

Received 30 October 2025, accepted 13 November 2025, date of publication 20 November 2025,
date of current version 4 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3635235

RESEARCH ARTICLE

Learnable Image Encryption Without Key Management for Privacy-Preserving Vision Transformer

MARE HIROSE¹, (Graduate Student Member, IEEE), SHOKO IMAIZUMI¹, (Member, IEEE),
AND HITOSHI KIYA², (Life Fellow, IEEE)

¹Graduate School of Informatics, Chiba University, Chiba 263-8522, Japan

²Faculty of System Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan

Corresponding author: Shoko Imaizumi (imaizumi@chiba-u.jp)

This work was supported in part by JSPS KAKENHI under Grant JP25K07750; in part by JST CREST, Japan, under Grant JPMJCR20D3; and in part by JST BOOST, Japan, under Grant JPMJBS2413.

ABSTRACT We propose a privacy-preserving image classification method based on perceptual encryption that does not require centralized key management. In the proposed method, each client independently generates an encryption key to protect visual information in both training and query images. The use of independent keys allows multiple clients to use a shared model without exchanging keys and to easily update their keys whenever needed. In addition, even if a key is compromised, the impact does not propagate to other clients. The use of perceptual encryption allows us to directly apply encrypted data for training and query images in the encrypted domain, but conventional approaches with perceptual encryption are known to degrade the accuracy of image classification when independent keys are used in each client due to significant visual distortion caused by encryption. Accordingly, we demonstrate that a novel method that focuses on the compatibility between block-wise image encryption and the embedding structure of vision transformer (ViT) is effective in improving the issue. We carried out experiments to demonstrate the effectiveness of the method in terms of accuracy and robustness on CIFAR-10 and Tiny ImageNet. Compared to conventional methods, when using independent keys, the accuracy was improved by 82% for CIFAR-10 and 83% for Tiny ImageNet. In addition, resistance to various attacks including brute-force attacks and jigsaw puzzle attacks was demonstrated under the assumption of ciphertext-only attacks. These results suggest the practicality and effectiveness of the method for secure image classification in real-world multi-client environments.

INDEX TERMS Vision transformer, image encryption, privacy preserving.

I. INTRODUCTION

Distributed systems for information processing such as cloud computing and edge computing have been spreading in many fields. However, distributed systems such as cloud computing are generally not considered trustworthy, so the processing can lead to serious problems for end users, such as the unauthorized use of services, data leaks, and privacy being compromised due to unreliable providers and accidents [1]. In contrast, the spread of deep neural networks (DNNs)

The associate editor coordinating the review of this manuscript and approving it for publication was Alex James¹.

has greatly contributed to solving complex tasks for many applications, such as computer vision, biomedical systems, and information technology. Deep learning utilizes a large amount of data, which includes sensitive personal information, to train models with many parameters. Therefore, privacy-preserving learning has become an urgent challenge.

Many studies on secure, efficient, and flexible communication/storage/computing have been reported [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. Privacy-preserving approaches include homomorphic encryption, which enables direct computation on encrypted data. However, such

techniques are computationally intensive and difficult to be directly applied to state-of-the-art models [10], [11], [12], [13]. Another alternative is federated learning, where multiple clients collaboratively train a shared model without exchanging raw data [14], [15]. However, this method does not protect the privacy of query inputs. Depending on the application of DNNs, there are trade-offs between security and other requirements such as a low processing demand, bitstream compliance, and signal processing in the encrypted domain. Several perceptual encryption methods [16], [17], [18], [19], [20], [21], [22], [23] called learnable encryption have been developed to balance these trade-offs.

In this paper, we address the above challenges by leveraging the Vision Transformer (ViT) [24] framework for privacy-preserving image classification. ViTs can directly process encrypted images in which sensitive visual information has been hidden, which makes them suitable for applications where privacy is a concern. However, existing learnable encryption methods typically require the use of a shared encryption key between the model provider and clients to avoid the accuracy degradation because image features change depending on the key. This constraint introduces two major issues: (1) clients must securely share or store the key, which creates a potential security vulnerability, and (2) the model must be retrained whenever the encryption key is updated, limiting flexibility and scalability. Accordingly, a novel method that takes into account that features change depending on the key is needed in multi-client settings.

To overcome these limitations, we propose a novel method that fine-tunes a pre-trained ViT model using training images encrypted with independently generated keys for each image. The resulting model is then deployed to the cloud and shared among clients. During inference, clients individually generate independent keys to encrypt their query images, which are then sent to the model for classification. This framework eliminates the need for key sharing or storage, thereby enhancing privacy even when multiple clients access a common model.

Our main contributions are summarized below:

- a) Per-client and per-image independent key generation: We propose a method for the first time that allows clients not to only generate their own encryption keys but to also assign distinct keys to each individual image by considering a model training method that takes into account that features change depending on the key. Consequently, the method eliminates the need to manage and update keys. In addition, even if one encryption key is compromised, the privacy of the remaining images is preserved.
- b) Effectiveness and limitations: The method should maintain a high classification accuracy even in multi-client settings and be robust against attacks. We verify the effectiveness and limitations of the proposed method in terms of image classification accuracy and robustness to balance these trade-offs. In contrast, previous methods are shown to significantly reduce accuracy.

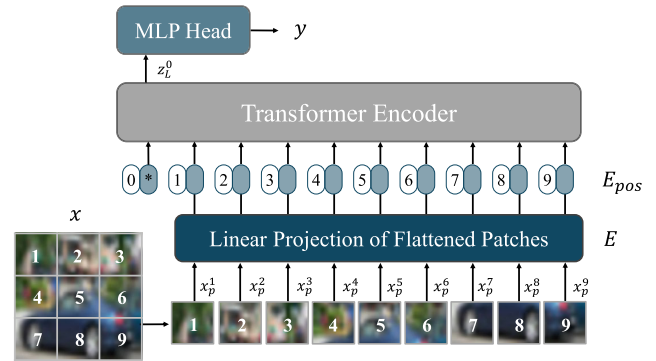


FIGURE 1. Overview of vision transformer.

The rest of this paper is organized as follows. Section II introduces ViT-based models and summarizes image encryption in the context of deep learning. Section III details the proposed method. Section IV presents experimental results, including classification performance and security analysis. Section V concludes the paper.

II. RELATED WORK

A. VISION TRANSFORMER

ViT is an image classification model, which is known to provide high classification performance [24]. Although ViT has been widely used in various studies and is a well-known model, we briefly review its structure here to help in understanding the differences between the proposed method and conventional methods. Fig. 1 illustrates an overview of ViT. First, an input image $x \in \mathbb{R}^{H \times W \times C}$ is divided into patches of a size of $P \times P$. Note that H , W , and C denote the height, width, and number of channels of the input image, respectively. Next, each patch i , $i = 1, 2, \dots, N$, is flattened using pixel values to convert it into a vector $x_p^i \in \mathbb{R}^{P^2 C}$. Then, each vector x_p^i is linearly mapped by a matrix $\mathbf{E} \in \mathbb{R}^{(P^2 C) \times D}$. A class token is added to the beginning of a sequence of patches, and $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$, which is position information of each patch, is embedded. Following the standard formulation of ViT [24], a sequence of embedded patches z_0 is defined as

$$z_0 = [x_{class}; x_p^1 \mathbf{E}; x_p^2 \mathbf{E}; \dots; x_p^i \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}_{pos},$$

$$\mathbf{E} \in \mathbb{R}^{(P^2 C) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}. \quad (1)$$

After z_0 is fed into the transformer encoder, the resulting class token z_L^0 is passed to the MLP head to output the estimated class y . The embedding structure of ViT is known to have an affinity with block-wise image encryption [21], [23].

B. LEARNABLE IMAGE ENCRYPTION

Various image transformation methods that use a secret key are often referred to as perceptual image encryption. In this paper, we focus on learnable images transformed with a secret key, which have been studied for deep learning. Learnable

encryption enables us to directly apply encrypted data to a model as training and testing data. Encrypted images have no sensitive visual information on plain images in general, so privacy-preserving learning can be carried out by using visually protected images. In addition, the use of a secret key allows us to embed unique features controlled with the key into images. Adversarial defenses and access control are carried out with encrypted data using these unique features.

A block-wise learnable image encryption (LE) with an adaptation layer [16] was introduced as the first learnable image encryption method, and then another encryption method, a pixel-wise encryption (PE) method that does not use any adaptation layer, was proposed [17]. To enhance the security of encryption, LE was also extended to an extended learnable image encryption method (ELE) [18] by adding a block scrambling (permutation) step and a pixel encryption operation with multiple keys.

However, ELE still has inferior accuracy compared with plain images, even when an additional adaptation network is used to reduce the influence of the encryption. Recently, block-wise encryption [19], [20], [21], [22], [23] was also pointed out to have a high similarity to isotropic networks such as ViT and ConvMixer [21], [25], and the similarity enables us to reduce performance degradation. However, conventional learnable encryption methods [19], [20], [21], [22], [23] have never considered models used by multiple clients. Furthermore, in all conventional learnable encryption methods, when inputting images generated with independent keys into a model, the model performance of existing approaches significantly drops. The main reason why existing approaches fail in multi-client settings is that models in existing approaches are trained with only the characteristics of images generated with a single key. To address the issue, models need to be trained with the characteristics of images randomly generated with multiple keys. Previous studies did not have this kind of focus.

C. BENEFITS OF PROPOSED METHOD

Previous learnable encryption methods assume that all clients use images encrypted with a common key. This limitation creates the following challenges in various deployments of encryption: (a) encryption keys are required to be securely managed to protect data privacy; (b) models have to be retrained if we want to update keys; and (c) model performance significantly drops if each client uses images encrypted with independent keys. From (c), previous learnable encryption methods are not effective in secure multi-client settings.

To overcome these issues, we propose a novel method for multi-client settings in which each client can independently apply its own key without requiring model retraining. By designing the encryption to be compatible with the embedding structure of ViTs, the method mitigates accuracy degradation even in multi-client settings.

III. PROPOSED METHOD

The proposed method is described here. The method allows clients to use different encryption keys from those of the model creator. In addition, multiple clients can independently prepare keys by themselves, even when they use a common model.

A. OVERVIEW

Fig. 2 illustrates an overview of a conventional method [22]. In this approach, the model creator encrypts both a pre-trained model and training images using a shared key K . The encrypted pre-trained model is then fine-tuned on the encrypted training images in cloud environments. Key K is distributed to u_L clients. Client u , $u = 1, 2, \dots, u_L$ holds n_u query images. Query image $x^{(u,i)}$ held by client u is encrypted by using key K , to generate an encrypted image $x'^{(u,i)}$, and $x'^{(u,i)}$ is then sent to the encrypted model for inference. In this framework, the provider has no key and visual information in images. However, the model creator and all clients have to use the shared key K , so they have to be assumed to be trusted. If each client uses an independent key, the performance of models is degraded compared with that of models without any encryption. In addition, if we want to update key K , the model needs to be retrained by using a new key.

Fig. 3 presents an overview of the proposed method. First, the model creator encrypts a portion of the model parameters in the pre-trained ViT using a designated encryption key K_b . Next, training images are encrypted with keys K_m in the manner consistent as shown below. The encrypted pre-trained ViT is then fine-tuned by using the encrypted training images in cloud environments.

In contrast, on the client side, each query image is encrypted using an independent encryption key generated by each client, and it is then input to the fine-tuned ViT. In particular, each query image $x^{(u,i)}$ can be encrypted using a key $K^{(u,i)}$, which is independently generated per image. The client receives a classification result from the cloud provider.

In the proposed method, encryption keys can vary for each client as well as for each image. Accordingly, each client is not required to store or manage its encryption keys.

As illustrated in Figs. 2 and 3, the main difference between the conventional and proposed methods lies in key management. In the conventional method, all clients are required to use a common key, whereas in the proposed method, each client can use independently generated keys. In addition, clients are not required to share their keys with the model creator.

B. THREAT MODEL AND SECURITY ASSUMPTIONS

A threat model includes a set of assumptions, such as an adversary's goals, knowledge, and capabilities. The aim of an attacker is to restore visual information from encrypted data. We assume that the attacker is able to use encrypted data and the encryption algorithm but does not have the secret key. Accordingly, the attacker can only perform ciphertext-only

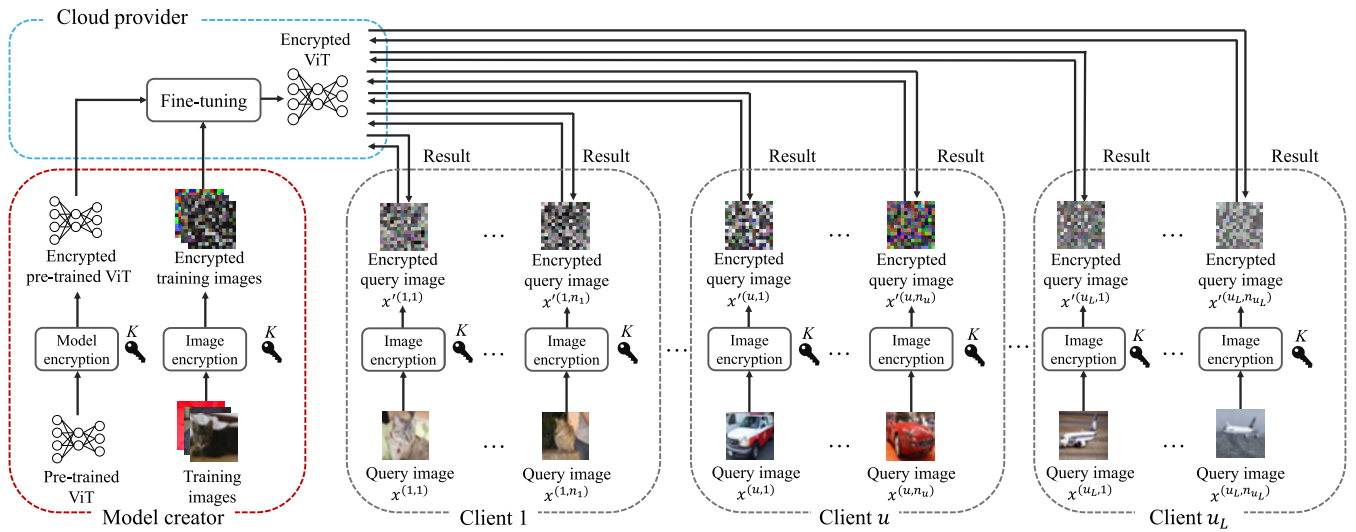


FIGURE 2. Overview of conventional method, where each client uses key shared by all clients.

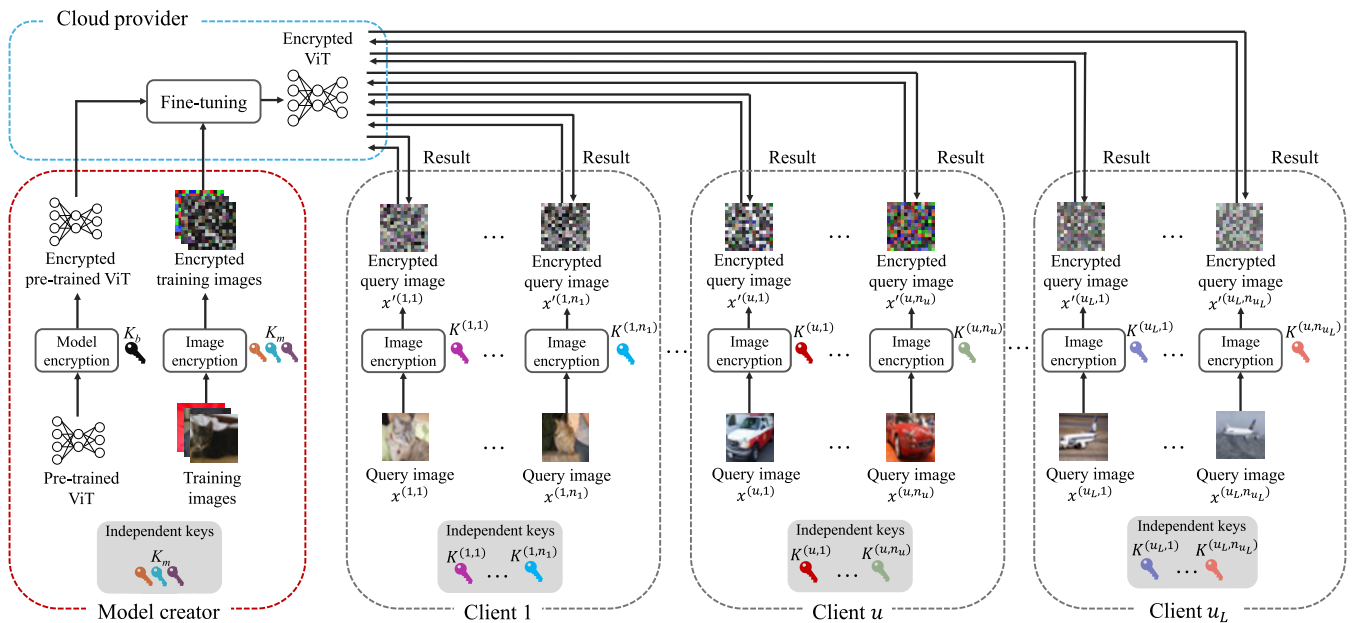


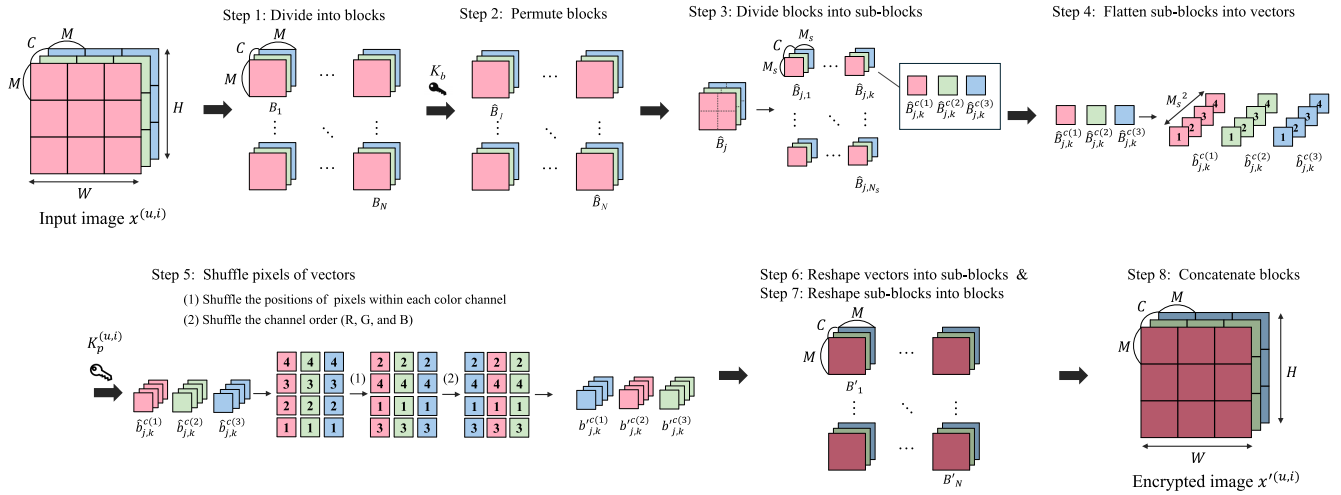
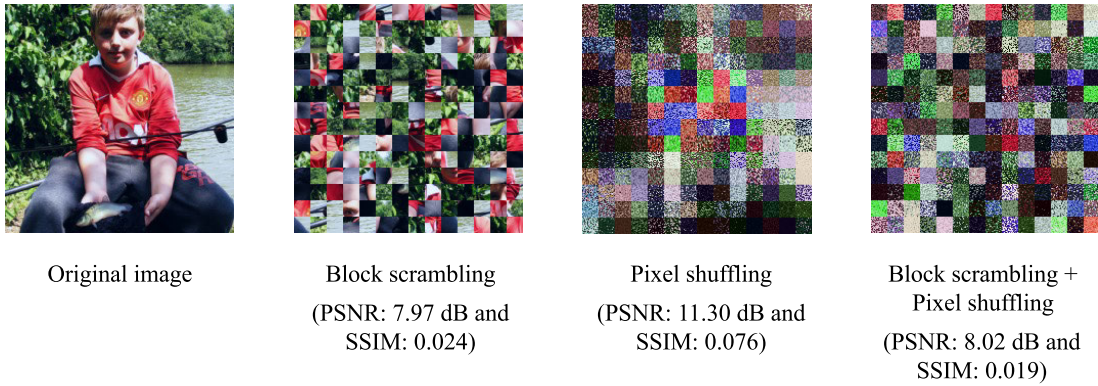
FIGURE 3. Overview of proposed method, where each client can use key generated by client itself.

attacks (COAs) using encrypted images. In Fig. 3, it is assumed that the cloud provider is untrusted, and the model creator and all other clients are semi-trusted. “Semi-trusted” means that a participant is expected to follow the rules of a protocol but may have the motivation to learn additional information if possible. Thus, they are not fully trusted; some clients may collude with external attackers or leak their keys. The proposed method is discussed by taking these environments into consideration in this paper.

C. IMAGE ENCRYPTION

The procedure of image encryption is summarized below, where $C = 3$ is assumed in this study (see Fig. 4).

- Step 1: Divide a query image $x^{(u,i)} \in \mathbb{R}^{H \times W \times C}$ into N non-overlapped blocks of size $M \times M$, resulting in a set of blocks $B = \{B_1, \dots, B_j, \dots, B_N\}$, where block $B_j \in \mathbb{R}^{M \times M \times C}$ and $M = P$ as in [22].
- Step 2: Permute the blocks using a key K_b , resulting in permuted blocks $\hat{B} = \{\hat{B}_1, \dots, \hat{B}_j, \dots, \hat{B}_N\}$ as in [22].
- Step 3: Divide each block \hat{B}_j into N non-overlapped sub-blocks of size $M_s \times M_s$, resulting in a set $\hat{B}_j = \{\hat{B}_{j,1}, \dots, \hat{B}_{j,k}, \dots, \hat{B}_{j,N_s}\}$, where block $\hat{B}_j \in \mathbb{R}^{M_s \times M_s \times C}$. Then, split each sub-block $\hat{B}_{j,k}$ into three color component sub-blocks $\{\hat{B}_{j,k}^{c(1)}, \hat{B}_{j,k}^{c(2)}, \hat{B}_{j,k}^{c(3)}\}$, where $\hat{B}_{j,k}^{c(i)} \in \mathbb{R}^{M_s \times M_s \times 1}$. Let $c(1)$, $c(2)$, and


FIGURE 4. Image encryption procedure ($N_s = 4$).

FIGURE 5. Example of encrypted images from ImageNet [26].

$c(3)$ denote R, G, and B channels, respectively as in [22].

Step 4: Flatten each color component in sub-block $\hat{B}_{j,k}^{c(i)}$ into a vector $\hat{b}_{j,k}^{c(i)} \in \mathbb{R}^{M_s^2}$.

Step 5: Shuffle the pixel positions within each vector $\hat{b}_{j,k}^{c(i)}$ using an independent key $K_p^{(u,i)}$ and obtain a shuffled vector $b'_{j,k}{}^{c(i)} \in \mathbb{R}^{M_s^2}$.

Step 6: Reshape shuffled vector $b'_{j,k}{}^{c(i)}$ into $B'_{j,k}{}^{c(i)} \in \mathbb{R}^{M_s \times M_s}$. Then, reshape $B'_{j,k}{}^{c(1)}$, $B'_{j,k}{}^{c(2)}$, and $B'_{j,k}{}^{c(3)}$ into sub-block $B'_{j,k}$, yielding a set $B'_j = \{B'_{j,1}, \dots, B'_{j,k}, \dots, B'_{j,N_s}\}$.

Step 7: Concatenate all the reshaped sub-blocks $B'_{j,k}$ to form an encrypted block $B'_j \in \mathbb{R}^{M \times M \times C}$ as in [22].

Step 8: Concatenate all the reshaped blocks B'_j to form an encrypted image $x'^{(u,i)} \in \mathbb{R}^{H \times W \times C}$ as in [22].

The above procedure is also applied to training images. Note that key $K^{(u,i)}$ consists of K_b used for block scrambling and the independent key $K_p^{(u,i)}$ used for pixel shuffling. That is $K^{(u,i)} = [K_b, K_p^{(u,i)}]$.

Fig. 5 shows an example of encrypted images from ImageNet [26], produced by applying only block scrambling, only pixel shuffling, and both operations. By combining these two operations, the visual confidentiality of images is further enhanced. Additionally, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) values are shown below each encrypted image. These values quantitatively demonstrate that the combination of the two operations enhances encryption strength.

To clarify the encryption procedure, we give details on block scrambling and pixel shuffling below.

1) BLOCK SCRAMBLING

Block scrambling operations from step 2 to step 3 are carried out in accordance with the following steps [22], where key K_b is shared with the model creator and all clients:

- 1: Define an integer vector $l = [1, 2, \dots, N]^T$ to represent the original order of the blocks.
- 2: Prepare a key K_b as

$$K_b = [K_b(1), K_b(2), \dots, K_b(m), \dots, K_b(N)], \quad (2)$$

where

$$K_b(m) \in \{1, \dots, N\}, \quad K_b(m) \neq K_b(m') \quad \text{if } m \neq m', \\ m, m' \in \{1, \dots, N\}.$$

3: Define $E_{bs(m,n)}$ as

$$E_{bs(m,n)} = \begin{cases} 0 & (n \neq K_b(m)), \\ 1 & (n = K_b(m)). \end{cases} \quad (3)$$

4: Define a permutation matrix $\mathbf{E}_{bs} \in \mathbb{R}^{N \times N}$ as

$$\mathbf{E}_{bs} = \begin{pmatrix} E_{bs(1,1)} & E_{bs(1,2)} & \dots & E_{bs(1,N)} \\ E_{bs(2,1)} & E_{bs(2,2)} & \dots & E_{bs(2,N)} \\ \vdots & \vdots & \ddots & \vdots \\ E_{bs(N,1)} & E_{bs(N,2)} & \dots & E_{bs(N,N)} \end{pmatrix}. \quad (4)$$

5: The permuted vector \hat{l} is given by

$$\hat{l} = \mathbf{E}_{bs} l = [\hat{l}(1), \dots, \hat{l}(N)]^\top. \quad (5)$$

6: Permute the block accordingly:

$$\hat{B}_m = B_{\hat{l}(m)}. \quad (6)$$

2) PIXEL SHUFFLING

The pixel shuffling operations from step 4 to step 6 are described below. This pixel shuffling is inspired by the conventional method [22].

1: Convert $\hat{B}_{j,k}^{c(i)}$ with M_s^2 pixels into a vector $\hat{b}_{j,k}^{c(i)}$. Each vector is given by

$$\begin{cases} \hat{b}_{j,k}^{c(1)} = [\hat{b}_{j,k}^{c(1)}(1), \dots, \hat{b}_{j,k}^{c(1)}(M_s^2)], \\ \hat{b}_{j,k}^{c(2)} = [\hat{b}_{j,k}^{c(2)}(1), \dots, \hat{b}_{j,k}^{c(2)}(M_s^2)], \\ \hat{b}_{j,k}^{c(3)} = [\hat{b}_{j,k}^{c(3)}(1), \dots, \hat{b}_{j,k}^{c(3)}(M_s^2)]. \end{cases} \quad (7)$$

2: Generate a random integer sequence $K_p^{(u,i)}$ as an independent key

$$K_p^{(u,i)} = [K_p^{(u,i)}(1), \dots, K_p^{(u,i)}(M_s^2)], \quad (8)$$

where

$$K_p^{(u,i)}(m) \in \{1, \dots, M_s^2\}, \\ K_p^{(u,i)}(m) \neq K_p^{(u,i)}(m') \quad \text{if } m \neq m', \\ m, m' \in \{1, \dots, M_s^2\}.$$

3: Define $E_{ps(m,n)}$ as

$$E_{ps(m,n)} = \begin{cases} 0 & (n \neq K_p^{(u,i)}(m)), \\ 1 & (n = K_p^{(u,i)}(m)) \end{cases}. \quad (9)$$

4: Define the permutation matrix $\mathbf{E}_{ps} \in \mathbb{R}^{M_s^2 \times M_s^2}$ as

$$\mathbf{E}_{ps} = \begin{pmatrix} E_{ps(1,1)} & E_{ps(1,2)} & \dots & E_{ps(1,M_s^2)} \\ E_{ps(2,1)} & E_{ps(2,2)} & \dots & E_{ps(2,M_s^2)} \\ \vdots & \vdots & \ddots & \vdots \\ E_{ps(M_s^2,1)} & E_{ps(M_s^2,2)} & \dots & E_{ps(M_s^2,M_s^2)} \end{pmatrix}. \quad (10)$$

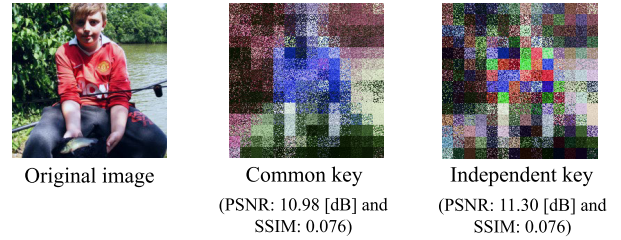


FIGURE 6. Example of pixel-shuffled images from ImageNet [26].

5: The permuted vector $\tilde{b}_{j,k}^{c(i)}$ is also given by

$$\begin{cases} \tilde{b}_{j,k}^{c(1)} = \hat{b}_{j,k}^{c(1)} \mathbf{E}_{ps} = [\tilde{b}_{j,k}^{c(1)}(1), \dots, \tilde{b}_{j,k}^{c(1)}(M_s^2)], \\ \tilde{b}_{j,k}^{c(2)} = \hat{b}_{j,k}^{c(2)} \mathbf{E}_{ps} = [\tilde{b}_{j,k}^{c(2)}(1), \dots, \tilde{b}_{j,k}^{c(2)}(M_s^2)], \\ \tilde{b}_{j,k}^{c(3)} = \hat{b}_{j,k}^{c(3)} \mathbf{E}_{ps} = [\tilde{b}_{j,k}^{c(3)}(1), \dots, \tilde{b}_{j,k}^{c(3)}(M_s^2)], \end{cases} \quad (11)$$

where

$$\tilde{b}_{j,k}^{c(1)}(m) \in \{\hat{b}_{j,k}^{c(1)}(1), \dots, \hat{b}_{j,k}^{c(1)}(M_s^2)\}, \\ \tilde{b}_{j,k}^{c(2)}(m) \in \{\hat{b}_{j,k}^{c(2)}(1), \dots, \hat{b}_{j,k}^{c(2)}(M_s^2)\}, \\ \tilde{b}_{j,k}^{c(3)}(m) \in \{\hat{b}_{j,k}^{c(3)}(1), \dots, \hat{b}_{j,k}^{c(3)}(M_s^2)\}.$$

6: Randomly permute $c(1), c(2)$, and $c(3)$ to obtain $c'(1), c'(2)$, and $c'(3)$ where $c'(i) \in \{c(1), c(2), c(3)\}$.

7: Obtain final vectors as

$$\begin{cases} b_{j,k}^{c'(1)} = \tilde{b}_{j,k}^{c'(1)}, \\ b_{j,k}^{c'(2)} = \tilde{b}_{j,k}^{c'(2)}, \\ b_{j,k}^{c'(3)} = \tilde{b}_{j,k}^{c'(3)}. \end{cases} \quad (12)$$

The above image encryption allows us not only to assign different keys to each client but to also apply different keys to each image. In addition, keys used for pixel shuffling do not need to be common among blocks in every image. Fig. 6 shows an example of encrypted images generated with a common key and with independent keys, respectively. As shown in the figure, for images encrypted with a common key, many blocks have similar colors, whereas for those encrypted with independent keys, blocks have more diverse colors. This property generally enhances visual information protection. In addition, images encrypted with independent keys have different properties as learnable encrypted images from those with a common key in addition to the difference in visibility.

In summary, the proposed method eliminates the need for key management and, at the same time, can generate encrypted images with diverse colors.

D. MODEL ENCRYPTION

To reduce the influence of image encryption prior to the fine-tuning of models, the domain adaptation is carried out in accordance with the embedding structure of ViT [22]. This transformation partially follows the operation introduced

in [22]. \mathbf{E}_{bs} in (5) is used to permute blocks in an image, so it has a close relationship with position embedding \mathbf{E}_{pos} in (1), where \mathbf{E}_{pos} includes the position information of a class token x_{class} . In contrast, \mathbf{E}_{bs} does not consider the information of x_{class} . To fill the gap between \mathbf{E}_{pos} and \mathbf{E}_{bs} , \mathbf{E}'_{bs} is extended as

$$\mathbf{E}'_{bs} = \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{E}_{bs} \end{pmatrix} \in \mathbb{R}^{(N+1) \times (N+1)}. \quad (13)$$

\mathbf{E}_{pos} is then encrypted by using \mathbf{E}'_{bs} as

$$\mathbf{E}'_{pos} = \mathbf{E}'_{bs} \mathbf{E}_{pos}. \quad (14)$$

Therefore, a sequence of the embedded patches \hat{z}_0 is represented as follows:

$$\hat{z}_0 = \mathbf{E}'_{bs}[x_{class}; x_p^1 \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}'_{pos}. \quad (15)$$

By substituting (14) into \mathbf{E}'_{pos} in (15), \hat{z}_0 is also given by

$$\begin{aligned} \hat{z}_0 &= \mathbf{E}'_{bs}[x_{class}; x_p^1 \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}'_{bs} \mathbf{E}_{pos} \\ &= \mathbf{E}'_{bs}([x_{class}; x_p^1 \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}_{pos}). \end{aligned} \quad (16)$$

As shown in the above equation in [22], the encryption preserves both the order of patches in the input to the original encoder and the associated positional information. In particular, part of (16) is the same as that of (1). Therefore, even when encrypted images are input as query ones, the model can process them in the same manner as a plain ViT processes plain images. This preservation helps mitigate the impact of encryption on classification performance.

The model parameters \mathbf{E} and the encrypted \mathbf{E}'_{pos} are fine-tuned using training images encrypted with different encryption keys. Through this process, both \mathbf{E} and \mathbf{E}'_{pos} are updated, enabling the model to correctly classify query images encrypted with different keys.

E. REQUIREMENTS FOR PROPOSED METHOD

Our method aims to satisfy the following requirements.

- Independence of key for each client: each client should have an independent key, respectively.
- Model capability: Privacy-preserving methods for DNNs should not decrease the model capability severely. A classifier trained with images encrypted by the proposed method is required to maintain an approximate accuracy as when using plain images.
- Security: Any perceptual information of plain images should not be reconstructed from images encrypted by the proposed method unless the key is exposed. In addition, even if the key is exposed, the damage should be small.
- Easy key management: Keys used for encryption should be easy to manage.

To satisfy the above requirements, models are trained using images encrypted with independent keys in the same manner as the query images.

TABLE 1. Image encryption conditions.

dataset	CIFAR-10 [27], Tiny ImageNet [28] (30 selected classes)
# of class	10 (CIFAR-10), 30 (Tiny ImageNet)
Image size	224 × 224
Block size M	16
Sub-block size M_s	16, 8

TABLE 2. Training settings for CIFAR-10 and Tiny ImageNet experiments.

Setting	CIFAR-10	Tiny ImageNet
Pre-trained model	vit_base_patch16_224 [29]	
Batch size	256	
Optimizer	AdamW [30]	
Learning rate	$1e^{-5}$	$1e^{-4}$
LR Scheduler	Cosine decay [31]	
Weight decay	0.01	0.03
Drop path rate	–	0.2
# of epochs	30	50
Warm-up epochs	–	10
Loss function	Cross-entropy	

IV. EXPERIMENTS

In experiments, we evaluated the proposed method from the perspective of classification accuracy and attack resistance.

A. EXPERIMENTAL SETUP

The experiments were conducted using an Intel Xeon W5-3435X CPU and an NVIDIA RTX 6000 Ada Generation 48 GB GPU.

1) DATASET

In Table 1, encryption conditions are summarized. Experiments were conducted on the CIFAR-10 [27] and Tiny ImageNet [28] datasets. CIFAR-10 comprises 60,000 images with 10 classes (6,000 images for each class), with 50,000 images for fine-tuning and 10,000 images for testing. Tiny ImageNet contains 200 classes of images with 64×64 pixels. We selected 30 classes from the Tiny ImageNet dataset (15,000 images for training and 1,500 images for testing). All images were resized to $224 \times 224 \times 3$ pixels to fit the input to ViT.

2) NETWORK

We used a pre-trained ViT with a patch size of $P = 16$, which was prepared in [29], where it was pre-trained with ImageNet-21k. ImageNet-21k is a dataset consisting of 21,000 classes with a total of 14 million images, which were resized to an image size of $224 \times 224 \times 3$ when pre-training ViT.

TABLE 3. Encryption setting comparison. “BS” and “PS” denote block scrambling and pixel shuffling, respectively. ✓: Yes, ×: No, -: Not applicable.

Method	BS key sharing	PS type	PS key sharing
Baseline	–	–	–
Conventional 1	✓	across color channels	Training ✓ / Query ✓
Conventional 2	✓	across color channels	Training ✓ / Query ×
Proposed	✓	each color channel	Training × / Query ×

Table 2 shows the training settings for each dataset. For CIFAR-10, no warm-up was applied; an initial learning rate was set to $1e^{-5}$ and decayed to 0 over 30 epochs using a cosine decay. For Tiny ImageNet, a learning rate was linearly increased from $1e^{-6}$ to $1e^{-4}$ during the first 10 warm-up epochs and then decayed to 0 following a cosine decay.

B. IMAGE CLASSIFICATION ACCURACY

The effectiveness of the proposed method was verified on CIFAR-10 and Tiny ImageNet in terms of image classification accuracy. Three kinds of models were compared as shown in Table 3 where “Baseline” means that the pre-trained model was fine-tuned with plain images and query images were also plain. In “Conventional 1,” both the pre-trained model and all training/query images were encrypted with a single key. For “Conventional 2,” the pre-trained model was encrypted with a single key, and the encrypted pre-trained model was fine-tuned by using training images encrypted with a common key. Query images were encrypted with independent keys. For “Proposed,” the pre-trained model was encrypted with a key K_b , and training images were encrypted with independent keys for each image as described in III-C.

Table 4 shows the experimental result of classification accuracy. For Conventional 1, a slight decrease in accuracy was observed compared with Baseline, and the degree of decrease varied depending on the dataset. In contrast, for Conventional 2, a significant drop in accuracy was observed when using query images encrypted with independent keys. Thus, conventional methods are unsuitable in multi-client settings. In contrast, the proposed method still maintained high accuracy. It significantly suppressed accuracy degradation from the original image usage stage and maintained practical performance even in multi-client environments. In addition, when using a smaller M_s , the accuracy was better. Encryption with independent keys exhibited much higher accuracy than encryption with shared keys.

Furthermore, in encryption with independent keys, a comparison between configurations with and without sub-block division shows that the use of sub-blocks resulted in higher classification accuracy. The improvement in classification accuracy with sub-block division was attributed to the restriction of pixel movement within localized regions.

To exclude the effect of pre-training, we also evaluated the proposed method when the model was trained from scratch.

TABLE 4. Classification accuracy [%] on CIFAR-10 and Tiny ImageNet ($M = 16$) where query images were encrypted with common key in Conventional 1, and query images were encrypted with independent keys in Conventional 2. Models used in conventional methods were fine-tuned with common key. Models used in proposed method were fine-tuned with independent keys.

Method	CIFAR-10	Tiny ImageNet
Baseline	98.41	96.87
Conventional 1	95.87	93.47
Conventional 2 (with independent keys)	13.96	6.33
Proposed ($M_s = 16$)	93.21	86.13
Proposed ($M_s = 8$)	96.33	89.13

TABLE 5. Classification accuracy [%] on CIFAR-10 and Tiny ImageNet ($M = 16$) trained from scratch where query images were encrypted with common key in Conventional 1, and query images were encrypted with independent keys in Conventional 2. Models used in conventional methods were fine-tuned with common key. Models used in proposed method were fine-tuned with independent keys.

Method	CIFAR-10	Tiny ImageNet
Baseline	73.99	43.87
Conventional 1	69.42	42.80
Conventional 2 (with independent keys)	11.75	4.47
Proposed ($M_s = 8$)	65.39	36.53

For CIFAR-10, the number of epochs was set to 50, the weight decay to 0.03, the learning rate to $1e^{-4}$, and the drop path rate to 0.03. For Tiny ImageNet, the number of epochs was set to 70, the weight decay to 0.05, the learning rate to $5e^{-6}$, and the drop path rate to 0.1. As shown in Table 5, similar trends were observed compared with the case using pre-trained models. These results show that the proposed method is also effective on models trained from scratch.

C. SECURITY ANALYSIS

Here, we analyze the security strength of the proposed image encryption method.

1) KEY SPACE

It is assumed that the attacker performs only ciphertext-only attacks on encrypted images in this paper, so we focus on brute-force attacks, in which the size of key space plays an important role.

The key space is the set of all possible keys that can be used with an encryption algorithm. The size of the key space is a crucial factor in determining the strength of an encryption algorithm, with larger key spaces generally offering greater resistance to brute-force attacks.

TABLE 6. Key space of proposed method, where $H \times W \times C = 224 \times 224 \times 3$ and $N = 196$. Note that M_s , BS, and PS represent sub-blocks size, block scrambling, and pixel shuffling, respectively.

M_s	Key assignment	Key space of BS	Key space of PS	Key space
16	Common	$14!$	$256! \times 3!$	$O_{16}^{com} = 14! \times 256! \times 3! \gg 2^{256}$
16	Independent	$14!$	$(256! \times 3!)^{14}$	$O_{16}^{ind} = 14! \times (256! \times 3!)^{14} \gg O_{16}^{com} \gg 2^{256}$
8	Common	$14!$	$64! \times 3!$	$O_8^{com} = 14! \times 64! \times 3! \gg 2^{256}$
8	Independent	$14!$	$(64! \times 3!)^{4 \times 14}$	$O_8^{ind} = 14! \times (64! \times 3!)^{4 \times 14} \gg O_8^{com} \gg 2^{256}$

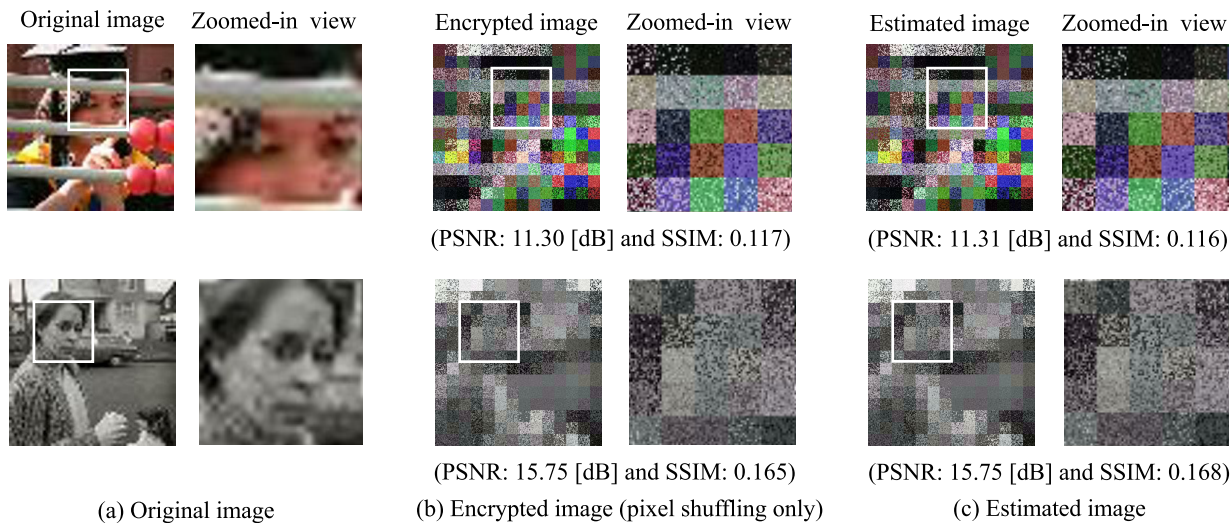


FIGURE 7. Estimated images by jigsaw puzzle solver attack on Tiny ImageNet. Zoomed-in views of boxed regions are shown on right of each image.

In the proposed method, N blocks are permuted by using block scrambling. Therefore, the key space of block scrambling O_{bs} is given by

$$O_{bs} = N!. \tag{17}$$

Additionally, in pixel shuffling, $M_s \times M_s$ pixels in each sub-block are shuffled. Then, three color components of these pixels are permuted. In the proposed method, pixel shuffling is applied to each sub-block using either a common key or independent keys. The key space of pixel shuffling with a common key O_{ps}^{com} is represented as follows:

$$O_{ps}^{com} = M_s^2! \times 3!. \tag{18}$$

On the other hand, the key space of pixel shuffling with independent keys O_{ps}^{ind} is given by

$$O_{ps}^{ind} = (M_s^2! \times 3!)^{N_s N}. \tag{19}$$

From the above equations, the overall key space O^{com} when using a common key is given by

$$O^{com} = O_{bs} \times O_{ps}^{com} = N! \times M_s^2! \times 3!. \tag{20}$$

In contrast, the key space O^{ind} when using independent keys is as follows:

$$O^{ind} = O_{bs} \times O_{ps}^{ind} = N! \times (M_s^2! \times 3!)^{N_s N}. \tag{21}$$

Table 6 exhibits the key space for each case. Thus, the proposed method provides a larger key space than that of the conventional one due to using independent keys.

In the experiments, images with a size of $224 \times 224 \times 3$ were divided into block images with a size of $16 \times 16 \times 3$, and the block images were then divided into sub-block images with a size of $M_s = 16 \times 16 \times 3$ or $8 \times 8 \times 3$. Table 6 shows the key space for each case. These values were significantly larger than 2^{256} , indicating that the proposed method provides a sufficiently large key space. Even if the shared key K_b is leaked, the key space of pixel shuffling with independent keys alone exceeds 2^{256} in all cases. Moreover, applying independent keys to each block significantly expands the total key space, regardless of the sub-block size.

2) CRYPTANALYTIC ATTACKS

Jigsaw puzzle solver attacks, which aim to restore visual information of images by exploiting the correlation among blocks or pixels in images encrypted with block-wise encryption [19], [20], [21], [22], [23], are a class of the state-of-the-art attacks [4], [32], [33]. In the proposed method, key K_b used for block scrambling is shared among all clients. Thus, an attacker may attempt to restore visual information from encrypted images by using K_b . Therefore, we applied a position restoration attack [33] to pixel-shuffled images. As shown in Fig. 7, all estimated images were similar to the images encrypted with pixel shuffling, so the jigsaw

TABLE 7. Quantitative evaluation of estimated images for CIFAR-10 by jigsaw puzzle solver attacks, where results are reported as mean value \pm standard deviation.

M_s	Key assignment	PSNR [dB]	SSIM
16	Common	22.33 ± 5.37	0.764 ± 0.094
16	Independent	17.21 ± 2.53	0.338 ± 0.110
8	Common	24.70 ± 7.43	0.881 ± 0.070
8	Independent	19.41 ± 3.43	0.439 ± 0.125

puzzle solver attack could not be effective in restoring the images encrypted with independent keys. In both the encrypted and estimated images, the sensitive information of the original images is successfully hidden. In addition, the PSNR and SSIM values relative to the original images are shown below each encrypted and estimated image in Fig. 7. Because the values for the estimated images are low, the attack was unsuccessful. To further evaluate the estimated images on each dataset, we randomly sampled 100 images from CIFAR-10 and 300 images from Tiny ImageNet, and we measured the PSNR and SSIM values of the estimated images with respect to the original ones. The results are summarized in Tables 7 and 8, where values are reported as mean value \pm standard deviation. These results demonstrate that the proposed method is robust against jigsaw puzzle solver attacks. In addition, since each image is encrypted with a different key and no key management is required, conventional attacks such as known-plaintext and chosen-plaintext attacks are not applicable because such attacks aim to estimate the key.

It is important to consider resistance against attacks that leverage neural networks. Several conventional encryption methods with block-wise perceptual encryption were verified to be robust against state-of-the-art attacks including generative model-based attacks [34], [35]. The proposed method can apply independent keys for image encryption, so it has stronger resistance against attacks that leverage neural networks than the conventional ones.

D. DISCUSSION

We discuss the proposed method in terms of computational cost, the relationship between security and accuracy, comparison with other encryption methods, and scalability to other datasets and tasks.

1) COMPUTATIONAL COST

The proposed method is implemented using perceptual encryption. Perceptual encryption allows us not only to generate learnable encrypted images but also to have low computational cost. Operations such as block scrambling and pixel shuffling are extremely lightweight; in our implementation, the average processing time for encrypting a color image with a size of 224×224 , computed over 200 images, was 0.0342 seconds. In contrast, the average training time

TABLE 8. Quantitative evaluation of estimated images for Tiny ImageNet by jigsaw puzzle solver attacks, where results are reported as mean value \pm standard deviation.

M_s	Key assignment	PSNR [dB]	SSIM
16	Common	17.17 ± 4.51	0.471 ± 0.222
16	Independent	14.53 ± 2.62	0.205 ± 0.106
8	Common	20.00 ± 5.70	0.740 ± 0.145
8	Independent	16.09 ± 2.97	0.288 ± 0.110

per epoch over 30 epochs was 293 seconds, and the average inference time was 37 seconds in the case of CIFAR-10. In other words, the computational cost of the encryption operation was about 1.2×10^{-4} times that of the training time and 9.2×10^{-4} times that of the inference time. Thus, the encryption cost is negligible compared with model training and inference, enabling the proposed method to serve as a lightweight framework for real-world applications.

2) TRADE-OFF BETWEEN SECURITY AND ACCURACY

The accuracy of our method depends on the size of sub-blocks M_s as shown in Table 4. The selection of a smaller block size provides higher accuracy. In contrast, the selection of a larger block size gives a large key space. Thus, there is a trade-off between key space and sub-block size.

Our method also allows us to assign independent keys to each block. This feature can enlarge the key space compared with a common key. Accordingly, even when selecting a small block size, a sufficiently large key space can be secured. In addition, the use of independent keys can minimize the damage caused by key leakage and key estimation.

3) COMPARISON WITH CONVENTIONAL ENCRYPTION METHODS

The proposed method suppressed the accuracy drop even in multi-client environments compared with classification using plain images. In contrast, conventional perceptual encryption methods suffered from significant accuracy degradation under the same setting. Moreover, the proposed method achieved accuracy comparable to conventional perceptual encryption with a single shared key. In addition, unlike conventional methods, the proposed scheme allows different keys to be applied per client or per image, which substantially reduces the risk of privacy violations in the event of a key leakage from a single client. In contrast to homomorphic encryption [10], [11], [12], [13], which does not allow users to freely assign keys, the proposed method also supports flexible key assignment per client, enhancing practicality. In addition, the encryption cost of the proposed method is very low compared with such encryption methods. Privacy-preserving federated learning [16], [20], [21] allows users to train a global model without centralizing the training data on one machine, but it cannot protect privacy during inference for test data when a model is deployed in an untrusted cloud server.

4) SCALABILITY TO OTHER DATASETS AND TASKS

In this paper, image classification tasks were carried out on the CIFAR-10 and Tiny ImageNet datasets. The influence of image encryption with a common key [10], [11], [12] has been verified on various datasets so far, and the accuracy was demonstrated to depend on the dataset. As shown in Table 4, for the CIFAR-10 dataset, which consists of low-resolution images, we achieved over 90% accuracy. In contrast, for the Tiny ImageNet dataset, the accuracy decreased compared with classification on plain images. The cause is thought to be the loss of detailed visual information caused by encryption. In particular, the effect tends to be stronger when independent keys are applied to high-resolution images. In contrast, as described above, image classification accuracy depends on the size of sub-blocks M_s , so selecting an appropriate M_s is important depending on the required accuracy.

Perceptual encryption has been applied to a variety of applications beyond image classification such as segmentation [36], model protection [37] and adversarial examples [7], [8], [38] by using images encrypted with a common key. Our method based on independent keys should also be applicable to such models. These subjects will be future work.

V. CONCLUSION

In this paper, we proposed a privacy-preserving image classification method based on perceptual encryption that eliminates the need for centralized key management. The method enables multiple clients to use a shared model with independently generated keys, while largely preserving classification accuracy. Through experiments, we demonstrated that the proposed method maintained high performance despite encryption and showed robustness against COAs. These results confirm both the practicality and effectiveness of the method for real-world multi-client environments.

Future work will focus on enhancing scalability to more challenging scenarios, such as datasets with higher resolutions, larger numbers of classes, or limited training samples. We also plan to extend the method to more complex tasks such as video analysis and to strengthen the security evaluation with broader attack models. These directions will further strengthen the practicality and widen the applicability of the proposed framework.

REFERENCES

- [1] Y. Liu, H. Chen, and Z. Yang, "Enforcing End-to-end security for remote conference applications," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2024, pp. 2630–2647.
- [2] J. Liu, J. Zhou, J. Tian, and W. Sun, "Recoverable privacy-preserving image classification through noise-like adversarial examples," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 20, no. 7, pp. 1–27, May 2024.
- [3] Q. Feng, P. Li, Z. Lu, C. Li, Z. Wang, Z. Liu, C. Duan, F. Huang, J. Weng, and P. S. Yu, "EViT: Privacy-preserving image retrieval via encrypted vision transformer in cloud computing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 8, pp. 7467–7483, Aug. 2024.
- [4] 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.
- [5] K. Madono, M. Tanaka, and M. Onishi, "ScrambleMix: A privacy-preserving image processing for edge-cloud machine learning," in *Proc. PSIVT*, Singapore, 2023, pp. 326–340.
- [6] H. Kiya, A. P. M. Maung, Y. Kinoshita, S. Imaizumi, and S. Shiota, "An overview of compressible and learnable image transformation with secret key and its applications," *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, 2022, Art. no. e11.
- [7] M. Maung, A. Pyone, and H. Kiya, "Encryption inspired adversarial defense for visual classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 1681–1685.
- [8] M. Aprilpyone and H. Kiya, "Block-wise image transformation with secret key for adversarially robust defense," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2709–2723, 2021.
- [9] R. Iijima, S. Shiota, and H. Kiya, "A random ensemble of encrypted vision transformers for adversarially robust defense," *IEEE Access*, vol. 12, pp. 69206–69216, 2024.
- [10] 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.
- [11] Y. Wang, J. Lin, and Z. Wang, "An efficient convolution core architecture for privacy-preserving deep learning," in *Proc. IEEE ISCAS*, May 2018, pp. 1–5.
- [12] N. M. Hijazi, M. Aloqaily, M. Guizani, B. Ouni, and F. Karray, "Secure federated learning with fully homomorphic encryption for IoT communications," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4289–4300, Feb. 2024.
- [13] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim, and J.-S. No, "Privacy-preserving machine learning with fully homomorphic encryption for deep neural network," *IEEE Access*, vol. 10, pp. 30039–30054, 2022.
- [14] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.
- [15] J. Konečný, H. Brendan McMahan, F. X. Yu, P. Richtárik, A. Theertha Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.
- [16] M. Tanaka, "Learnable image encryption," in *Proc. IEEE Int. Conf. Consum. Electronics-Taiwan (ICCE-TW)*, Taichung, Taiwan, May 2018, pp. 1–2.
- [17] W. Sirichotedumrong, Y. Kinoshita, and H. Kiya, "Pixel-based image encryption without key management for privacy-preserving deep neural networks," *IEEE Access*, vol. 7, pp. 177844–177855, 2019.
- [18] K. Madono, M. Tanaka, M. Onishi, and T. Ogawa, "Block-wise scrambled image recognition using adaptation network," 2020, *arXiv:2001.07761*.
- [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)*, Taipei, Taiwan, Sep. 2019, pp. 674–678.
- [20] H. Lin, S. Imaizumi, and H. Kiya, "Privacy-preserving ConvMixer without any accuracy degradation using compressible encrypted images," *Information*, vol. 15, no. 11, p. 723, Nov. 2024.
- [21] M. Aprilpyone and H. Kiya, "Privacy-preserving image classification using an isotropic network," *IEEE MultimediaMag.*, vol. 29, no. 2, pp. 23–33, Apr. 2022.
- [22] T. Nagamori, S. Shiota, and H. Kiya, "Efficient fine-tuning with domain adaptation for privacy-preserving vision transformer," *APSIPA Trans. Signal Inf. Process.*, vol. 13, no. 1, 2024, Art. no. e8.
- [23] H. Kiya, R. Iijima, A. Maungmaung, and Y. Kinoshita, "Image and model transformation with secret key for vision transformer," *IEICE Trans. Inf. Syst.*, vol. E106.D, no. 1, pp. 2–11, Jan. 2023.
- [24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, 2021, pp. 1–21.
- [25] A. Trockman and J. Zico Kolter, "Patches are all you need?" 2022, *arXiv:2201.09792*.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.
- [27] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [28] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

- [29] R. Wightman. *PyTorch Image Models*. Accessed: Jul. 20, 2025. [Online]. Available: <https://github.com/huggingface/pytorch-image-models>
- [30] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2017, *arXiv:1711.05101*.
- [31] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” 2016, *arXiv:1608.03983*.
- [32] D. Sholomon, O. E. David, and N. S. Netanyahu, “An automatic solver for very large jigsaw puzzles using genetic algorithms,” *Genetic Program. Evolvable Mach.*, vol. 17, no. 3, pp. 291–313, Sep. 2016.
- [33] T. Chuman and H. Kiya, “A jigsaw puzzle solver-based attack on image encryption using vision transformer for privacy-preserving DNNs,” *Information*, vol. 14, no. 6, p. 311, May 2023.
- [34] K. Madono, M. Tanaka, M. Onishi, and T. Ogawa, “SIA-GAN: Scrambling inversion attack using generative adversarial network,” *IEEE Access*, vol. 9, pp. 129385–129393, 2021.
- [35] A. P. Maung Maung, I. Echizen, and H. Kiya, “On the security of learnable image encryption for privacy-preserving deep learning,” *IEEE Access*, vol. 12, pp. 126415–126425, 2024.
- [36] H. Kiya, T. Nagamori, S. Imaizumi, and S. Shiota, “Privacy-preserving semantic segmentation using vision transformer,” *J. Imag.*, vol. 8, no. 9, p. 233, Aug. 2022.
- [37] A. Maungmaung and H. Kiya, “A protection method of trained CNN model with a secret key from unauthorized access,” *APSIPA Trans. Signal Inf. Process.*, vol. 10, no. 1, p. 10, 2021.
- [38] A. MaungMaung, I. Echizen, and H. Kiya, “Efficient key-based adversarial defense for ImageNet by using pre-trained models,” *IEEE Open J. Signal Process.*, vol. 5, pp. 902–913, 2024.



MARE HIROSE (Graduate Student Member, IEEE) received the B.E. and M.E. degrees from Chiba University, Japan, in 2024 and 2025, respectively, where she is currently pursuing the Ph.D. degree. Her research interests include multimedia security and machine learning.



SHOKO IMAIZUMI (Member, IEEE) received the B.Eng., M.Eng., and Ph.D. degrees from Tokyo Metropolitan University, Japan, in 2002, 2005, and 2011, respectively. From 2003 to 2004, she was with the Ministry of Education, Culture, Sports, Science and Technology of Japan. From 2005 to 2011, she was a Researcher with the Industrial Research Institute of Niigata Prefecture. In 2011, she joined Chiba University, where she is currently an Associate Professor with the Graduate

School of Informatics. Her research interests include image processing and multimedia security. She is a Senior Member of IEICE and a member of APSIPA, ITE, and IEEEJ. She serves as the Director for the Society of Photography and Imaging of Japan (SPIJ).



HITOSHI KIYA (Life Fellow, IEEE) received the B.E. and M.E. degrees from Nagaoka University of Technology, Japan, in 1980 and 1982, respectively, and the Dr.Eng. degree from Tokyo Metropolitan University, Japan, 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 currently a fellow of IEICE, AAIA, and ITE. He has received

numerous awards, including 12 best paper awards. He has organized a lot of international conferences in roles, such as the TPC Chair of IEEE ICASSP 2012 and the General Co-Chair of IEEE ISCAS 2019. He served as the President for APSIPA, from 2019 to 2020, and the Regional Director-at-Large for Region 10 of the IEEE Signal Processing Society, from 2016 to 2017.

...