIX Security)*, Austin, TX, USA, Aug. 2016, pp. 601–618.

[22] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 36–52.

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

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

[25] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. USENIX Secur. Symp.*, San Diego, CA, USA, Aug. 2014, pp. 17–32.

[26] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," Dec. 2013, *arXiv:1312.6199*. [Online]. Available: https://arxiv.org/abs/1312.6199

[27] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, "Intriguing properties of neural networks," in *Proc. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 387–402.

[28] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, Sep. 2017, pp. 2021–2031.

[29] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.0.0," Jul. 2018, *arXiv:1807.01069*. [Online]. Available: https://arxiv.org/abs/1807.01069

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," Jun. 2017, *arXiv:1706.06083*. [Online]. Available: https://arxiv.org/abs/1706.06083

[31] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 39–57.

[32] Y. Aono, T. Hayashi, L. T. Phong, and L. Wang, "Privacy-preserving logistic regression with distributed data sources via homomorphic encryption," *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 8, pp. 2079–2089, 2016.

[33] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 805–817.

[34] T. Araki, A. Barak, J. Furukawa, T. Lichter, Y. Lindell, A. Nof, K. Ohara, A. Watzman, and O. Weinstein, "Optimized honest-majority mpc for malicious adversaries—Breaking the 1 billion-gate per second barrier," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 843–862.

[35] W. Lu, S. Kawasaki, and J. Sakuma, "Using fully homomorphic encryption for statistical analysis of categorical, ordinal and numerical data," *IACR Cryptol. ePrint Archive*, vol. 2016, p. 1163, Dec. 2016.

[36] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.

[37] 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.

[38] L. T. Phong and T. T. Phuong, "Privacy-preserving deep learning via weight transmission," Sep. 2018, *arXiv:1809.03272*. [Online]. Available: https://arxiv.org/abs/1809.03272

[39] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2016-3, Feb. 2016.

[40] Y. Wang, J. Lin, and Z. Wang, "An efficient convolution core architecture for privacy-preserving deep learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[41] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," Aug. 2017, *arXiv:1708.04896*. [Online]. Available: https://arxiv.org/abs/1708.04896

[42] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 630–645.

[45] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2261–2269.

[46] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," *J. Mach. Learn. Res.*, vol. 15, pp. 215–223, 2011.

WARIT SIRICHOTEDUMRONG received the B.Eng. and M.Eng. degrees from the King Mongkut's University of Technology Thonburi, Thailand, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with Tokyo Metropolitan University. His research interests include image processing and information security. He received the International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC) Best Paper Award, in 2018, and the 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE 2019) Best Paper Award.

YUMA KINOSHITA received the B.Eng. and M.Eng. degrees from Tokyo Metropolitan University, Japan, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree. His research interest is in the area of image processing. He is a Student Member of IEICE. He received the IEEE ISPACS Best Paper Award, in 2016, the IEEE Signal Processing Society Japan Student Conference Paper Award, in 2018, and the IEEE Signal Processing Society Tokyo Joint Chapter Student Award, in 2018.

HITOSHI KIYA received the B.E. and M.E. degrees from the Nagaoka University of Technology, Japan, in 1980 and 1982, respectively, and the Dr.Eng. degree from Tokyo Metropolitan University, in 1987. In 1982, he joined Tokyo Metropolitan University, where he became a Full Professor, in 2000. From 1995 to 1996, he attended The University of Sydney, Australia, as a Visiting Fellow. He is a Fellow of IEICE and ITE. He is a member of nine technical committees, including the APSIPA Image, Video, and Multimedia Technical Committee (TC) and the IEEE Information Forensics and Security TC. He was a recipient of numerous awards, including nine best paper awards. He has been the Chair of two technical committees. He has organized a lot of international conferences in such roles as the TPC Chair of the IEEE ICASSP 2012 and as the General Co-Chair of the IEEE ISCAS 2019. He has been an Editorial Board Member of eight journals, including the IEEE Transactions on Signal Processing, the IEEE Transactions on Image Processing, and the IEEE Transactions on Information Forensics and Security. He was also the President of the IEICE Engineering Sciences Society, from 2011 to 2012, where he served as the Vice President and the Editor-in-Chief for the IEICE Society Magazine and Society Publications. He currently serves as the President of the APSIPA, and he served as the Inaugural Vice President (Technical Activities) of APSIPA, from 2009 to 2013, and as the Regional Director-at-Large for Region 10 of the IEEE Signal Processing Society, from 2016 to 2017.