

Block-Wise Image Transformation With Secret Key for Adversarially Robust Defense

Maungmaung Aprilpyone^{ID}, *Graduate Student Member, IEEE*, and Hitoshi Kiya^{ID}, *Fellow, IEEE*

Abstract—In this paper, we propose a novel defensive transformation that enables us to maintain a high classification accuracy under the use of both clean images and adversarial examples for adversarially robust defense. The proposed transformation is a block-wise preprocessing technique with a secret key to input images. The proposed defense obfuscates gradients in the absence of the secret key unlike previously defeated obfuscating defenses. We developed three algorithms to realize the proposed transformation: Pixel Shuffling, Bit Flipping, and FFX Encryption. Experiments were carried out on the CIFAR-10 and ImageNet datasets by using both black-box and white-box attacks with various metrics including adaptive ones. The results show that the proposed defense achieves high accuracy close to that of using clean images even under adaptive attacks for the first time. In the best-case scenario, a model trained by using images transformed by FFX Encryption (block size of 4) yielded an accuracy of 92.30% on clean images and 91.48% under PGD attack with a noise distance of 8/255, which is close to the non-robust accuracy (95.45%) for the CIFAR-10 dataset, and it yielded an accuracy of 72.18% on clean images and 71.43% under the same attack, which is also close to the standard accuracy (73.70%) for the ImageNet dataset. Overall, all three proposed algorithms are demonstrated to outperform state-of-the-art defenses including adversarial training whether or not a model is under attack.

Index Terms—Adversarial defense, image encryption, image classification.

I. INTRODUCTION

ALTHOUGH deep neural networks (DNNs) have led to major breakthroughs in computer vision, for a wide range of applications, where safety and security are critical, there is concern about their reliability. DNNs in general suffer from attacks such as model inversion attacks [1], membership inference attacks [2], and adversarial attacks [3]. In particular, carefully perturbed data points known as adversarial examples are indistinguishable from clean data points, but they cause DNNs to make erroneous predictions [3], [4]. As an example, in Fig. 1, the network here classified the clean image correctly as “tabby” with a 47.96% probability. After adding a small fraction of noise, the network misclassified the tabby cat as

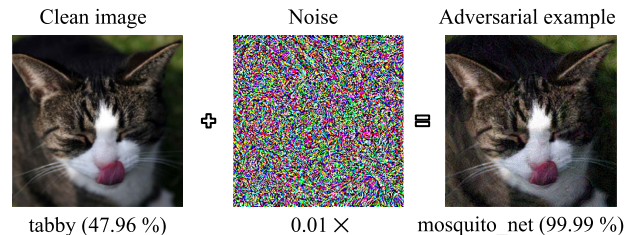


Fig. 1. Adversarial example.

“mosquito_net” with 99.99% confidence. Adversarial examples create a rising concern where DNNs are to be deployed in security-critical applications such as autonomous vehicles, speech recognition, natural language processing, and malware detection. Therefore, a lot of effort has been put towards adversarial robustness.

Researchers have proposed numerous ways of constructing adversarial examples. Such works include [3], [5]–[9], in which ℓ_p -bounded perturbation has been found. In the context of computer vision, these threat models do not match real world applications [10], [11] because there can be various physical conditions (e.g., camera angle, lighting/weather), physical limits on imperceptibility, etc. However, it has been proved that adversarial threats on neural networks remain real [12]–[16]. In addition, ℓ_p -bounded threat models are crucial for principled deep learning due to their well-defined nature [17]. They are helpful not only for evaluating the robustness of deep learning models but also for understanding them better. It is almost certain that models that are not robust against ℓ_p -bounded attacks will fail in real world scenarios.

With the development of adversarial attacks, numerous adversarial defenses have been proposed in the literature. To the best of our knowledge, there is no robust model that has a similar accuracy to a non-robust one. Some of the most reliable defenses are certified ones and adversarial training. However, certified defenses are not scalable, and the accuracy of adversarial training is not comparable to that of standard training. Alternatively, researchers have also come up with preprocessing approaches to improve the classification accuracy. Unfortunately, most of these approaches are broken by powerful adaptive attacks [18]. Therefore, finding ways to achieve high accuracy and adversarial robustness is a growing concern and an on-going area of research with a high demand for computer vision because of the wide range of applications.

More importantly, adversarial attacks and defenses have entered into an arms race in the literature. New defenses are

Manuscript received May 19, 2020; revised October 1, 2020, November 25, 2020, and January 18, 2021; accepted February 10, 2021. Date of publication March 1, 2021; date of current version April 1, 2021. This work was supported by the Japan Science and Technology Agency (JST), Core Research for Evolutional Science and Technology (CREST), Japan, under Grant JPMJCR20D3. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vanesa Daza. (*Corresponding author: Hitoshi Kiya.*)

The authors are with the Department of Computer Science, Tokyo Metropolitan University, Tokyo 191-0065, Japan (e-mail: april-pyone-maungmaung@ed.tmu.ac.jp; kiya@tmu.ac.jp).

Digital Object Identifier 10.1109/TIFS.2021.3062977

also broken by performing adaptive attacks [19]. Conventional adversarial defenses either reduce classification accuracy significantly or are completely broken. Therefore, in this work, we aim to achieve a high classification accuracy not only for clean examples but also for adversarial ones.

We propose a block-wise defensive transformation with a secret key that is inspired by perceptual image encryption techniques such as [20]–[26]. Modern deep convolutional neural networks such as ResNet are known to be sensitive to small image transformation [27]. Therefore, many researchers have been seeking learnable image encryption methods [22]–[25] that do not cause big drops in accuracy for privacy-preserving DNNs, but they have not considered robustness against adversarial examples. Deriving from such encryption methods, we develop three block-wise transformation algorithms to carry out the proposed transformation: Pixel Shuffling, Bit Flipping, and FFX Encryption. The proposed transformation is utilized to transform training/test images as a preprocessing technique, and a model is trained/tested by the transformed images. In addition, we also design adaptive attacks while accounting for obfuscated gradients [18] to evaluate models trained by the proposed transformation algorithms. As a result, the models trained by the proposed transformation make correct predictions for both clean images and adversarial examples. We make the following contributions in this paper.

- We apply extended perceptual image encryption techniques with a secret key to adversarial defenses for the first time.
- We develop three block-wise transformation algorithms: Pixel Shuffling, Bit Flipping, and FFX Encryption.
- We conduct extensive experiments on both black-box and white-box attacks including adaptive ones and present empirical results to show the effectiveness of the proposed defense.

In experiments, the proposed defense is confirmed to outperform state-of-the-art adversarial defenses. A part of this work (Pixel Shuffling) was introduced in [28]. We not only evaluate Pixel Shuffling under both black-box and white-box attacks with different metrics, and adaptive attacks with key estimation approaches but also introduce other novel algorithms in this paper.

The rest of this paper is structured as follows. Section II presents related work on adversarial attacks and defenses. Section III describes threat models. Regarding the proposed defense, Section IV includes notations, an overview, the three proposed block-wise transformation algorithms, the properties of block-wise transformations with keys, and a discussion on key management and robustness against adaptive attacks. Experiments on various attacks including adaptive ones are presented in Section V, and Section VI concludes this paper.

II. RELATED WORK

A. Adversarial Attacks

The goals of adversarial attacks on neural networks are confidence reduction, misclassification, and targeted misclassification. The attacks can be divided into two categories:

poisoning/causative attacks (i.e., training time attacks) and evasion/exploratory attacks (i.e., test time attacks) [29]. Poisoning attacks happen during training time, where an adversary introduces crafted malicious examples into training data to manipulate the behavior of models. Even one single poisonous image can compromise a model when transfer learning is used [30]. Evasion attacks are also called “adversarial examples,” in which crafted imperceptible perturbations are added. In this work, we focus on defending against evasion attacks.

Traditionally, evasion attacks are classified into three groups based on the knowledge of a particular model and training data available to the adversary: white-box, black-box, and gray-box. Under white-box settings, the adversary has direct access to the model, its parameters, training data, and defense mechanism. However, the adversary does not have any knowledge on the model, except the output of the model in black-box attacks. Between white-box and black-box methods, there are gray-box attacks that imply that the adversary knows something about the system (i.e., partial knowledge of the model such as its architecture, parameters, or training data).

Under white-box settings, given an input image x and a classifier $f_{\theta}(\cdot)$ parameterized by θ , an adversarial example x' is constructed such that $f_{\theta}(x') \neq y$, where y is a true class. This is done by minimizing the perturbation δ ,

$$\underset{\delta}{\text{minimize}} \|\delta\|_p, \quad \text{s.t. } f_{\theta}(x + \delta) \neq y, \quad (1)$$

or by maximizing the loss function,

$$\underset{\delta \in \Delta}{\text{maximize}} \mathcal{L}(f_{\theta}(x + \delta), y). \quad (2)$$

Usually, a typical threat model is bounded by an ℓ_p norm such that $\Delta = \{\delta : \|\delta\|_p \leq \epsilon\}$ for some perturbation distance $\epsilon > 0$.

One of the easy and popular ways of generating adversarial examples is the fast gradient sign method (FGSM) [5] under an ℓ_{∞} norm with a single gradient step. Its iterative version is the basic iterative method (BIM) [6]. BIM with multiple random restarts and initialization with uniform random noise is recognized as a projected gradient descent (PGD) [9] adversary. There are other iterative optimization-based attacks such as the Carlini and Wagner attack (CW) [8] for the ℓ_2 bounded metric and the elastic-net attack (EAD) [31] for the ℓ_1 bounded metric. CW finds the smallest noise under the ℓ_2 metric with a new loss function. CW is also a special case of EAD, where the ℓ_1 regularization parameter is set to zero [31]. In this work, we utilize three state-of-the-art attacks (PGD, CW, and EAD) to generate different sets of adversarial examples under ℓ_{∞} , ℓ_2 , and ℓ_1 metrics to evaluate the proposed defense.

Under black-box settings, the above white-box attacks can be applied via a substitute model. Several techniques have been proposed to improve transferability with this type of black-box attack [32]–[34]. Moreover, there are also gradient-free methods that estimate gradients such as [35]–[38]. Another recent black-box attack, NATTACK, learns a probability distribution centered around the input such that a sample drawn from that distribution is likely an adversarial example [39]. Additionally, the OnePixel attack constructs an adversarial example by modifying one or a few pixels without accessing the weights

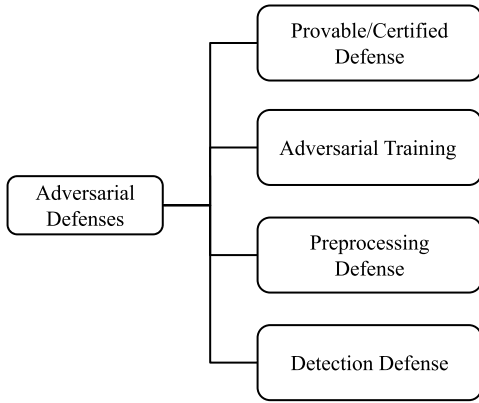


Fig. 2. Different approaches for adversarial defense.

of the model with differential evolution [40]. In this work, we employ the OnePixel attack [40], NATTACK [39], and one of the gradient estimation attacks, SPSA [37], to evaluate the proposed defense under black-box settings.

B. Adversarial Defenses

The goal of a defense method is to make a model that is accurate not only for clean input but also for adversarial examples. There are many different approaches to achieving this goal, such as certified and provable defenses, adversarial training, preprocessing techniques, and detection algorithms, as shown in Fig. 2.

Ideally, provable defenses are desired. Inspiring works such as [41]–[43] proposed provable secure training. Although these methods are attractive, they are not scalable. Some certified defenses have been scaled to a certain degree [44]–[47], but the accuracy is still not comparable to empirically robust models.

Current state-of-the-art empirically robust defenses are under the use of adversarial training. The earliest form of adversarial training is to inject FGSM-based adversarial noise into the training data [5]. Since FGSM is not iterative and not robust against iterative attacks such as PGD, FGSM-based training was found to be ineffective [6], [9]. Madry *et al.* proposed adversarial training with a PGD adversary, achieving the best empirical robustness to date [9]. However, PGD training is computationally expensive. To make the computation of adversarial training more feasible, “free” adversarial training was proposed in which gradients are computed with respect to the network parameters and the input image on the same backward pass [48]. In addition to “free” adversarial training, “fast” adversarial training was proposed and uses FGSM and standard efficient training tricks [49]. Although FGSM-based adversarial training was dismissed before, it is shown to be effective when random initialization is introduced [49]. Nevertheless, while adversarial training is repeatedly found to be robust against the best known adversaries [18], the accuracy is still very low compared with non-robust models.

Another approach to adversarial defenses is the use of preprocessing techniques. The works that take this direction utilize various ways of transformation such as thermometer encoding [50], image processing-based techniques [51], [52],

making small changes to pixels with the intent of removing adversarial noise [53], and GAN-based transformation [54]. These preprocessing defenses are appealing at first due to their higher accuracy. However, they have all been broken because these conventional preprocessing defenses rely on obfuscated gradients by [18]. Accounting for this problem, Raff *et al.* came up with a preprocessing defense that uses a number of random different transforms with random parameters [55]. Although their work claims majorly improved accuracy on ImageNet, applying many transforms for each image is computationally expensive and reduces the accuracy when the model is not under attack. In addition, one work enforces the use of 1-bit dithered images for training and testing a model [56]. However, typical cameras capture an image in 8-bit, and the use of 1-bit limits the range of application scenarios.

Moreover, instead of defending against adversarial examples directly, there are defenses to detect adversarial examples. Metzen *et al.* proposed a detection method that trains a binary classification network to distinguish clean data from adversarial examples [57]. Another work by [58] detects adversarial examples by looking at the features in the subspace of deep neural networks. However, it is reported that detection methods can also be bypassed [59].

All in all, adversarial defense approaches decrease the classification accuracy of a model. Even worse, most of the defenses, especially preprocessing-based methods, are defeated due to obfuscated gradients [18] and do not embed a secret key into the model inference process. Therefore, attaining robustness as well as high accuracy remains an open problem in adversarial defense research.

In this work, we approach adversarial defense in a different way by taking inspiration from perceptual image encryption techniques such as [20], [21], [23]–[25]. Perceptual image encryption techniques have never been applied before in this line of work. Similar to our work (Pixel Shuffling), Taran *et al.* first introduced a pixel shuffling approach (pixel-wise manner) with a secret key by using a standard random permutation [60]. Although their method [60] was effective to defend against adversarial examples, it was tested only on small datasets (MNIST [61] and F-MNIST [62]) and clean accuracy is significantly dropped on larger datasets such as CIFAR-10 [63] and ImageNet [64]. The reason is that shuffling in a pixel-wise manner loses spatial perceptual information. In contrast, the proposed algorithm (Pixel Shuffling) is block-wise pixel shuffling and designed to maintain a high clean accuracy. In this paper, we will show that the extension of perceptual image encryption techniques is effective in defending against adversarial examples.

III. THREAT MODELS

The goal of an adversarial defense is to keep the classification accuracy on both clean images and adversarial examples high. To evaluate a defense method, precisely defining threat models is necessary. A threat model includes a set of assumptions such as an adversary’s goals, knowledge, and capabilities [17]. We also define attack scenarios considering practical applications.

A. Adversary's Goals

An adversary can construct adversarial examples to achieve different goals when attacking a model: whether to reduce the performance accuracy (i.e., untargeted attacks) or to classify a targeted class (i.e., targeted attacks). Formally, untargeted attacks will cause a classifier f_θ to misclassify a true class y_{true} , given an adversarial example x' (i.e., $f_\theta(x') \neq y_{\text{true}}$), and targeted ones will force the classifier to a targeted label (i.e., $f_\theta(x') = y_{\text{targeted}}$). In this paper, we focus on untargeted attacks, although targeted attacks can be launched in a similar fashion.

B. Adversary's Knowledge

According to [17], the adversary's knowledge can be white-box (inner workings of the defense mechanism, complete knowledge on the model and its parameters), black-box (no knowledge on the model) and gray-box, that is, anything in between white-box and black-box. As in the field of cryptography, there can be a small amount of secret information even in white-box settings if the secret information must be easily replaceable and non-extractable [17]. For our proposed defense, we introduce a secret key with a transparent algorithm for the first time. The secret key can be replaced by retraining the model, and it cannot be extracted from the training data nor the model. The key is utilized to preprocess input on the fly just before the input goes into the model. In this work, we consider both white-box and black-box attacks while keeping a secret key.

C. Adversary's Capabilities

Depending on the requirements of different applications, a secret key may or may not be required for inference. However, it should be securely stored or distributed. We assume the adversary does not have access to information with respect to the secret key (either the key itself or model output with respect to the correct secret key). However, the adversary may guess/estimate the secret key and observe the model. Then, they can perform untargeted attacks in which small changes are made under different metrics (ℓ_0 , ℓ_1 , ℓ_2 , ℓ_∞) that change the true class of the input.

D. Attack Scenarios

We consider the following practical application scenarios.

Black-box: The attacker queries the protected model with their key and observes the output of the model. Specifically, the attacker performs three powerful black-box attacks: OnePixel [39], NATTACK [40], and SPSA [37].

White-box: In the proposed defense, the key is not a part of the model parameters. The model may be stolen in the case of sharing the model. We assume a scenario in which the model weights and the defense algorithm are available to the attacker. Since the defense algorithm is known, the attacker may carry out white-box attacks with their key. Specifically, the attacker carries out three strong white-box attacks: PGD [9], CW [8], and EAD [31]. To make the attacks more successful, we assume the attacker incorporates

the defense algorithm with an unknown key during the attacks. In other words, the white-box attacks are run on top of the defense algorithm with the attacker's key.

IV. PROPOSED DEFENSE

Image classification is the task of classifying an input image into a class category according to its visual content. The proposed defense targets robust predictions in the image classification task, which is a core problem in computer vision.

A. Notation

The following notations are utilized throughout this paper.

- w , h , and c are used to denote the width, height, and the number of channels of an image.
- The tensor $x \in [0, 1]^{c \times h \times w}$ represents an input color image.
- δ denotes adversarial noise.
- δ_a denotes adaptive adversarial noise.
- The tensor $x_t \in [0, 1]^{c \times h \times w}$ represents a transformed image.
- M is the block size of an image.
- Tensors $x_b, x'_b \in [0, 1]^{h_b \times w_b \times p_b}$ are a block image and a transformed block image, respectively, where $w_b = \frac{w}{M}$ is the number of blocks across width w , $h_b = \frac{h}{M}$ is the number of blocks across height h , and $p_b = M \times M \times c$ is the number of pixels in a block.
- A pixel value in a block image (x_b or x'_b) is denoted by $x_b(i, j, k)$ or $x'_b(i, j, k)$, where $i \in \{0, \dots, w_b - 1\}$, $j \in \{0, \dots, h_b - 1\}$, and $k \in \{0, \dots, p_b - 1\}$ are indices corresponding to the dimension of x_b or x'_b .
- B is a block of an image, and its dimension is $M \times M \times c$.
- \hat{B} is a flattened version of block B , and its dimension is $1 \times 1 \times p_b$.
- An encryption key is denoted by K .
- A password required for format-preserving encryption, which refers to encrypting in such a way that the output is in the same format as its input, is denoted as P .
- $\text{Enc}(n, P)$ denotes format-preserving Feistel-based encryption (FFX) [65] with a length of 3, where n is an integer (used only in FFX Encryption).
- A classifier with parameters θ is denoted as $f_\theta(\cdot)$.

B. Overview

We propose a general key-based adversarial defense that satisfies two requirements: defending against adversarial examples and maintaining a high classification accuracy. Assuming the key stays secret, an attacker will not obtain any useful information on the model, which will render the adversarial attack ineffective. The main idea of the proposed method is to embed a secret key into the model structure with minimal impact on model performance. To maintain a high classification accuracy, the proposed defense is designed in such a way that each block position in an input image is not changed.

Based on different types of key management, the proposed defense can be applied in two scenarios as shown in Fig. 3:

(1) Scenario A, where key K is saved with a provider, and (2) Scenario B, where key K is required by a provider for inference. As an example, Scenario A can be deployed in self-driving cars, and Scenario B can be utilized in vision application programming interfaces (APIs).

However, black-box attacks are possible in Scenario A because key K is saved at the provider and the attacker can get correct output with respect to the correct key K . In contrast, for Scenario B, the attacker has to perform black-box attacks with his key. In this situation, the attacker cannot get useful output about the correct secret key, and thus, the attack is not effective. To solve this issue under Scenario A, the proposed defense can be extended as a simple voting ensemble where users do not need to provide key K as mentioned in [66]. In this work, we particularly focus on Scenario B for black-box attacks where an attacker needs an assumed key to carry out the attacks (see Section III-D).

The proposed defense is a preprocessing technique that transforms an input image with a secret key in a block-wise manner. Both training and testing images are transformed with a secret key prior to training or testing by the provider. Generally, there are three parts to the proposed defensive transformation: block segmentation, block-wise transformation, and block integration (see Fig. 4). The process of the proposed transformation is shown as follows.

1) **Block Segmentation:** The process of block segmentation is illustrated in Fig. 5.

- An input image x is divided into blocks such that $\{B_{11}, B_{12}, \dots, B_{w_b h_b}\}$.
- Each block in x is flattened to obtain $\{\hat{B}_{11}, \hat{B}_{12}, \dots, \hat{B}_{w_b h_b}\}$.
- The flattened blocks are concatenated in such a way that the relative spatial location among blocks in x_b is the same as that among blocks in x .

2) **Block-wise Transformation:** Given a secret key K that is a seed for generating a pseudo random integer vector with a size of p_b , x_b is transformed by using a block-wise transformation algorithm, $t(x_b, K)$. The transformed block image is written as

$$x'_b = t(x_b, K). \quad (3)$$

3) **Block Integration:** The transformed blocks in x'_b are integrated back to the original dimension (i.e., $c \times h \times w$) in the reverse order to the block segmentation process for obtaining a transformed image x_t .

C. Three Proposed Block-Wise Transformations

We propose three block-wise transformation algorithms, Pixel Shuffling, Bit Flipping, and FFX Encryption, for realizing $t(x_b, K)$. A block-wise transformation takes a block image x_b and a key K and then outputs a transformed block image x'_b . In the case of FFX Encryption, there is an additional parameter, password P , for format-preserving encryption. Although one of the three algorithms (Pixel Shuffling) was discussed in our previous work [28], we extend the evaluation of Pixel Shuffling with various white-box and black-box attacks under

different metrics and adaptive attacks with key estimation in this paper.

Pixel Shuffling: There are two steps to pixel shuffling as described in Algorithm 1:

- 1) Generate a random permutation vector $v = (v_0, v_1, \dots, v_k, \dots, v_{k'}, \dots, v_{p_b-1})$ that consists of randomly permuted integers from 0 to $p_b - 1$ by using key K . Let $k, k' \in \{0, \dots, p_b - 1\}$ and $v_k \neq v_{k'}$ if $k \neq k'$.
- 2) Perform block-wise shuffling on the basis of v . Basically, positions of pixel values in each block are changed on the basis of v , i.e.,

$$x'_b(i, j, v_k) = x_b(i, j, k). \quad (4)$$

Algorithm 1 Pixel Shuffling

Input: x_b, K

Output: x'_b

Generate a random permutation vector v by K

$x'_b \leftarrow x_b[:, :, v]$

Bit Flipping: There are four steps to pixel intensity inversion as described in Algorithm 2:

- 1) Generate a random binary vector $r = (r_0, r_1, \dots, r_k, \dots, r_{p_b-1})$, $r_k \in \{0, 1\}$ by using key K . To keep the transformation consistent, r is distributed with 50% of “0”s and 50% of “1”s.
- 2) Convert every pixel value to be in 255 scale with 8 bits (i.e., multiply x_b by 255).
- 3) Perform block-wise negative/positive transformation on the basis of r . Basically, every pixel value in block \hat{B}_{ij} is applied to

$$x'_b(i, j, k) = \begin{cases} x_b(i, j, k) & (r_k = 0) \\ x_b(i, j, k) \oplus (2^L - 1) & (r_k = 1), \end{cases} \quad (5)$$

where L is the number of bits used in $x_b(i, j, k)$, and $L = 8$ is used in this paper.

- 4) Convert every pixel value back to $[0, 1]$ scale (i.e., divide x'_b by 255).

Algorithm 2 Bit Flipping

Input: x_b, K

Output: x'_b

Generate a random binary vector r by K

// Make pixel values be at 255 scale

$x_b \leftarrow x_b \cdot 255$

$x'_b[:, :, r] \leftarrow 255 - x_b[:, :, r]$

$x'_b \leftarrow x'_b / 255$

FFX Encryption: In Bit Flipping, there are only two possibilities: whether the intensity of a pixel value is inversed or not. In contrast, we replace Bit Flipping with a cryptographic property (i.e., FFX mode) to generate a unique pattern in a block-wise manner, where the number of patterns is much larger

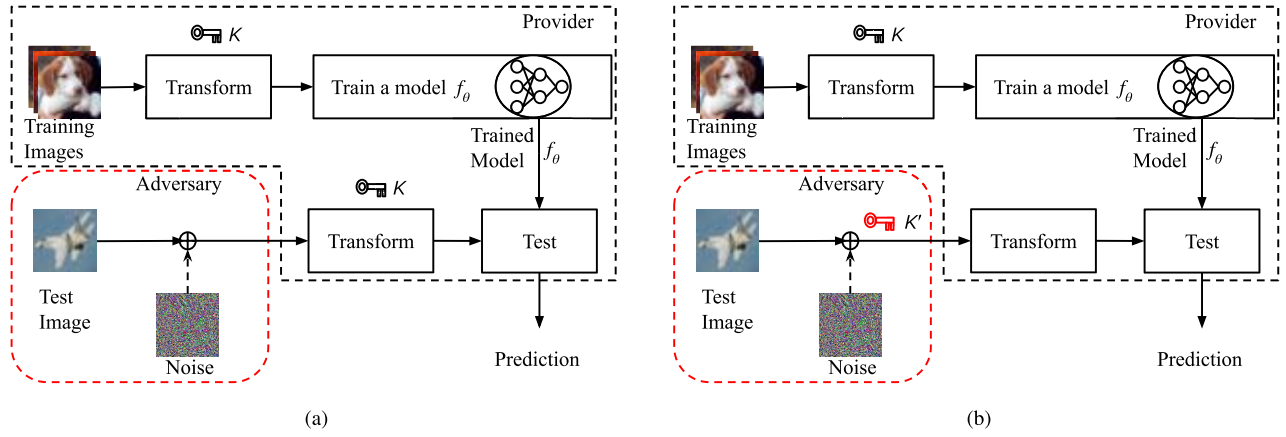


Fig. 3. Scenarios of image classification with proposed defense. (a) Scenario A where key K is saved with provider. (b) Scenario B where key K is required from user/adversary by provider for inference.

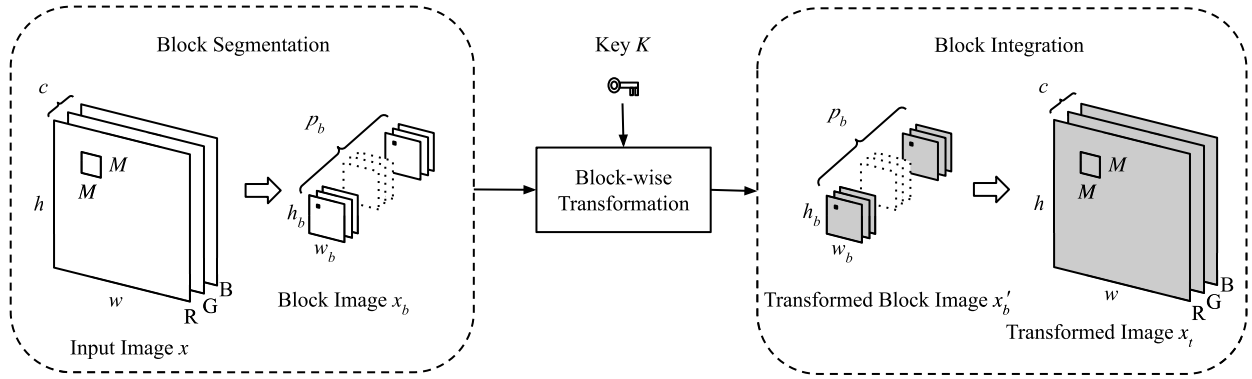


Fig. 4. Process of proposed transformation.

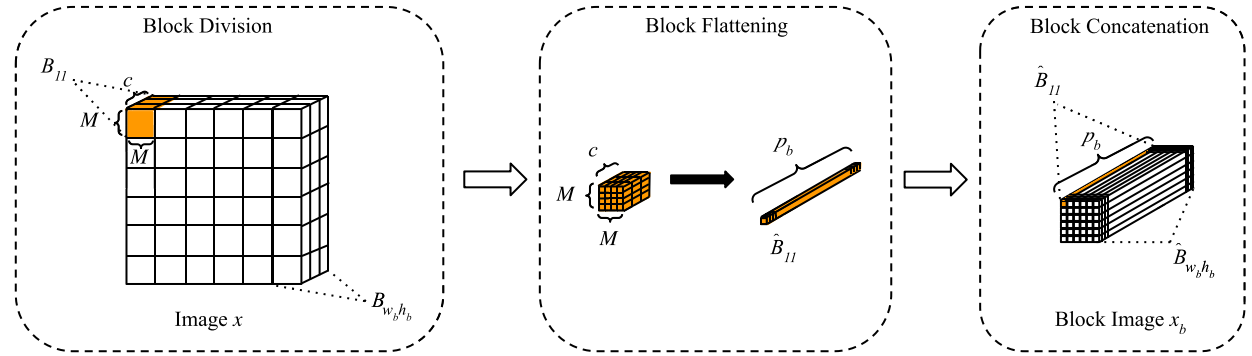


Fig. 5. Process of block segmentation.

than that of Bit Flipping. For this reason, FFX Encryption is applied to adversarial defense for the first time.

Apart from key K , FFX-based transformation also requires a password P for format-preserving Feistel-based encryption (FFX) [65]. The pixel value $x_b(i, j, k) \in \{0, 1, \dots, 254, 255\}$ is encrypted by FFX with a length of 3 digits to cover the whole range from 0 to 255. FFX randomly transforms each pixel with an integer value of (0–255) into a pixel with an integer value of (0–999) preserving the integer format. There are four steps to FFX-based transformation as described in Algorithm 3:

1) Generate a random binary vector $r = (r_0, r_1, \dots, r_k, \dots, r_{p_b-1})$, $r_k \in \{0, 1\}$ by using

key K . To keep the transformation consistent, r is distributed with 50% of “0”s and 50% of “1”s.

- 2) Convert every pixel value to be at 255 scale with 8 bits (i.e., multiply x_b by 255).
- 3) Perform block-wise FFX-based transformation on the basis of r and P . Basically, every pixel value in block \hat{B}_{ij} is applied to

$$x'_b(i, j, k) = \begin{cases} x_b(i, j, k) & (r_k = 0) \\ \text{Enc}(x_b(i, j, k), P) & (r_k = 1). \end{cases} \quad (6)$$

- 4) Convert every pixel value back to $[0, 1]$ scale (i.e., divide x'_b by the maximum value of x'_b).

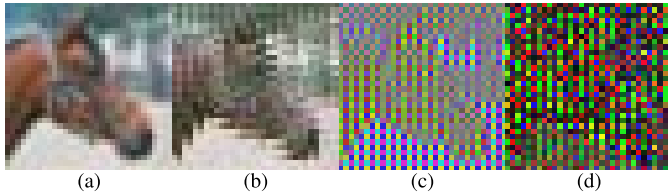


Fig. 6. Example of images generated by three proposed transformations with $M = 2$. (a) Original image. (b) Pixel Shuffling. (c) Bit Flipping. (d) FFX Encryption.

On a side note, only pixel values of 0 to 255 are encrypted once, and the block-wise transformation uses a lookup table. Therefore, the computational cost of encrypting 256 integer values in FFX mode is negligible and does not cause any significant overheads when training or testing a model.

Algorithm 3 FFX Encryption

Input: x_b, K, P

Output: x'_b

Generate a random binary vector r by K

// Make pixel values be at 255 scale

$x_b \leftarrow x_b \cdot 255$

$x'_b[:, :, r] \leftarrow \text{Enc}(x_b[:, :, r], P)$

$\max \leftarrow$ the maximum value of the encryption

$x'_b \leftarrow x'_b / \max$

D. Properties of Block-Wise Transformation With Key

A classifier model, $f_\theta(\cdot)$, trained by using transformed images is affected by both key K and the block-wise transformation used for transforming images. Each transformation algorithm creates a unique pattern as illustrated in Fig. 6, where a test image (“horse”) was transformed by the three proposed algorithms. The pattern created by the defensive transformation makes the gradients of the loss function with respect to the parameters unique to the particular transformation and the key, i.e.,

$$\nabla_\theta \mathcal{L}(f_\theta(x_t), y) \not\approx \nabla_\theta \mathcal{L}(f_\theta(x), y), \quad (7)$$

and

$$\nabla_\theta \mathcal{L}(f_\theta(t(x_b, K_1)), y) \not\approx \nabla_\theta \mathcal{L}(f_\theta(t(x_b, K_2)), y), \quad (8)$$

where K_1 and K_2 are different keys, and the symbol $\not\approx$ denotes approximately not equal to. Consequently, a model trained by the transformed images works well only when images are transformed under the use of the same transformation and key as those used for training the model. Therefore, the equations

$$f_\theta(x_t) \neq f_\theta(x) \quad (9)$$

and

$$f_\theta(t(x_b, K_1)) \neq f_\theta(t(x_b, K_2)) \quad (10)$$

are satisfied.

Another notable property of the proposed defense is the low computation cost. The block-wise operation utilized in the

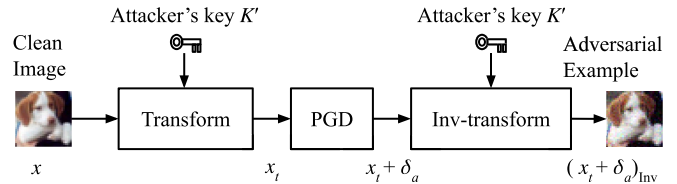


Fig. 7. Scenario of adaptive attack with estimated key.

proposed defense can be efficiently implemented by vectorized operations and is available for large-scale systems without any noticeable overheads during training/inference. Therefore, the proposed defense has potential for real-world applications including real-time ones.

E. Key Management and Robustness Against Adaptive Attacks

One of the properties of the proposed transformation is the use of a secret key. The key can be saved with a provider or can be required by the provider as a parameter for inference as shown in Fig. 3. Key management and robustness against adaptive attacks are discussed here to evaluate the effectiveness of the proposed defense.

As pointed out in [17], [19], adaptive attacks, which are adapted to the specific details of the proposed defense, are necessary in evaluating adversarial defenses to avoid a false sense of security. Optimization-based attack methods require correct gradients of the loss function with respect to the input. Therefore, many defenses make the gradients incorrect by introducing non-differentiable transformation or other obfuscation means such as randomization. These defenses that rely on obfuscated gradients are defeated by adaptive attacks [18]. One of the reasons adaptive attacks are successful is that useful gradients can be approximated because defensively transformed input is similar to the original input (i.e., $g(x) \approx x$, where $g(\cdot)$ is a defensive transform). In contrast, in the proposed defense, the input is transformed in a systematic way with a secret key, and the resulting input is not similar to the initial input (i.e., $t(x, K) \not\approx x$).

We carry out the following adaptive attacks to evaluate the proposed defense. In experiments, the proposed defense will be demonstrated to still maintain robustness against adaptive attacks.

1) *Inverse Transformation Attack:* An adversary may generate adversarial examples by adding noise to transformed images and inverse transform them with an assumed key. To simulate such an attack scenario, we designed an adaptive attack as shown in Fig. 7. Since key K is not available to the adversary, it has to be guessed randomly or heuristically for the adversary to carry out the adaptive attack. When an estimated key is close enough to the correct key, the adversary may be able to fool the model. However, we show that searching for a key close to key K is not easy.

2) *Estimation Over Transformation Attack:* The Estimation over Transformation Attack (EOT) is effective for estimating gradients in adversarial defenses with randomization as explained in [18]. Instead of taking one step in the direction of

gradients $\nabla_x f(x)$, we move in the direction of $\sum_{i=1}^{30} \nabla_x f(x)$. In other words, we use 30 keys to generate adversarial examples under a PGD attack.

3) *Transferability Attack*: Since the proposed defensive transformation method is transparent, an attacker can train a substitute model with their key. Then, the attacker generates adversarial examples over the substitute model. We simulate this attack scenario in experiments.

V. EXPERIMENTS

To verify the effectiveness of the proposed defense, we ran a number of experiments on different datasets. All the experiments were carried out in PyTorch platform.

A. Datasets

We used the CIFAR-10 [63] and ImageNet [64] datasets. CIFAR-10 consists of 60,000 color images (dimension of $32 \times 32 \times 3$) with 10 classes (6000 images for each class) where 50,000 images are for training and 10,000 for testing. We utilized a batch size of 128 and live augmentation (random cropping with a padding of 4 and random horizontal flip) on a training set.

ImageNet comprises 1.28 million color images for training and 50,000 color images for validation. We progressively resized images during training starting with larger batches of smaller images to smaller batches of larger images. We adapted three phases of training from the DAWNBench top submissions as mentioned in [49]. Phases 1 and 2 resized images to 160 and 352 pixels, respectively, and phase 3 used the entire image size from the training set. The augmentation methods used in the experiment were random resizing and cropping (sizes of 128, 224, and 288 respectively for each phase) and random horizontal flip.

B. Networks

We utilized deep residual networks [67] with 18 layers (ResNet18) for the CIFAR-10 dataset and trained for 200 epochs with efficient training techniques from the DAWNBench top submissions: cyclic learning rates [68] and mixed-precision training [69]. The parameters of the stochastic gradient descent (SGD) optimizer were a momentum of 0.9, weight decay of 0.0005, and maximum learning rate of 0.2. For ImageNet, we used ResNet50 with pre-trained weights. We adapted the training settings from [49] with the removal of weight decay regularization from batch normalization layers. The network was trained for 15 epochs in total for the ImageNet dataset.

C. Attack Settings

We utilized an attack library [70] for SPSA, PGD, CW, and EAD attacks, a publicly available implementation of the OnePixel attack, and code from the original authors for NATTACK.

Three black-box attacks, OnePixel, NATTACK, and SPSA, were deployed to evaluate the proposed defense. The OnePixel attack was configured for 10 pixels, 100 iterations, and a

population size of 400. For NATTACK, the population size was 300, the sigma was 0.1, the learning rate was 0.02, and 500 iterations were used for the CIFAR-10 dataset, and population sizes of 200 and 200 iterations were used for the ImageNet dataset. SPSA was set up with a delta value of 0.01, a learning rate of 0.01, a batch size of 256, and 100 maximum iterations for CIFAR-10 and a batch size of 128 for ImageNet.

Three white-box attacks, PGD, CW, and EAD, were used to attack the proposed defense. The PGD attack was configured with a step size of $2/255$, 50 iterations, and random initialization. Since we focused on untargeted attacks, CW and EAD were configured with a confidence value of 0, learning rate of 0.01, binary search steps of 9, and an initial constant of 0.001 for 1000 iterations for CIFAR-10 and 100 iterations for ImageNet. EAD was set up with the elastic-net (EN) decision rule.

D. Evaluation Metrics

We used two metrics: accuracy (ACC) and attack success rate (ASR). ACC is given by

$$\text{ACC} = \begin{cases} \frac{1}{N} \sum_{i=1}^N \mathbb{1}(f_{\theta}(x_i) = y_i) & \text{(clean)} \\ \frac{1}{N} \sum_{i=1}^N \mathbb{1}(f_{\theta}(x_i + \delta_i) = y_i) & \text{(attacked)}, \end{cases} \quad (11)$$

and ASR is defined as

$$\text{ASR} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(f_{\theta}(x_i) = y_i \wedge f_{\theta}(x_i + \delta_i) \neq y_i), \quad (12)$$

where N is the number of test images, $\mathbb{1}(\text{condition})$ is one if condition is true, otherwise zero, $\{x_i, y_i\}$ is a test image (x_i) with its corresponding label (y_i), and δ_i is its respective adversarial noise depending on a specific attack.

E. Robustness Against Black-Box and White-Box Attacks

A noise distance ϵ value of $8/255$ was used in ℓ_{∞} -bounded attacks. Table I captures the performance of both the baseline model (standard) and the proposed defense models for different block sizes ($M \in \{2, 4, 8, 16\}$) in terms of clean ACC and ASR. ACC was calculated for the whole test set (10,000 images for CIFAR-10 and 50,000 for ImageNet), and we computed ASR for 1,000 randomly selected images that were correctly classified by the proposed defense models. The models are denoted by their defense method and block size. For example, a model trained by using Pixel Shuffling with a block size of $M = 2$ is indicated as ‘‘Pixel Shuffling ($M = 2$).’’ From Table I, the results are summarized as follows.

1) CIFAR-10:

- **Standard:** Although the baseline (non-protected) model achieved the highest accuracy, it was most vulnerable to all attacks.
- **Pixel Shuffling:** The model with $M = 2$ provided a clean ACC of 94.45%, and the worst case ASR was 11.30% with SPSA. The model with $M = 16$ reduced the clean ACC to 76.22% although the ASRs for all attacks were low. Overall, the model with $M = 4$ performed reasonably well whether or not it was under attack.

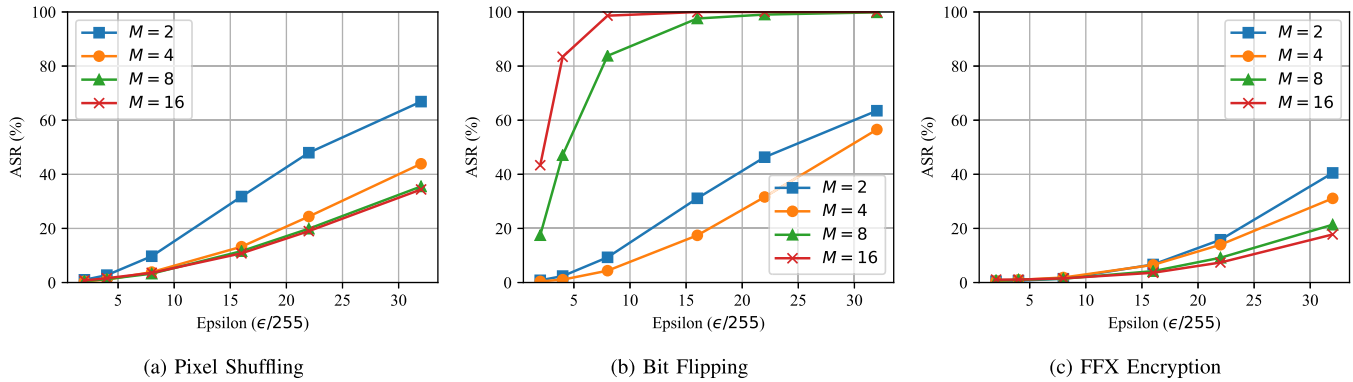


Fig. 8. ASR of proposed defense against PGD attack for CIFAR-10 dataset. ASR was calculated over 10,000 images (whole test set).

- **Bit Flipping:** The ACC of the model with $M = 2$ was very close to that of the standard model (i.e., 95.32%). However, the ASR was more than 10% for OnePixel and SPSA attacks. The models with $M = 8$ and 16 were broken as the ASR was high. For Bit Flipping, the model with $M = 4$ achieved the overall best accuracy.
- **FFX Encryption:** Although the ACCs of the models with FFX Encryption were slightly lower, they had better resistance against all of the attacks. Again, for the FFX Encryption defensive transformation, the model with $M = 4$ performed better overall.

In summary, a bigger block size reduced the classification accuracy for Pixel Shuffling and increased the ASR for Bit Flipping. However, for FFX Encryption, although the clean ACC was slightly lower, it provided better defense throughout the attacks (i.e., a lower ASR) for all different block sizes. Our experiments suggest that $M = 4$ is the optimal parameter for all three proposed defensive transformations. In addition, we plotted the ASR against noise distance ϵ (maximum of 32/255) for the whole test set under PGD attack in Fig. 8. The ASR for all three transformations increased with respect to bigger ϵ values. FFX Encryption had a lower ASR throughout all noise levels, and Bit Flipping had a very high ASR for $M = 8$ and 16.

2) *ImageNet*: Since $M = 4$ provided overall better results, we used $M = 4$ for the ImageNet dataset.

- **Standard:** Similarly, the standard model achieved the highest clean accuracy and ASR for all attacks.
- **Pixel Shuffling:** The ASRs of SPSA and PGD were 6.26% and 5.69%, respectively, and those of the other attacks were very low.
- **Bit Flipping:** Similarly, the ASRs of SPSA and PGD for Bit Flipping were 6.16% and 6.56%, respectively, and those of the other attacks were very low.
- **FFX Encryption:** The results show that FFX Encryption provided a lower ASR compared with Pixel Shuffling and Bit Flipping for SPSA and PGD attacks (i.e., 5.77% and 3.77% respectively). The ASRs of the other attacks were also very low.

Notably, the proposed defense achieved almost the same clean accuracy as the standard one (i.e., $\approx 72\%$), and the ASR was lower than 7% for all cases. Figure 9 shows the performance

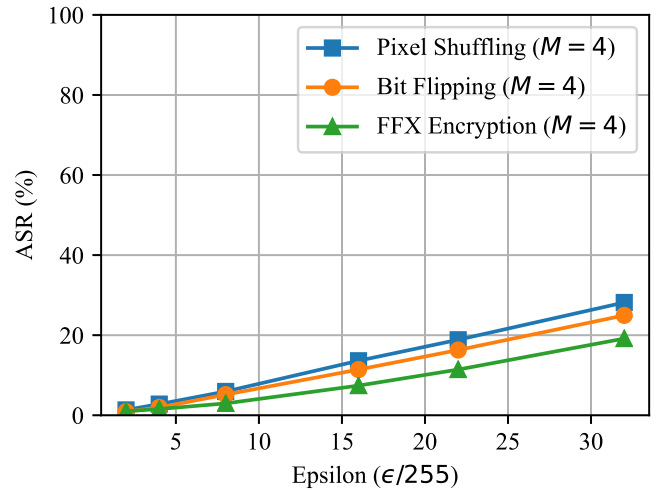


Fig. 9. ASR of proposed defense against PGD attack for ImageNet dataset. ASR was calculated over 10,000 images randomly selected from validation set.

of the proposed defense with the PGD attack under different noise distances. For the worst-case scenario (i.e., $\epsilon = 32/255$), the ASRs for Pixel Shuffling and Bit Flipping were approximately 28% and 25%, respectively. In contrast, the ASR of FFX Encryption was less than 20%.

F. Comparison With State-of-the-Art Defenses

First, we compared the accuracy of the proposed defense among the three key-based transformations with different block sizes under PGD attack for the CIFAR-10 dataset. Graphs of accuracy versus perturbation budget ϵ are shown in Fig. 10. When $\epsilon = 8/255$, the model with FFX Encryption ($M = 2$) achieved the highest accuracy (93.01%). As for the worst case ϵ (i.e., 32/255), the model with FFX Encryption ($M = 16$) yielded 76.74%. Notably, Bit Flipping for $M = 8$ and 16 reduced the accuracy significantly even for small ϵ values. However, the models with $M = 4$ provided the overall best accuracy, especially for an ϵ value of 8/255 for all three transformations. Therefore, we used the models with $M = 4$ as representatives for comparison with state-of-the-art defenses.

Most preprocessing-based defenses such as [50]–[54] were defeated by adaptive attacks due to obfuscated gradients [18].

TABLE I
CLEAN ACCURACY (ACC) (%) AND ATTACK SUCCESS RATE (ASR) (%) OF STANDARD AND PROPOSED DEFENSE MODELS UNDER DIFFERENT ATTACKS
WHERE $\epsilon = 8/255$ FOR ℓ_∞ METRIC

CIFAR-10								
Model	Clean ACC		ASR (Black-box)			ASR (White-box)		
	Standard	Protected	OnePixel (ℓ_0)	NATTACK (ℓ_∞)	SPSA (ℓ_∞)	PGD (ℓ_∞)	CW (ℓ_2)	EAD (ℓ_1)
Standard	95.45	–	79.90	99.90	100.00	100.00	100.00	100.00
Pixel Shuffling ($M = 2$)		94.45	9.50	0.80	11.30	9.80	0.00	0.09
Pixel Shuffling ($M = 4$)	–	91.84	5.00	0.20	3.00	3.30	0.00	0.00
Pixel Shuffling ($M = 8$)		85.12	3.90	0.00	3.60	4.00	0.09	0.19
Pixel Shuffling ($M = 16$)		76.22	2.80	0.20	3.37	3.42	0.00	0.00
Bit Flipping ($M = 2$)		95.32	10.50	0.70	10.25	9.62	0.00	0.18
Bit Flipping ($M = 4$)	–	93.41	5.30	0.20	4.29	4.64	0.00	0.09
Bit Flipping ($M = 8$)		91.54	21.40	32.50	88.14	84.34	2.26	3.11
Bit Flipping ($M = 16$)		92.68	27.10	71.80	98.24	98.98	5.40	8.75
FFX Encryption ($M = 2$)		93.67	6.30	1.90	2.46	1.77	0.47	0.00
FFX Encryption ($M = 4$)	–	92.30	3.90	2.10	2.45	1.96	0.28	0.00
FFX Encryption ($M = 8$)		91.99	2.00	4.20	3.57	1.41	0.00	0.00
FFX Encryption ($M = 16$)		91.38	3.20	6.30	6.62	1.60	0.28	0.00

ImageNet								
Model	Clean ACC		ASR (Black-box)			ASR (White-box)		
	Standard	Protected	OnePixel (ℓ_0)	NATTACK (ℓ_∞)	SPSA (ℓ_∞)	PGD (ℓ_∞)	CW (ℓ_2)	EAD (ℓ_1)
Standard	73.70	–	13.30	97.80	99.60	100.00	100.00	100.00
Pixel Shuffling ($M = 4$)	–	72.41	1.90	0.40	6.26	5.69	0.10	0.10
Bit Flipping ($M = 4$)	–	72.63	1.20	0.00	6.16	6.56	0.00	0.00
FFX Encryption ($M = 4$)	–	72.18	0.70	0.60	5.77	3.77	0.90	0.00

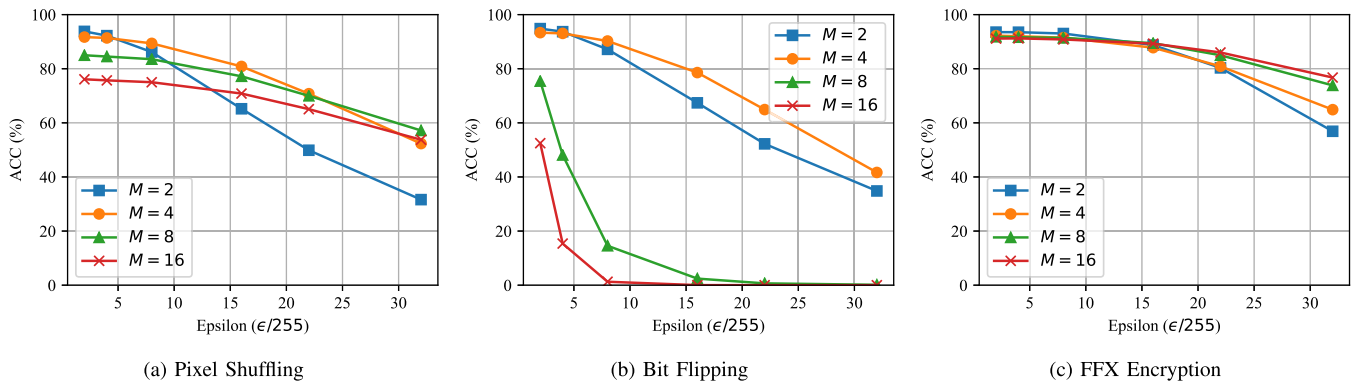


Fig. 10. ACC of proposed defense against PGD attack for CIFAR-10 dataset. ACC was calculated over 10,000 images (whole test set).

Even the most recent state-of-the-art defenses were invalidated by rigorous adaptive attacks [19]. To the best of our knowledge, only adversarial training (AT) is repeatedly found effective to defend against adversarial examples. However, AT has been known to be extremely difficult at the ImageNet scale

due to the high computation cost [6]. Recently, “Fast” AT was proposed to overcome such difficulty [49]. We compared the proposed defense models with the latest efficient AT (i.e., Fast AT) [49] as a baseline defense, a recent feature scattering-based approach (FS) [71], and another key-based defense

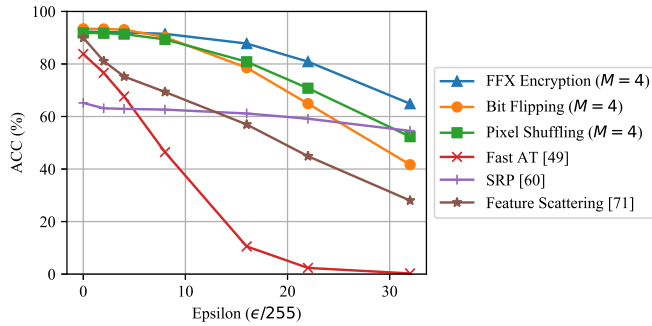


Fig. 11. Comparison of proposed defense with state-of-the-art defenses in terms of ACC under PGD attack for CIFAR-10 dataset. ACC was calculated over 10,000 images (whole test set).

using standard random permutation (SRP) [60] in terms of accuracy, whether or not the model was under PGD attack with various perturbation budgets. We exclude defenses that are already broken or that have a very low clean accuracy from comparison.

1) *CIFAR-10*: Figure 11 shows the performance of the proposed defense models with $M = 4$ compared with Fast AT [49], FS [71], and SRP [60]. In terms of clean accuracy, the model with Bit Flipping ($M = 4$) achieved the highest accuracy (i.e., 93.41%), while Fast AT was 83.80%, FS was 89.98%, and SRP was 65.16%. When the noise distance was $8/255$, the model with FFX Encryption ($M = 4$) outperformed all of the methods, achieving 91.48% compared with Fast AT (46.44%), FS (69.35%), and SRP (62.63%). When the perturbation budget was increased to $32/255$, the model with FFX Encryption ($M = 4$) still provided the highest accuracy (64.86%). Overall, all of the models with the proposed transformations outperformed state-of-the-art defenses at any given perturbation budget.

2) *ImageNet*: In a similar fashion, we conducted the PGD attack with different perturbation budgets to confirm the effectiveness of the proposed defense. The accuracy of SRP [60] for ImageNet was 9.99%; therefore, we excluded SRP from comparison. Moreover, FS [71] is not available for the ImageNet dataset. Therefore, we compared the proposed defense with the Fast AT [49] released by the original authors, which was trained with an ϵ value of $4/255$. Figure 12 shows the performance comparison under the PGD attack with different ϵ values in terms of ACC. The model with FFX Encryption outperformed all other methods for any given perturbation budget. In the literature, there is no defense that can maintain clean accuracy close to the standard one at the ImageNet scale. We are the first to achieve the closest clean accuracy as well as a high accuracy under the attacks even on the ImageNet dataset.

G. Robustness Against Adaptive Attacks

Without the correct key or a near-correct key, conventional attacks will not work on the proposed defense with a secret key. Therefore, we assume an attacker may estimate the correct key K randomly or heuristically. Once K was estimated, we ran the PGD attack by using the estimated key K' since

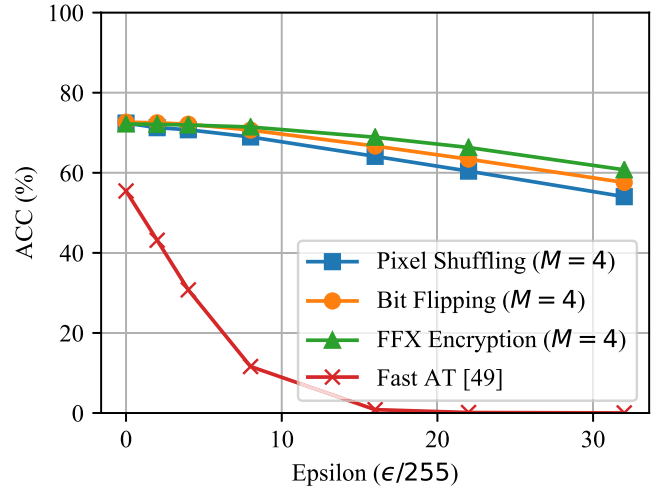


Fig. 12. Comparison of proposed defense with state-of-the-art defenses in terms of ACC under PGD attack for ImageNet dataset. ACC was calculated over 50,000 images (whole validation set).

PGD is one of the strongest adversaries, and the proposed defense was confirmed effective when the key was correct. Apart from key estimation methods, we also deployed the adaptive attacks described in IV-E to evaluate the proposed defense.

1) *Random Key Estimation Approach*: One of the ways of estimating key K is to randomly search for a key. As in black-box settings, the attacker may query the model with their key. We allow the attacker to query the model for a maximum of 20,000 queries. In other words, the attacker uses a single image and a key K' at a time to query the model. When the model makes the correct prediction for the test image with respect to the key K' , the attacker stops the random search and uses K' to generate adversarial examples. While considering the worst-case scenario, we also assume the attacker has the weights of the model (white-box) and can use a batch of images to test a key K' over the average accuracy.

The key space can be varied depending on the number of pixels in a block p_b . The key space of Pixel Shuffling is given by

$$\mathcal{K}_{\text{shuffling}}(p_b) = p_b!. \quad (13)$$

For Bit Flipping and FFX Encryption, 50% of the pixels in each block are inverted/encrypted, and the key controls which pixels are inverted/encrypted. Therefore, their key spaces are the same and written as

$$\mathcal{K}_{\text{flipping/encryption}}(p_b) = \binom{p_b}{\frac{p_b}{2}} = \frac{p_b!}{(\frac{p_b}{2})! \cdot (\frac{p_b}{2})!}. \quad (14)$$

2) *Heuristic Key Estimation Approach*: As in white-box settings, we assume the attacker knows the model weights and inner workings of the defense algorithm. In this case, instead of trying a key randomly, key K may be estimated by using a heuristic approach. In other words, K is not directly estimated, but the transformation pattern caused by K is estimated. A key is used to generate a random permutation vector $v = (v_0, v_1, \dots, v_{p_b-1})$ for Pixel Shuffling and a

TABLE II
ATTACK SUCCESS RATE (ASR) (%) OF ADAPTIVE ATTACKS FOR CIFAR-10 DATASET

Model	Key Search			Inverse Transformation	EOT	Transferability Attack
	Random		Heuristic			
	Single	Batch				
Pixel Shuffling ($M = 4$)	3.80	4.45	3.70	3.87	1.70	5.07
Bit Flipping ($M = 4$)	3.50	3.82	77.76	78.22	1.58	1.49
FFX Encryption ($M = 4$)	1.70	1.23	3.27	–	5.24	6.41

random binary vector $r = (r_0, r_1, \dots, r_{p_b-1})$ for Bit Flipping and FFX Encryption. Therefore, the adversary can modify v or r by using the average accuracy over a batch of images as a guide to carry out an adaptive attack as follows (see Algorithm 4).

- 1) Initialize a permutation vector v (for Pixel Shuffling) or a binary vector r (for Bit Flipping/FFX Encryption) with a random key K' .
- 2) Calculate the accuracy of the model over a batch of images.
- 3) Repeatedly swap two values in v : v_i and v_j (for Pixel Shuffling) or in r : r_i and r_j (for Bit Flipping/FFX Encryption) for T rounds if the accuracy improves.
- 4) Return the tuned v (for Pixel Shuffling) or r (for Bit Flipping/FFX Encryption) to proceed with the adaptive attack.

Algorithm 4 Heuristic Key Estimation Approach

Input: A batch of images

Output: v or r

```

Initialize  $v$  or  $r$  with a random key  $K'$ 
accuracy  $\leftarrow$  Calculate the accuracy of the model
for  $t \leftarrow 1 \dots T$  do
  for  $i \leftarrow 0, \dots, p_b - 1$  do
    for  $j \leftarrow i + 1, \dots, p_b - 1$  do
      if accuracy improves then
        Swap  $v_i$  and  $v_j$  for Pixel Shuffling
        or
        Swap  $r_i$  and  $r_j$  for Bit Flipping/FFX Encryption
      end if
    end for
  end for
end for

```

We implemented the key search approaches on the CIFAR-10 dataset with a batch size of 128 and parameter $T = 10$. A note on FFX Encryption is that password P does not matter since the length of FFX encryption is fixed (i.e., 3). Therefore, the attacker can assume any password during the attack. Table II summarizes the results of the key search attacks. In all transformations, random key search approaches (either by a single image or batch of images) did not guarantee that a close-enough key was found since ASR was very low. For the heuristic approach, ASR was 77.76% for Bit Flipping ($M = 4$), 3.7% for Pixel Shuffling ($M = 4$), and 3.27%

for FFX Encryption ($M = 4$). Although the ASR for Bit Flipping was high, this type of attack is only possible when the model weights are available to the attacker. However, Pixel Shuffling and FFX Encryption were still resistant to such attacks. Moreover, one key belongs to one model only and, therefore, the attacker cannot generalize the attack.

3) *Inverse Transformation Attack:* In Fig. 13, examples of adversarial and adaptive adversarial examples are illustrated under the PGD attack for each algorithm, where key K was estimated by using the heuristic approach (see Algorithm 4) with $T = 10$. For FFX Encryption, the visibility of the adaptive adversarial example was heavily changed compared with Pixel Shuffling and Bit Flipping. In other words, the perturbations were clearly perceptible, and valid adversarial examples were not found under this type of attack for FFX Encryption. Therefore, we do not report the result of this adaptive attack for FFX Encryption in Table II. Since the estimated key was not good enough, the ASR was still very low for both Pixel Shuffling and Bit Flipping.

4) *Estimation Over Transformation Attack:* The results for the EOT attack are summarized in Table II. From the experiments, the ASR was also very low (less than 2%) for Pixel Shuffling and Bit Flipping and $\approx 5\%$ for FFX Encryption. Therefore, the proposed defense was still resistant against such adversarial examples.

5) *Transferability Attack:* We simulated this attack scenario, and the results are presented in Table II. The ASR was 5.07% for Pixel Shuffling, 1.49% for Bit Flipping, and 6.41% for FFX Encryption. The results suggest that the proposed method can still defend against adversarial examples under this type of attack.

H. Discussion on Obfuscated Gradients

Obfuscated gradients occur under three conditions: (1) shattered gradients (non-existent/incorrect) due to non-differentiable operations or numerical instability, (2) stochastic gradients because of test-time randomness, and (3) vanishing/exploding gradients for very deep computation [18]. These obfuscated gradients can be bypassed by using attack techniques such as Backward Pass Differentiable Approximation (BPDA) as in [18].

The proposed defense does not produce good gradients because of the use of a block-wise transformation, therefore, it is a kind of gradient obfuscation as in the above two conditions: (1) and (3). The gradients from the proposed defense

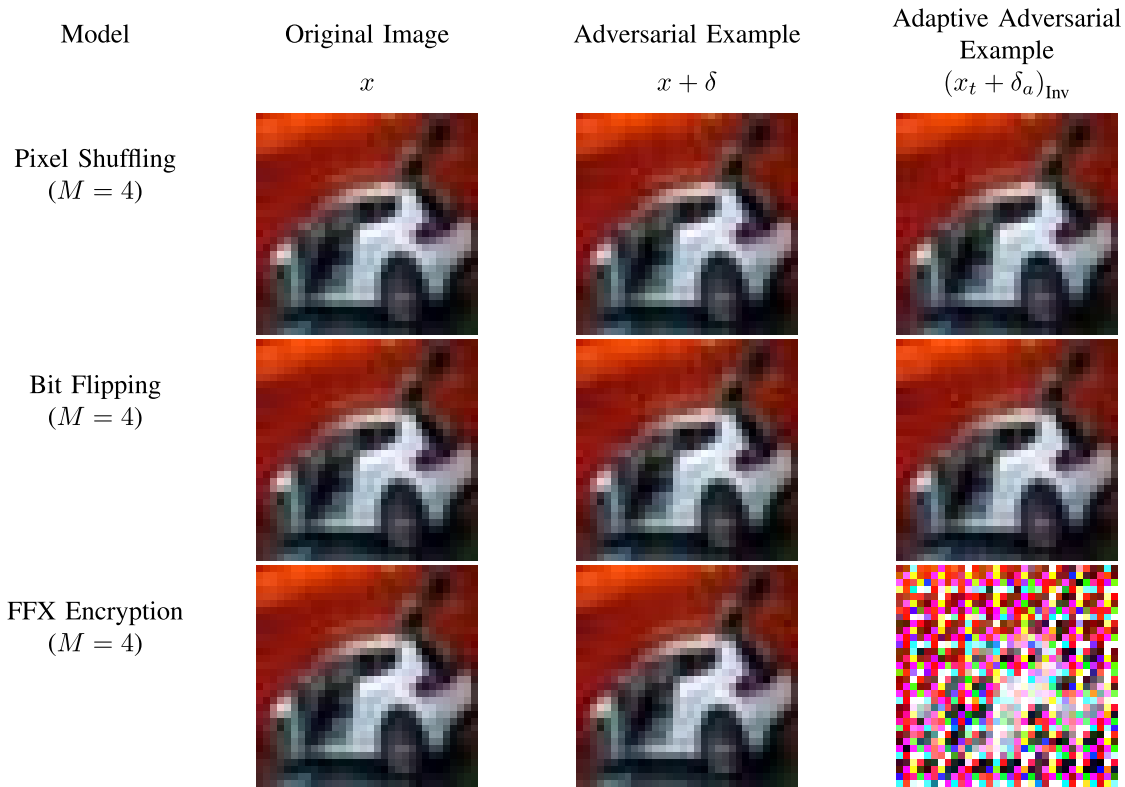


Fig. 13. Example of original test image, adversarial examples, and adaptive adversarial examples generated with estimated key for PGD (ℓ_∞) with $\epsilon = 8/255$, where adaptive adversarial example includes severe distortion for FFX encryption.

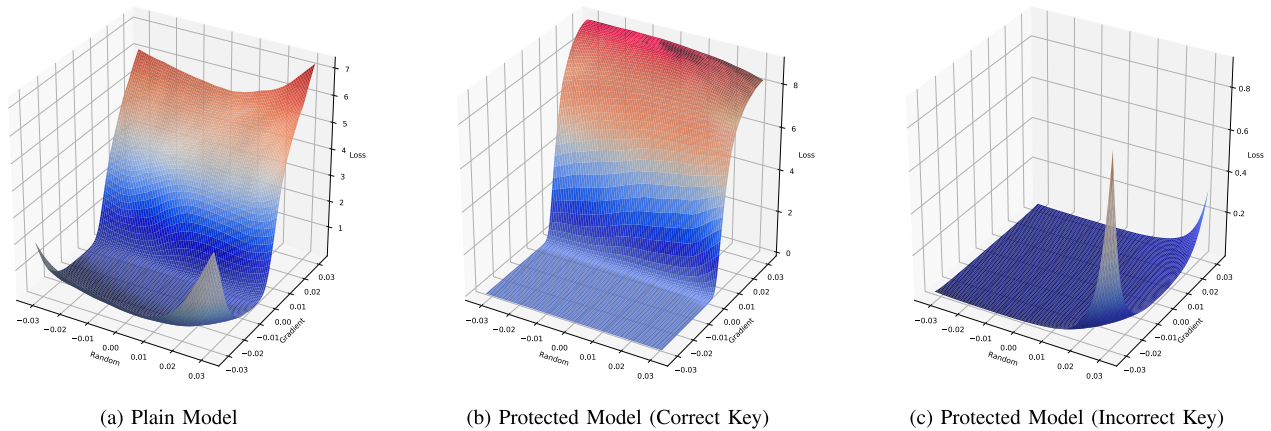


Fig. 14. Loss surfaces along two dimensions in the input space: actual gradient direction and random direction of (a) plain (non-protected) model, (b) protected model with a correct key, and (c) protected model with an incorrect key. The protected models were trained by using images transformed by Pixel Shuffling ($M = 4$) on the CIFAR-10 dataset.

are directly controlled by using key K unlike conventional obfuscation methods as mentioned in [18]. In other words, when an estimated key is close to the correct key, the attack will be more successful. We have verified that when the correct secret key is given, the attacks (both white-box and black-box) are 100% successful. The proposed defense adds a layer of protection in which the secret key has to be first guessed to carry out a successful attack. Therefore, the proposed defense can be viewed as a hard-obfuscating defense.

To further explain the robustness of the proposed defense in the absence of key K , we show the graphs of the loss

surface along two dimensions in the input space: actual gradient direction and random direction of (a) the plain (non-protected model), (b) protected model with a correct key, and (c) protected model with an incorrect key in Fig. 14. From the figure, the loss surface of (a) plain model is similar to that of (b) protected model with the correct key; the loss increases substantially in both random and gradient directions. In contrast, for Fig. (c) protected model with the incorrect key, the loss surface is completely different from those of (a) and (b) (i.e., the loss is mostly close to zero in both directions). Specifically, out of 10,000 samples, only 3 of them

were misclassified for the graph in (c). Therefore, the proposed defense is robust against attacks when the key is secret (i.e., incorrect key is used for attacks).

VI. CONCLUSION

In this paper, we proposed a novel block-wise image transformation as a preprocessing defense method, where both input images and test ones are preprocessed by using the proposed transformation with a key. To realize the proposed transformation, we developed three algorithms: Pixel Shuffling, Bit Flipping, and FFX Encryption. The results showed that the proposed defense was robust against conventional threat models under various metrics (ℓ_∞ , ℓ_2 , ℓ_1 , ℓ_0), achieving more than 90% accuracy for both clean images and adversarial examples. In addition, we also conducted various adaptive attacks to further evaluate the effectiveness of the proposed defense. Under PGD attack with different perturbation budgets, the proposed defense outperformed the state-of-the-art adversarial defenses with the CIFAR-10 and ImageNet datasets. Moreover, the proposed defense was confirmed to bring robust accuracy close to non-robust accuracy for both the CIFAR-10 and ImageNet datasets for the first time.

REFERENCES

- [1] 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.* New York, NY, USA: ACM, Oct. 2015, pp. 1322–1333.
- [2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 3–18.
- [3] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–10.
- [4] B. Biggio *et al.*, "Evasion attacks against machine learning at test time," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2013, pp. 387–402.
- [5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–11.
- [6] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–17.
- [7] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.
- [8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.
- [9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–28.
- [10] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," 2017, *arXiv:1712.02779*. [Online]. Available: <http://arxiv.org/abs/1712.02779>
- [11] J. Gilmer, R. P. Adams, I. Goodfellow, D. Andersen, and G. E. Dahl, "Motivating the rules of the game for adversarial example research," 2018, *arXiv:1807.06732*. [Online]. Available: <http://arxiv.org/abs/1807.06732>
- [12] K. Eykholt *et al.*, "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1625–1634.
- [13] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 284–293.
- [14] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.
- [15] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519.
- [16] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1528–1540.
- [17] N. Carlini *et al.*, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*. [Online]. Available: <http://arxiv.org/abs/1902.06705>
- [18] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.
- [19] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," 2020, *arXiv:2002.08347*. [Online]. Available: <http://arxiv.org/abs/2002.08347>
- [20] 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.
- [21] 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, pp. 1–15, Dec. 2019.
- [22] W. Sirichotedumrong, T. Maekawa, Y. Kinoshita, and H. Kiya, "Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 674–678.
- [23] 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.
- [24] K. Madono, M. Tanaka, M. Onishi, and T. Ogawa, "Block-wise scrambled image recognition using adaptation network," 2020, *arXiv:2001.07761*. [Online]. Available: <http://arxiv.org/abs/2001.07761>
- [25] M. Tanaka, "Learnable image encryption," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW)*, May 2018, pp. 1–2.
- [26] K. Kurihara, S. Imaizumi, S. Shiotani, and H. Kiya, "An encryption-then-compression system for lossless image compression standards," *IEICE Trans. Inf. Syst.*, vol. E100.D, no. 1, pp. 52–56, 2017.
- [27] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *J. Mach. Learn. Res.*, vol. 20, no. 184, pp. 1–25, 2019.
- [28] 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.
- [29] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, 2010.
- [30] A. Shafahi *et al.*, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6103–6113.
- [31] P. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. 32nd AAAI Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, 2018, pp. 10–17.
- [32] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.
- [33] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4312–4321.
- [34] C. Xie *et al.*, "Improving transferability of adversarial examples with input diversity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2730–2739.
- [35] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 15–26.
- [36] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 2142–2151.
- [37] J. Uesato, B. O'Donoghue, P. Kohli, and A. van den Oord, "Adversarial risk and the dangers of evaluating against weak attacks," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 5032–5041.

- [38] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 10934–10944.
- [39] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "NATTACK: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 3866–3876.
- [40] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.
- [41] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.
- [42] K. Dvijotham, R. Stanforth, S. Gowal, T. A. Mann, and P. Kohli, "A dual approach to scalable verification of deep networks," in *Proc. UAI*, 2018, pp. 550–559.
- [43] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5283–5292.
- [44] H. Salman *et al.*, "Provably robust deep learning via adversarially trained smoothed classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11289–11300.
- [45] S. Gowal *et al.*, "On the effectiveness of interval bound propagation for training verifiably robust models," 2018, *arXiv:1810.12715*. [Online]. Available: <http://arxiv.org/abs/1810.12715>
- [46] M. Mirman, T. Gehr, and M. T. Vechev, "Differentiable abstract interpretation for provably robust neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 3575–3583.
- [47] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter, "Scaling provable adversarial defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8400–8409.
- [48] A. Shafahi *et al.*, "Adversarial training for free!" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3353–3364.
- [49] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–17.
- [50] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–22.
- [51] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.
- [52] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille, "Mitigating adversarial effects through randomization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.
- [53] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "PixelDefend: Leveraging generative models to understand and defend against adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.
- [54] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.
- [55] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6521–6530.
- [56] M. Aprilpyone, Y. Kinoshita, and H. Kiya, "Adversarial robustness by one bit double quantization for visual classification," *IEEE Access*, vol. 7, pp. 177932–177943, 2019.
- [57] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–12.
- [58] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, *arXiv:1703.00410*. [Online]. Available: <http://arxiv.org/abs/1703.00410>
- [59] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.
- [60] O. Taran, S. Rezaeifar, and S. Voloshynovskiy, "Bridging machine learning and cryptography in defence against adversarial attacks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 1–13.
- [61] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [62] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: <http://arxiv.org/abs/1708.07747>
- [63] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.
- [64] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [65] M. Bellare, P. Rogaway, and T. Spies, *Addendum to 'The FFX Mode of Operation for Format-Preserving Encryption': A Parameter Collection for Enciphering Strings Arbitrary Radix Length*, document Draft 1.0, NIST, 2010.
- [66] M. Aprilpyone and H. Kiya, "Ensemble of models trained by key-based transformed images for adversarially robust defense against black-box attacks," 2020, *arXiv:2011.07697*. [Online]. Available: <http://arxiv.org/abs/2011.07697>
- [67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [68] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," 2017, *arXiv:1708.07120*. [Online]. Available: <http://arxiv.org/abs/1708.07120>
- [69] P. Micikevicius *et al.*, "Mixed precision training," 2017, *arXiv:1710.03740*. [Online]. Available: <http://arxiv.org/abs/1710.03740>
- [70] G. W. Ding, L. Wang, and X. Jin, "AdverTorch v0.1: An adversarial robustness toolbox based on PyTorch," 2019, *arXiv:1902.07623*. [Online]. Available: <http://arxiv.org/abs/1902.07623>
- [71] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1831–1841.



IEEE ICCE-TW Best Paper Award in 2016.

Maungmaung Aprilpyone (Graduate Student Member, IEEE) received the B.C.S. degree from International Islamic University Malaysia in 2013, under the Albukhary Foundation Scholarship, and the M.C.S. degree from the University of Malaya in 2018, under the International Graduate Research Assistantship Scheme. He is currently pursuing the Ph.D. degree with Tokyo Metropolitan University, under the Tokyo Human Resources Fund for City Diplomacy Scholarship. His research interests are in the area of information security. He received the



Hitoshi Kiya (Fellow, IEEE) 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 was a recipient of numerous awards, including ten best paper awards. He currently serves as the President of 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. He was also the President of the IEICE Engineering Sciences Society from 2011 to 2012, and he served as the Vice President and Editor-in-Chief for the IEICE Society Magazines and Society Publications. He has been an editorial board member of eight journals, including IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the Chair of two technical committees, and a member of nine technical committees, including the APSIPA Image, Video, and Multimedia Technical Committee (TC) and IEEE Information Forensics and Security TC. He has organized a lot of international conferences in such roles as the TPC Chair of IEEE ICASSP 2012 and as the General Co-Chair of IEEE ISCAS 2019.