공학석사 학위논문

픽셀 강도 암호화를 통한 적대적 강건성 강화

Improving Adversarial Robustness Using Pixel Intensity Encryption

2021년 8월

서울대학교 대학원

지능정보융합학과

이윤아

픽셀 강도 암호화를 통한 적대적 강건성 강화

Improving Adversarial Robustness Using Pixel Intensity Encryption

지도 교수 Wonjong Rhee

이 논문을 공학석사 학위논문으로 제출함
2021년 8월

서울대학교 융합과학기술대학원
지능정보융합학과
이윤아

이윤아의 공학석사 학위논문을 인준함
2021년 8월

위 원 장 Kyogu Lee

부위원장 Wonjong Rhee

위 원 Jaeheung Park

Abstract

Improving Adversarial Robustness Using Pixel Intensity Encryption

Yoonah Lee

School of Convergence Science & Technology

The Graduate School

Seoul National University

Neural networks are known to be vulnerable to gradient-based adversarial examples which are made by leveraging input gradients toward misclassification. Due to these attacks, adversarial defense has become a topic of significant interest in recent years. The most empirically successful approach to defending against such adversarial examples is adversarial training, which incorporates a strong self-attack during training. However, this approach is computationally expensive and hence is hard to scale up. As a result, a series of studies has been undertaken to develop gradient masking methods. One of the method is to to hide the gradient using encryption. This was achieved by transforming the location of pixels. However, there have been no studies regarding how pixel-intensity encryption could work as an adversarial defense.

This study proposes a new defense method that uses pixel intensity encryption to defend against the gradient-based attacks. Furthermore, A

new adaptive attack setup for encryption methods is presented in the study to evaluate its effectiveness as an adversarial defense. The experiment shows that the proposed defense is more robust than that of the previous studies under adaptive attack. Moreover, the correlation coefficient of an image is found to make the key role on learnability of the model.

Keywords: Adversarial examples, adversarial attack, adversarial defense, perceptual image encryption

Student Number: 2019-20659

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

As deep neural networks have been deployed in many security-sensitive applications such as self-driving cars, health-care, facial recognition, robustness is highly demanded to guarantee the reliability of neural networks. However, recent research has demonstrated that neural networks can be easily fooled by adversarial examples which are deliberately crafted input samples (Szegedy et al., 2014). The adversarial perturbations are imperceptible to humans but can cause neural networks to misclassify with high confidence. Due to this threat, adversarial defense, which aims to defend the adversarial attack, has received a significant amount of attention as of late.

Attacking a network is straightforward. The most successful way to make adversarial examples is to use input gradients that indicate the direction to the maximal loss of the network. This attack method is called a gradient-based attack.

Adversarial training (Madry et al., 2017) is by far the most successful method against gradient-based adversarial attacks. It trains the network model using adversarial examples which are generated during training. While this method achieves strong adversarial robustness, this success comes with certain drawbacks. Firstly, it is notoriously slow as it necessitates the creation of adversarial examples for use in each training epoch. Therefore, it is not available for large datasets such as ImageNet. Secondly, training exclusively on

l_∞ adversarial examples provides the model with only limited robustness to adversarial examples under other distortion metrics (Sharma & Chen, 2017).

The difficulty of adversarial training suggests an alternative path: Can we mask gradients so that attackers cannot use gradients to make adversarial examples? In this line of thought, a series of studies has been undertaken to develop gradient masking methods. Those methods work by inserting a non-differentiable layer into the network so that the attacker cannot access to a useful gradient. However, most of these methods, called obfuscated gradients, hide gradients while training the network so that the gradient does not successfully optimize the loss, offering a tradeoff between accuracy and robustness. It was recently pointed out by Athalye et al. (2018) that hidden gradients can still be approximated in these methods and adaptive attacks are required to evaluate the true adversarial robustness of gradient masking methods.

Rather than obfuscating the gradient, research was also undertaken to develop methods to hide the gradient by introducing encryption to the network (AprilPyone et al., 2020; Taran et al., 2018). A pixel-location shuffling method was used as an image encryption technique and its suitability as an adversarial defense was evaluated. However, there has been no studies about how encrypting the intensity of an pixel could work as an adversarial defense and the adaptive attack framework was not constructed for an adversarial encryption defense.

In this study, we make the following contributions.

- We first propose a new encryption method as an adversarial

defense that uses pixel intensity encryption

- We design the adaptive attack framework for the adversarial encryption defense method to evaluate its true robustness.
- Through experimentation, we show the proposed method exhibits high robustness in the adaptive attacks.
- We first empirically examine the relationship between the security of encryption and learnability and find that the correlation coefficient of the image plays an important role in the learnability of the model.

1.1 Terminology

Here we introduce the definition of terms in this study. The following terms are based on the terminology by Tabassi et al. (2019) that recently summarized published papers regarding to adversarial examples.

- Adversary: The agent who intends to conduct detrimental activities, perhaps by creaing an adversarial example.
- Adversarial examples: Input samples formed by applying a small but intentional perturbation to a clean example, such that the perturbed input causes a learned model to output an incorrect answer.
- Adversarial perturbation: the noise added to an input sample to

make in an adversarial example. 0.031 perturbation is usually used to make an adversarial examples on CIFAR10.

• Gradient masking: an ML technique in which gradients are minimized to reduce the model's sensitivity to gradient based adversarial examples. Hides the gradient direction used to craft adversarial examples.

• White-box attack: Attack that exploits model internal information. It assumes complete knowledge of the targeted model, including its parameter values, architecture, training method, and in some cases its training data as well.

• Black-box attack: Attack that assumes no knowledge about the model under attack. The adverary may use context or historical information to infer model vulnerability.

• Adaptive attack: attack that was specifically designed to target a given defense.

• Standard accuracy: accuracy of the model on the original dataset

• Robust accuracy: accuracy of the model on the adversarially perturbed dataset.

• Adversarial robustness: the ability of an ML model/algorithm to maintain correct and reliable performance under adversarial examples.

• Transferability of adversarial examples: the ability of an adversarial example to remain effective even for the models other than the one used to generate it.

In addition to the above, we will use the term “Learnability” to denote the ability of an ML model to learn features from encrypted inputs so that the performance on the original dataset is not sacrificed throughout this paper.

Chapter 2. Related Work

2.1 Gradient-based Attack

If an adversary has full access to the network structure and parameters, then the adversary can exploit the input’s gradient with respect to the loss by that indicates how to perturb input to trigger the maximal loss of network by back-propagation. These methods are called gradient-based adversarial attacks and a baseline of such attacks is the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014), which constructs the adversarial example x’ of a given labeled data (x, y) using a gradient-based rule:

$$x_{adversarial} = x + \epsilon \cdot sign(\nabla_x L(f(x;\theta), y)$$

Where $f(x;\theta)$ denotes the neural network model’s output, $L(f(x;\theta), y)$ is the loss function provided $f(x;\theta)$ and the input label y, and ϵ is the perturbation range for the allowed adversarial example.

Extending the FGSM, the projected Gradient Descent (PGD) (Kurakin et al., 2016) utilizes the local first order gradient of the

network in an iterative way, and is considered the strongest first-order adversary (Madry e al., 2017). In each step of the PGD, the adversary example is updated by a FGSM rule, namely,

$$x_{t+1} = x_t + \alpha \cdot sign(\nabla_x L(f(x;\theta), y)) ,$$
$$\text{where } \|x - x_t\| \leq \epsilon \; for \; all \; t$$

Where x_t is the adversary examples after t steps, projecting x_t back into an allowed perturbation range epsilon.

2.2 Gradient Masking Defense

2.2.1 Obfuscated Gradients

Adversarial defense methods that conceal the gradient information from the attacker have been termed as gradient masking (Papernot et al., 2017) techniques.

One method of gradient masking is to exploit randomness by randomly introducing stochastic layers in the network. Dhillon et al. (2018) proposed Stochastic Activation Pruning (SAP), which introduces randomness into the evaluation of a neural network to defend against adversarial examples. SAP essentially applies dropout (Srivastava et al., 2014) during evaluation but nodes are dropped with a weighted distribution inversely proportional to their absolute values instead of dropping with uniform probability. The values which are retained are scaled up to retain accuracy.

Xie et al. (2018) proposed to add a randomization layer before the input to the classifier. The authors resized and added random padding to

the input. For example, 299×299 size of images were used in the experimentation and the authors first randomly rescaled the images to r×r , with r included [299,331], and then randomly zero-padded the images so that the result became 331×331 size of images. The output was then fed to the classifier.

Another type of gradient masking is to introduce discretization layers in the network so that the adversary cannot directly backpropagate through the discretization function to determine how to adversarially modify the input. Buckman et al. (2018) proposed to break the linear extrapolation behavior of neural networks by preprocessing the input with an nonlinear function. The authors proposed a new encoding method called thermometer encoding to discretize the input image without losing the relative distance information. This encoding discretizes an pixel value x_i into a l-dimensional vector $\tau(x_i)$, where $\tau(x_i)_k$ =1 if $x_i > \mathrm{k/l}$, and 0 otherwise. For example, if 10-level thermometer encoding is used, $\tau(0.66)$=1111110000.

However, the defense methods above are called obfuscated gradients and are attacked by Athalye et al. (2018). They offer a tradeoff between standard accuracy and robustness on the degree of randomness and discretization. The first defense method that adds randomization can be attacked 100% by computing the expectation over instantiations of randomness. The second defense method that adds discretization can be attacked easily as their functions are similar to the identity function. The discretization method can be attacked

easily by setting an identity function on their transformation layer when doing a backward pass or by transfer attack which attacks the model with adversarial examples generated on a standard trained model.

2.2.2 Adversarial Encryption Defense

The above gradient-masking methods have the limitations that they make obfuscated gradients, which make the network gradient either non-existent or incorrect. As a result, there exists a tradeoff between robustness and standard accuracy.

To solve these limitations, rather than by obfuscating the gradient, methods to hide the gradient with encryption were investigated. This was achieved by introducing an encryption layer to the network. Taran et al. (2018) first proposed an image encryption method as an adversarial defense. A pixel-location shuffling method was used and its suitability as an adversarial defense was evaluated. This study described Kerckhoff's cryptographic principle, that the key is not known to attackers and explained that encryption can be used as a gradient masking defense. This encryption has $n!$ key space, where n is the number or pixels in an image.

A study by AprilPyone et al. (2020) used a similar encryption method as that proposed by Taran et al. (2018). The main difference is that AptilPyone et al. (2020) proposed to shuffle pixels locally in certain block sizes, naming this technique block-wise pixel shuffling. They used different block sizes to see which sizes do not reduce

standard accuracy and chose 4 as the best block size for adversarial defense, which does not lose standard accuracy, while also exhibits high robust accuracy.

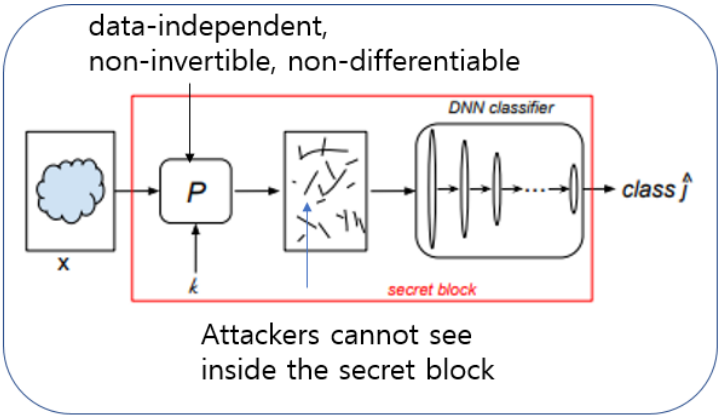


Figure 1. Principle scheme of Taran et al. (2018)

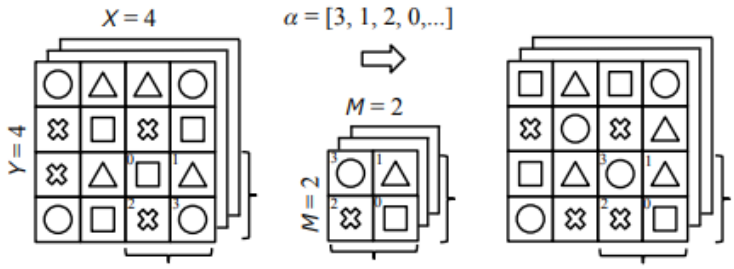


Figure 2. 2×2 block-wise pixel shuffling (AprilPyone et al., 2020)

Chapter 3. Research Questions

This study aims to propose the effectiveness of the new encryption method that transforms the intensity of an pixel as an adversarial defense and construct the adaptive attack framework for adversarial encryption defense. To do that, we answer the following threee research questions.

- RQ1. How does the pixel intensity encryption work as an adversarial defense?
- RQ2. What is a proper strategy to attack adversarial encryption defense adaptively?

Chapter 4. Proposed Method

This approach is different from block shuffling method by Aprilpyone et al. (2020): they only transform pixel location while, we first introduce the encryption of the pixel intensity for adversarial defense. The other is that we designed the new adaptive attack framework for encryption adversarial defense methods. In order to demonstrate that the gradient masking method is really as robust as argued by Athalye et al. (2018), it should be tested by way of an adaptive attack. We designed an adaptive attack scenario for encryption adversarial defense and observed that their method does not show strong robustness under adaptive attacks.

Following the cryptographic principle, it is assumed that all details of the proposed algorithms are publicly known and available to the attackers besides the key (Taran et al., 2018). Thus, it is important to make key space large and make the potential for a key to be guessed more difficult to ensure the strength of the proposed defense.

Based on this idea, we propose a defense method that encrypts the pixel intensity locally using larger key space than that of the previous studies and show its higher robustness under key guess attack.

4.1 Pixel-Intensity encryption

We introduce a transform that transforms pixel-intensity with a

secret key as an adversarial defense for the first time. The overview of the proposed defense is depicted in Figure 3. Training images are transformed using a secret key and the model is trained by the transformed images. Test images are also transformed using the same key before passing to the model.

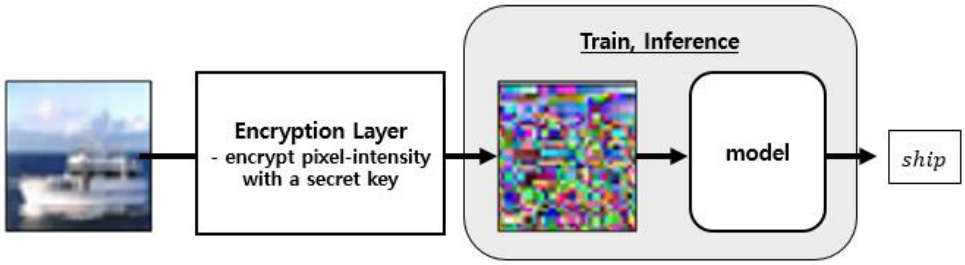


Figure 3. The overview of the proposed method

4.1.1 Affine encryption

First, we introduce Affine encryption, the basic encryption method, as a pixel-intensity transformation. The Affine encryption function is

$$\boldsymbol{E(x) = key1x + key2\ (mod\ N)}, \tag{3.1}$$

where N is the size of the pixel and key1 and key2 are the keys of the cipher. Since we use 8-bit images, N is set to 256. The multiplicative inverse of $\boldsymbol{key1}$ only exists if $\boldsymbol{key1}$ and N are coprime. Hence without the restriction on $\boldsymbol{key1}$, decryption might not be possible. Therefore, key1 should be an odd number in this case so key1 space is

128 and key2 space is 256.

The Affine decryption function is

$$\boldsymbol{D(x) = key1^{-1}(x - key2)\ (mod\ N)}, \tag{3.2}$$

where $\boldsymbol{key1^{-1}}$ is the modular multiplicative inverse of $\boldsymbol{key1\ mod\ N}$. i.e., it satisfies the equation

$$\boldsymbol{1 = key1^{-1}key1(mod\ N)} \tag{3.3}$$

The decryption function is the inverse of the encryption function and it can be shown as follows

$$\begin{aligned}\boldsymbol{D(E(x))} &= \boldsymbol{key1^{-1}(E(x) - k2)\ (mod\ N)}\\ &= \boldsymbol{key1^{-1}((key1x + key2)mod\ N - key2)\ (mod\ N)}\\ &= \boldsymbol{key1^{-1}(key1x + key2 - key2)\ (mod\ N)}\\ &= \boldsymbol{key1^{-1}key1x\ (mod\ N)}\\ &= \boldsymbol{x\ (mod\ N)}\end{aligned} \tag{3.4}$$

Algorithm 1 Affine Encryption

Input : input x, key1, key2

Output: x'

$X' \leftarrow$ key1*x + key2 (mod 256)

4.1.2 Pixel-intensity shuffling

We introduce a transform that changes pixel intensity locally with a secret key. This transformation is different from block-wise pixel

shuffling proposed by AprilPyone et al. (2020) in that it doesn't change the pixel location but the pixel intensity. For 8-bit images, a pixel can have 256 values (0~255) and shuffling the intensity of a pixel by setting the block size is possible. If the block size is set to 256, then the value of pixel intensity can be mapped to any value from 0 to 255. However, if the block size is set to 4, the value of pixel intensity can be mapped to another value from 0 to 3, 4 to 7, … , 252 to 255. If the block size is 1, this means it can only change its intensity within its original value so the transformed image is same as original image. The process of the transformation is illustrated in Fig.1. Key space is $m!$, where m is the number of pixels in a block. Both training and testing images are transformed with the same secret key before they are passed to deep neural networks.

Algorithm 2 Pixel Intensity Encryption

Input: input x, key, block size

Output: x'

Generate a random mapping vector v with length of block size by key

mapping_v ←concat($v + i$ for i in range(0, 256-block size +1, block size)

x'= mapping_v[x]

4.2 Upgraded Encryption

Rather than using one block mapping, different mapping blocks can be used to make key space larger in pixel location and intensity encryption. We present the upgraded Encryption method which uses a different mapping for each block.

For the location shuffle method, the key size is $\left(\frac{32}{\mathrm{m}}\right) * \left(\frac{32}{\mathrm{m}}\right) * m!$ when different mappings for each block are used, which was originally $m!$, where m is the number of pixels in a block. For example, the authors presented that 4 is the best block size for adversarial defense and it had $12!$ key size. If different mappings are used, the key size is $(8) * (8) * 12!$

For the pixel intensity shuffle method, the key size becomes $\left(\frac{256}{\mathrm{m}}\right) * m!$, where m is the number of pixels in a block. For example, we present 16 as the best block size for adversarial defense. The key size is $16!$ If same mapping is used for every block but the key size becomes $16 * 16!$

Throughout this paper we use the term “block shuffling” to denote the defense proposed by AprilPyone et al. (2020) and “pixelintensity” to denote the defense that shuffles pixel intensity. For each method, block size 4 is used for the block shuffling method as it shows the highest robustness in the previous studies and block size 16 is used for the pixelintensity method. “Upgrade” encryption describes that for each block, different mappings are used.

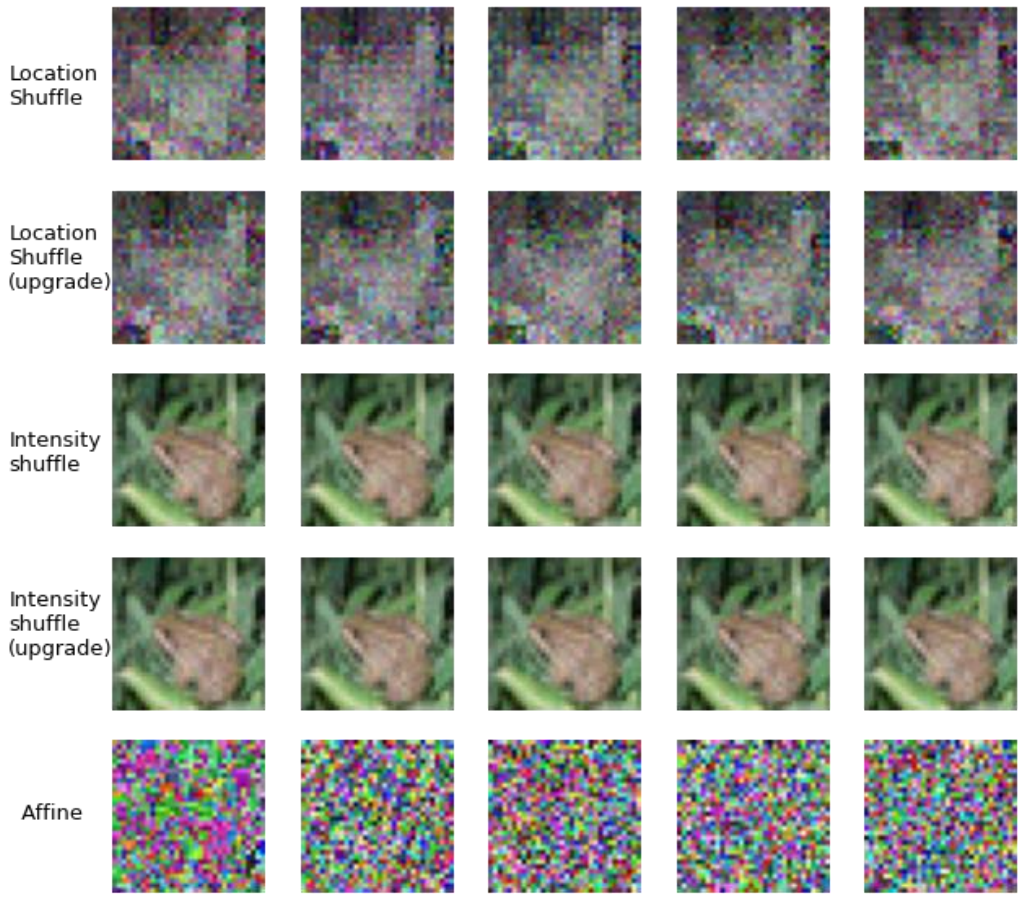


Figure 4. Examples of encrypted images generated by different random seed. For the location block shuffling method, block size 4 was used. For the pixel intensity shuffling method, block size 16 was used. Except the first row, four rows are examples of the proposed methods.

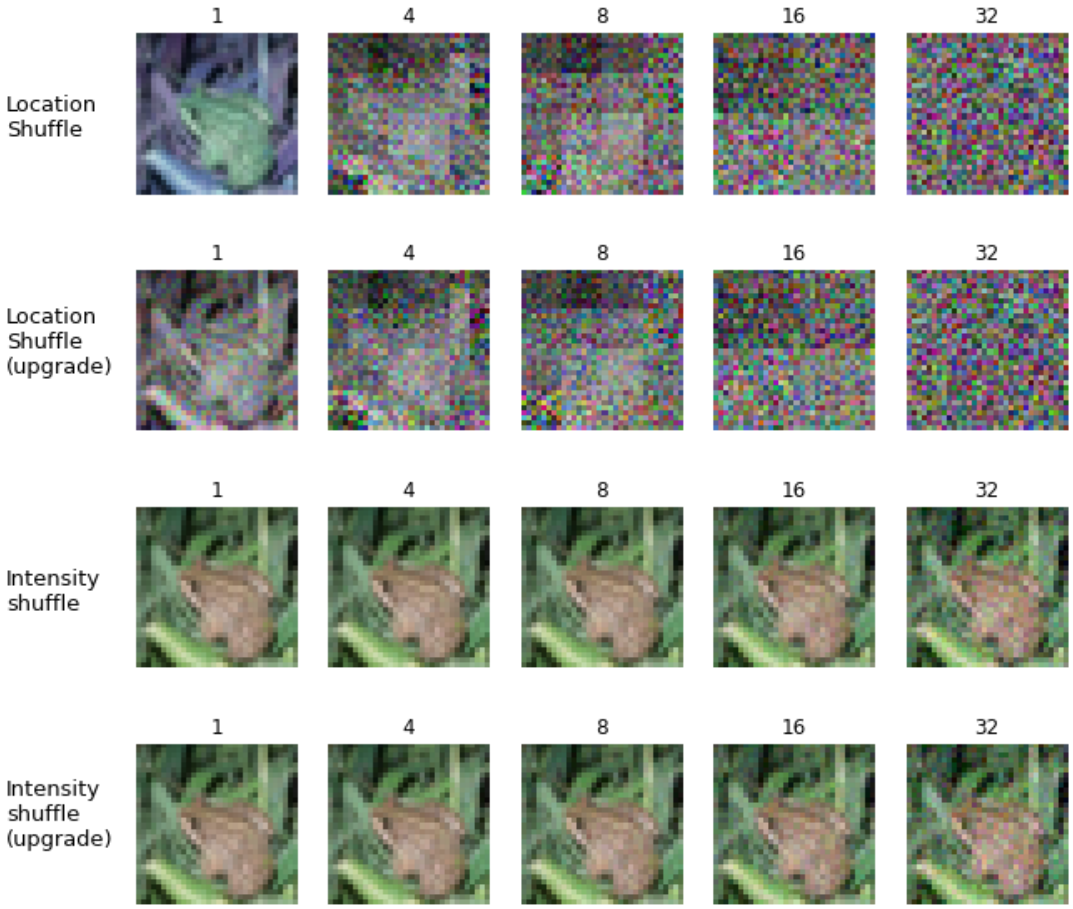


Figure 5. Examples of encrypted images generated by different block size. For the location block shuffling method, block size 4 was used. For the pixel intensity shuffling method, block size 16 was used. Except the first row, three rows are examples of the proposed methods.

4.3 Adaptive attack framework for adversarial encryptiond defense.

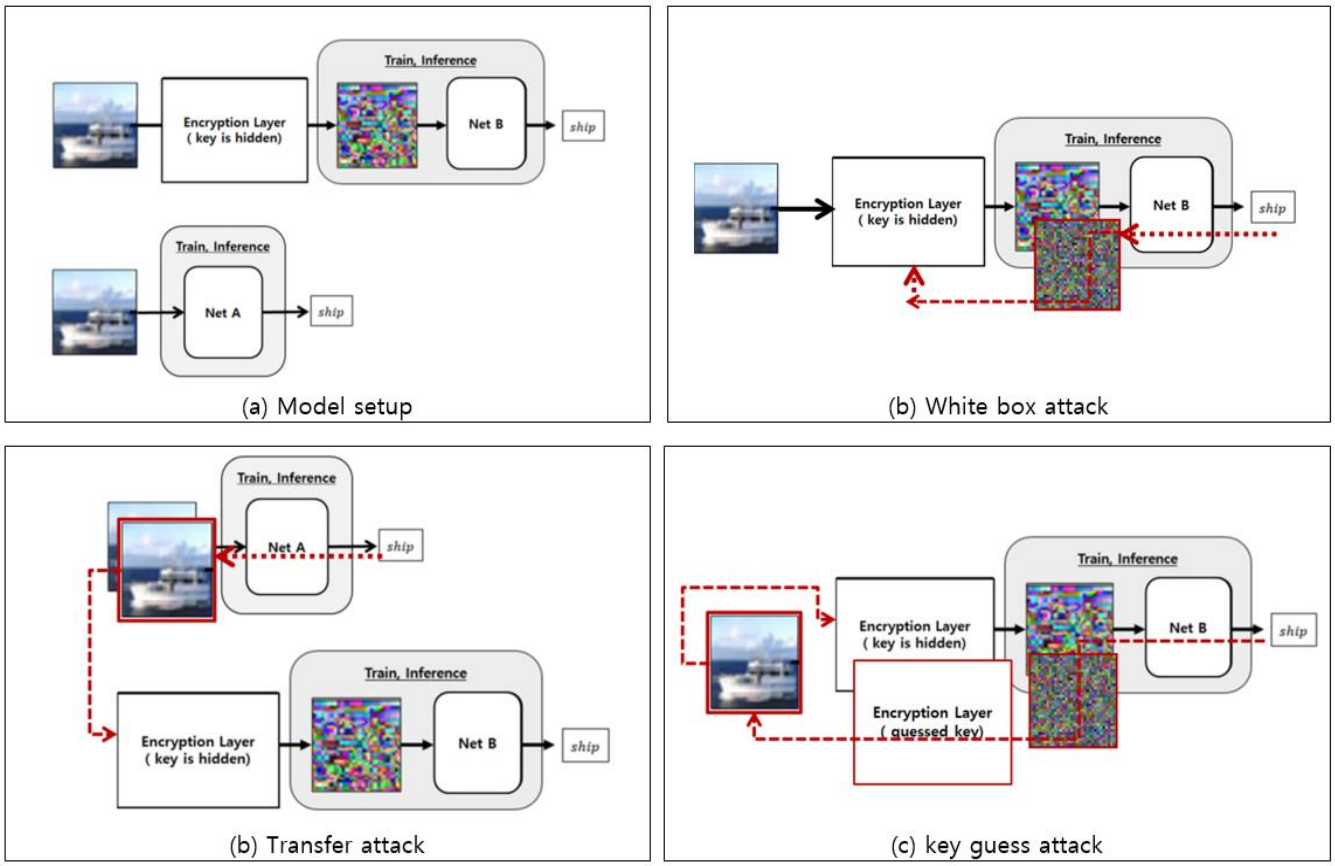


Figure 6. The adaptive attack framework
for adversarial encryption defense

As suggested by Athalye et al. (2018), adaptive attacks are necessary in evaluating adversarial robustness even when white-box gradient attacks are not possible. Many methods that used obfuscated gradients were easily attacked by adaptive attacks since gradients can be approximated easily because the defensively transformed input is

similar to the original input. To ensure and verify the strength of the proposed defense, we designed two adaptive attack scenarios (Figure 5). As pointed by Taran et al. (2018), based on Kerckhoff's cryptographic principle that the key is not known to attackers, the key of the encryption is not part of the model parameters.

We assumed a scenario where attackers have a knowledge of the model weights and the defense algorithms. Experimental setup is described in Figure 5. (a). Network A is trained with natural images (without encryption) and Network B is trained with encrypted images.

The first attack is the identity function(white box) attack which assumes that the defense function is similar to an identify function. A lot of previous methods that used gradient masking technique were easily attacked as their non-differentiable function were similar to an identity function. Thus, we conlcluded it is important to add this attack for all encryption defense methods to investigate if they are robust and do not work as an identity function. The second attack is a transfer attack which makes adversarial examples from Network A and pass them to encryption layer+Network B. This is designed from the natural transferability of an adversarial example which remains effective for the models other than the one used to generate it.

The third attack is a key-guess attack which guesses keys and makes adversarial examples based on the approximated keys. We designed this algorithm based on the cryptographic principle that all details of the proposed algorithms are publicly known and available to the attackers besides the key (Taran et al., 2018). It is important to

have large key space if the defense algorithm is publicly known and this attack evaluates the potential for a key to be guessed more difficult to ensure the strength of the proposed defense.

Algorithm 3 Identity function (white box) attack

Input: encryption algorithm, NetB, dataloader

Output: accuracy, attack success rate

adversarial examples of NetB ← LinfPGDattack(data, NetB, label)

accuracy, attack success rate = Test(adversarial examples of NetB, encryption, NetB)

return accuracy, attack success rate

Algorithm 4 Transfer attack

Input: encryption algorithm, NetA, NetB, dataloader

Output: accuracy, attack success rate

adversarial examples of NetA ← LinfPGDattack(data, NetA, label)

accuracy, attack success rate = Test(adversarial examples of NetA, encryption, NetB)

return accuracy, attack success rate

Algorithm 5 Key guess attack

```
Input: repeat number (N), encryption algorithm, NetB, dataloader
Output: accuracy, attack success rate
    key ← None
    acc ← 0
    for i in range(N):
        Select random key and make guessed encryption layer
        current_acc ← get_accuracy(guessed_encryption, NetB)
        If current_acc > acc:
            Key ← current selected key
    mid_data = guessed_encryption(data)
    adversarial examples of NetB ← LinfPGDattack(mid_data, NetB, label)
    adversarial examples ← guessed_decryption(adversarial examples of NetB)
    accuracy, attack success rate = Test(adversarial examples, encryption, NetB)
    return accuracy, attack success rate
```

Chapter 5. Experiments

To verify the effectiveness of the proposed defense, we designed experiments consisting of two tasks. The first task is designed to investigate how the proposed method affects the learnability of the model and the second experiment is designed to evaluate how robust the proposed method is against adversarial attacks.

5.1 Setup

Dataset and Network

The CIFAR-10 dataset (Krizhevky & Hinton, 2009) was used for the image classification and trained ResNet-18 model (He et al., 2016) using SGD, with minibatches of size 128, momentum of 0.9, weight decay of 0.0005, and maximum learning rate of 0.2. The training setup is the same as that used in previous studies (Aprilpyone et al., 2020) to compare the effectiveness of the proposed defense. The network was trained for 200 epochs with efficient training techniques from the DAWNBench top submissions: cyclic learning rates (Smith et al., 2017) and mixed-precision training (Micikevicius et al., 2017).

Attack Settings

The perturbation range of 0.031 was used to evaluate the robust accuracy under PGD attack for pixels in the range of [0,1]. The PGD attack was configured with a step size of 2/255, 50 iterations, and random initialization.

Evaluation Metrics

We use accuracy to evaluate learnability and Attack Success Rate (ASR) to evaluate robustness. The ASR (Carlini and Wagner 2017b) is defined as the percentage of adversarial examples that are classified as the target class (which is the different from the original class).

Accuracy and ASR are defined as below:

$$\text{Accuracy} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} 1(f_\theta(x_i) = y_i), & \text{(standard accuracy)} \\ \frac{1}{N}\sum_{i=1}^{N} 1(f_\theta(x_i + \delta_i) = y_i), & \text{(robust accuracy)} \end{cases}$$

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

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

The defense method is effective if it results in a higher accuracy and a lower attack success rate.

5.1 Learnability

5.2.1 Experiment Design

We designed the experiment to investigate whether the proposed methods maintain the stable performance on the original dataset although the different key is used. We considered this because the encryption can affect the image with different strength even the same method is used. For example, Affine encryption can have a different strength of encryption depending on the key values and shuffling in

block methods has a different strength of encryption depending on the block size and key values. To investigate the stability of the proposed method on the learnability of the model, we experiment training the network with encrypted images that are made from different keys and block sizes. For affine encryption, we evaluate all possible values for key1 since it has 128 key space. However, we evaluate 20 random seeds for block shuffling and pixel intensity methods as their key space is larger than 12! and assume that if the variation is low enough, it is stable to use those methods. Furthermore, we use the below analyses to compute the strength of encryption and to see the relationship with its learnability.

(a) Information Entropy Analysis

Information entropy is defined as the expression of the degree of uncertainties in the system. An encrypted image is expected to have a uniform distribution of pixel values in an image, making it difficult for the attacker to learn something about the image. The maximum entropy of an 8-bit grayscale image is 8 when all of the pixels are equally distributed, which shows that the information is random. Entropy of an image can be evaluated by Equation (5.1), where n is the number of bits that is required to represent the symbol m_i, and $p(m_i)$ is the probability of symbol n.

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)}, \quad (5.1)$$

(b) NPCR and UACI Analysis

The NPCR (Number of Pixel Change Rate) is the change rate of the absolute number of pixels and the UACI (Unified Average Changing Intensity) computes the average difference of color intensities between two images when the change in one image is subtle. The NPCR and UACI values can be evaluated by Equations (5.3) and (5.4), where T denotes the total number of pixels in the ciphertext, symbol F denotes the largest supported pixel value compatible with the ciphertext image format, and $|\cdot|$ denotes the absolute value function.

$$D(i,j) = \begin{cases} 0, \ if \ \ C^1(i,j) = C^2(i,j) \\ 1, \ if \ \ C^1(i,j) \neq C^2(i,j) \end{cases} \tag{5.2}$$

$$\text{UACI: } \mathrm{U}(C^1, C^2) = \sum_{i,j} \frac{|C^1(i,j) - C^2(i,j)|}{F \cdot T} \times 100\% \tag{5.3}$$

$$\text{NPCR: } \mathrm{N}(C^1, C^2) = \sum_{i,j} \frac{D(i,j)}{T} \times 100\% \tag{5.4}$$

Suppose that small noise is added on P^1 and let this image be named P^2. C^1, C^2 are cipher images of plain images P^1, P^2. Sufficiently high NPCR/UACI scores for C^1, C^2 are usually considered to have a strong encryption strength. The range of NPCR and UACI is [0,1]. When $\mathrm{N}(C^1, C^2) = 0$, it implies that all pixels in C^2 remain the same values as in C^1. When $\mathrm{N}(C^1, C^2) = 1$, it implies that all pixel values in C^2 are changed from those in C^1. To evaluate the NPCR and the UACI in this experiment, 0.03 gaussian noise is added on the original images.

(c) Correlation Coefficient

The Correlation Coefficient assesses the correlation between two

adjoining pixels in an image. Generally, adjacent pixels in plain images have strong correlation. An encrypted image should have low correlation between two adjoining pixels, so that it becomes difficult to guess the value of neighboring pixels. The correlation between adjacent pixels can be measured in the horizontal, vertical, and diagonal orientations. We will compare the result of the average of the correlation coefficients measured in the horizontal, vertical, and diagonal orientations. The correlation coefficient of any two grayscale images can be measured by the following Equation (5.8):

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{5.5}$$

$$\mathrm{Var}(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{5.6}$$

$$\mathrm{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{5.7}$$

$$CorrCoeff(x, y) = \frac{cov(x,y)}{\sqrt{Var(x)}\sqrt{Var(y)}} \tag{5.8}$$

where N is the number of the chosen pixels and x, y are the pixel values of two adjacent pixels.

5.2.2 Experiment Results

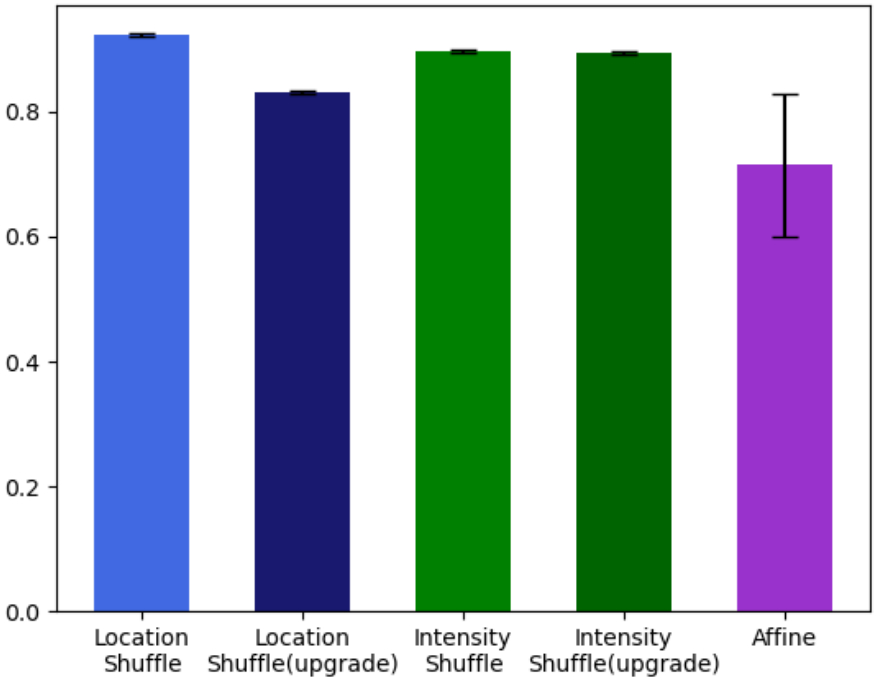


Figure 7. The standard accuracy of different ResNet-18 models trained on CIFAR10 encrypted with different encryption methods. Random 20 seeds were selected to evaluate the stability of four shuffle methods. Different 128 keys were used to evaluate the stablity of the affine encryption.

Figure 9 and Table 1 describes the stability of each methods under different key values. The block shuffling method proposed by Aprilpyone et al. (2020) shows the highest standard accuracy 92.31% and the pixel intensity shuffling method also shows high standard accuracy as 89.63%. If the upgraded encryption is used, the standard

accuracy of the block shuffling method becomes much lower from 92.31% to 83.14% and the pixel intensity shuffling method still remains the high standard accuracy as 89.49%. Although 20 random seeds were selected to evaluate the stability of the pixel location shuffle method which uses block size 4 and the pixel intensity shuffle method which uses block size 16, their lower standard deviation than 0.05% exhibit that it is stable to use them if the location and the intensity shuffle locally in the block. However, the resulf of affine encryption described in Figure 9 presents that it results in significantly different performance on the learnability depending on the value of key, presenting that it is not stable to use the encryption method if it is not processed locally as it has the high standard deviation 11.34% described in Table 1. We conclude not to consider the affine encryption as the adversarial defense because of this problem.

We further experimented the strength of encryption including the correlation coefficient, NPCR and UACI, entropy and the relationship with the learnability to investigate why learnability is different depending on the value of affine key and the size of block size in the pixel location and inentisy shuffling methods.

Defense	Standard accuracy	
	mean	Standard deviation
Location Shuffle (block size=4)	**92.31%**	**0.22%**
Upgraded Locaion Shuffle (block size=4) (ours)	83.14%	0.27%
Intensity Shffle (block size=16) (ours)	**89.63%**	**0.25%**
Upgraded Intensity Shuffle (block size=4) (ours)	**89.49%**	**0.31%**
Affine (ours)	71.46%	11.34%

Table 1.Standard accuracy of different ResNet-18 models trained on CIFAR10 encrypted with different encryption methods. Random 20 seeds were selected to evaluate the stability of the four shuffle methods. 128 values were used to evaluate the stability of the affine encryption.

(a) Information Entropy Analysis

Information entropy of an image is not changed for the proposed

methods and for the block shuffling method proposed by Aprilpyone et al (2020) as they are one-to-one mapping methods.

(b) NPCR and UACI

As one-to-one mapping encryptions are used, the NPCR value remains the same for the proposed methods and the block shuffling method. The UACI value slightly changes depending on key size or value when pixel intensity encryption is used. However, the block shuffling method does not change the intensity of an pixel and only change the location of an pixel, the NPCR value remains the same.

(c) Correlation Coefficient

The Correlation Coefficient assesses the correlation between two adjoining pixels in an image. Figure 7 describes the standard acccuray of the model and the average correlation coefficient of the encrypted images when block shuffling method is used. It shows the linear relationship between the correlation coefficient value and the standard accuracy. Evaluation of the pixel intensity encryption method in regard to the correlation coefficient is described in Figure 8. It also shows the linear relationship between the correlation coefficient value and the standard accuracy. Figure 9. describes the standard accuracy of the model and the correlation coefficient of encrypted imges when different key1 is used in affine encryption. As linear relationship between standard accuracy and the correlation coefficient is exhibited and other strengths of encryptions including entropy, NPCR, UACI remains the same or show slight difference, we concluded that the

correlation coefficient of images play a key role in regard to the learnability of the model

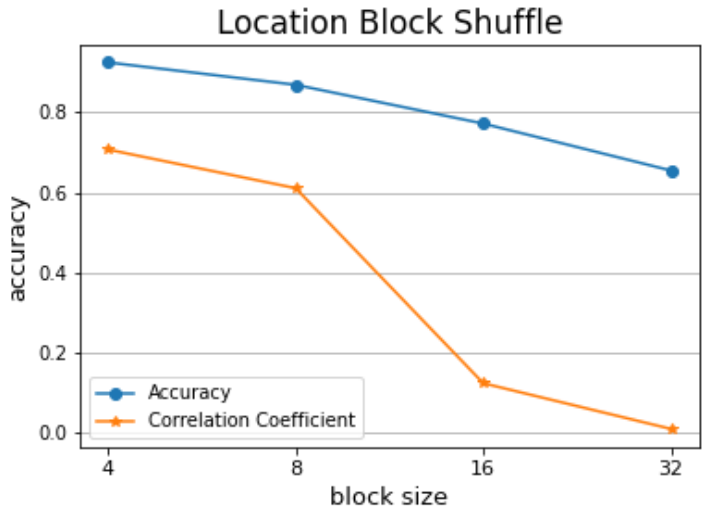


Figure 8. The standard accuracy of different ResNet-18 models trained on CIFAR10 block shuffling encrypted images and their correlation coefficient value depending on the block size

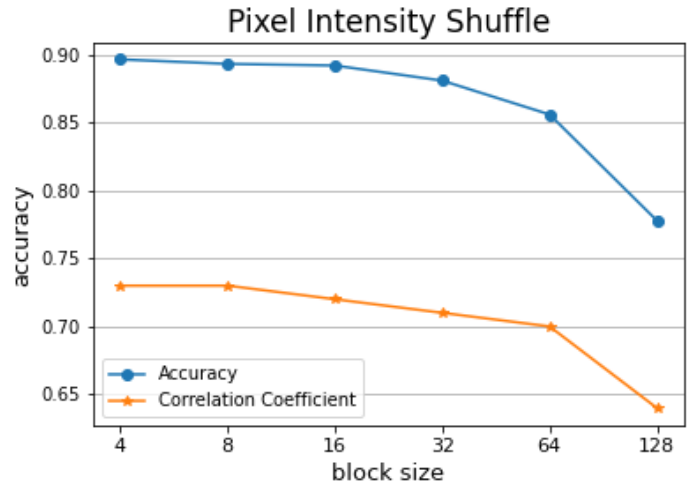
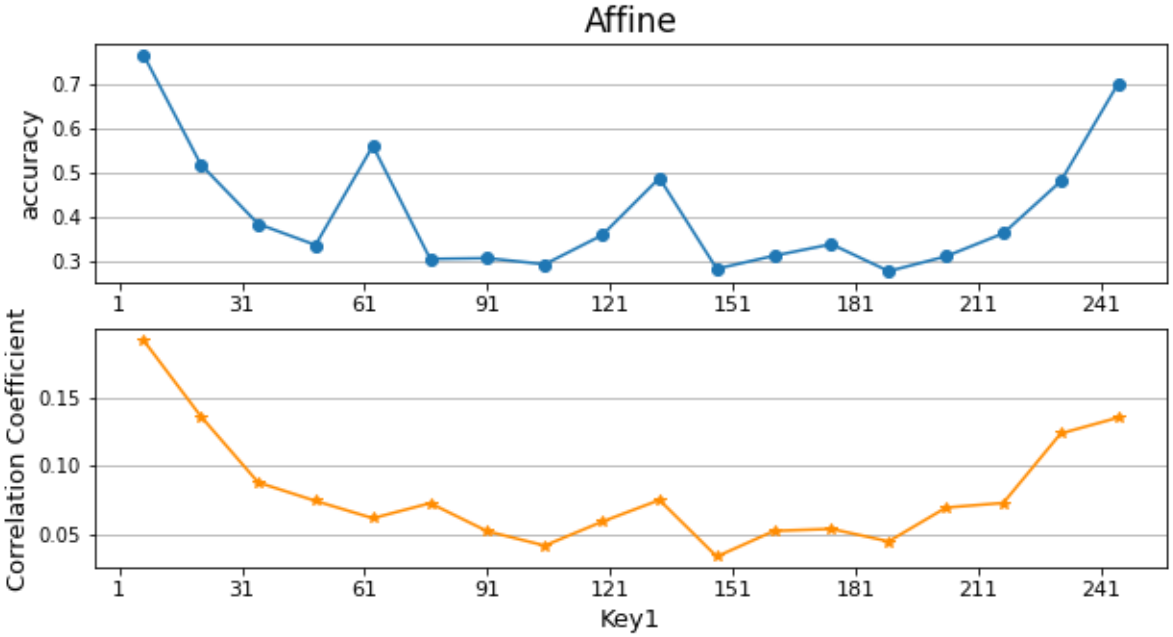


Figure 9. The standard accuracy of different ResNet-18 models trained

on CIFAR10 pixel intensity encrypted images and their correlation



coefficient value depending on the block size

Figure 10. The standard accuracy of different ResNet-18 models trained on CIFAR10 Affine encrypted images and their correlation coefficient value depending on the value of key1. Note that key2 is set to 1

5.3 Adversarial Robustness

.

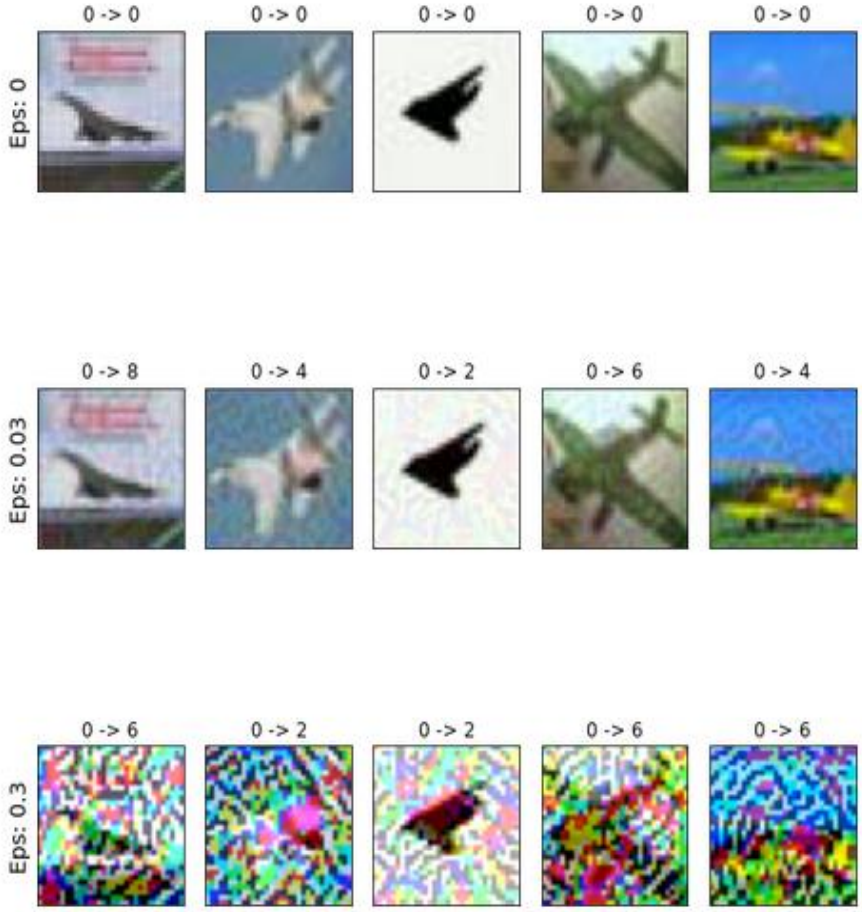


Figure 11. CIFAR10 adversarial examples by various perturbation size. First row describes original image and 0.031 perturbation is used to make adversarial examples in second row, which are used in the experiment

5.3.1 Experiment Design

We designed the adversarial adaptive attack framework to investigate the adversarial robustness of adversarial encryption defense in chapter 4. we experimented 20 random seeds for the block shuffling and the pixel intensity methods to evaluate their mean and standard deviation of the evaluation metrics. For the key-guess attack, we evaluated the repeat number N which decides the number of iteration to guess the key that makes high standard accuracy of 100, 500, 1000 and 1000 was set to evaluate the robustness.

5.3.2 Experiment Results

Examples of CIFAR10 images and its adversarial examples are described in Figure 6. The second row image which has 8/255 adversarial perturbation is used for advesrsarial attack.

We evaluated the robustness of the proposed method on the CIFAR10 dataset under the PGD attack and compared the robustness with block shuffling method proposed by Aprilpyone et al. (2020).

Table 2. summarizes the results of white-box PGD attack on the base ResNet18 model and adversarial trained model, which is the state-of-the-art defense model. We added the result of an identity fuction(white box) attack to compare the robustness under white box attack. The result presents that the base model without any defense method is attacked easily that its high accuracy 95.46% becomes 4.32%

and attack successsses with 95.37%. Although adversarially trained model is considered to be the most successful adversarial defense, the white box attack successes with 53.09%. However, the proposed method and the method suggested by Aprilpyone et al. (2020) show strong adversarial robustness against white box adversarial attack. It demonstrates that the encryption method has strengths if the typical white box attacks are used to attack the model.

Model	Standard accuracy	White-box PGD attack	
		Accuracy	Attack Success Rate
ResNet18	95.46%	4.32%	95.37%
Adversarial Training	85.06%	51.10%	53.09%
Block Shuffling (location block size=4)	92.31% ±0.22%	90.89% ±2.26%	**1.47%** **±1.83%**
PixelIntensity (intensity block size=4) (ours)	89.63% ±0.25%	86.56% ±0.53%	**5.27%** **±0.44%**
Upraded Block Shuffling (location block size=4) (ours)	83.14% ±0.27%	80.65% ±0.25%	**3.12%** **±0.26%**
Upgraded PixelIntensity (intensity block size=4) (ours)	89.49% ±0.31%	86.67% ±0.61%	**4.95%** **±0.55%**

Table 2. Comparison of adversarial robustness to white-box PGD attack on CIFAR10.

Next, we evaluate the proposed method against the transfer attack and the key guess attack. The results are summarized in Table 3. Table 4 describes the attack success rate under 100, 500, 1000 repeat numbers. This implies that a slight change in repeat number doesn't affect the attack success rate significantly if the key space is significantly large. We selected 1000 as the repeat number in the experimentation.

Both the location block shuffling methods and intensity shuffling methods result in high robustness under transfer attack as they result in lower than 10 percent of attack success rate. However, the encryption methods result in significantly different robustness under key guess attack. Although the block shuffling method show the highest standard accuracy, we show that this method can be easily attacked by a brute force key-guess attack as it shows a 78.34 percent of attack success rate under this attack. In contrast, the pixel intensity shuffling method has larger key space and it shows high robustness as it shows around 5 percent of attack success rate under a brute force key-guess attack. This study's upgraded block shuffling encryption has much larger key space than the block shuffling method, and it shows higher robustness under a key-guess attack with 45.98 percent of attack success rate. This study's upgraded pixel intensity method shows the highest robustness under key guess attack as it has the largest key space.

Model	Key space	Transfer attack		Key guess attack	
		Accuracy	Attack Success Rate	Accuracy	Attack Success Rate
Block shuffling (location block size=4)	12!	90%±0.21%	1.94%±0.17%	20.53%±**2**.76%	78.34±2.83%
PixelIntensity (intensity block size=16) (ours)	16!	82.70%±0.24%	8.44%±0.28%	86.41%±0.56%	**5.24±0.56%**
Upgraded Block shuffling (location block size=4) (ours)	64*12!	82.81%±0.27%	0.57%±0.08%	47.22%±0.78%	45.98±1.00%
Upgraded PixelIntensity (intensity block size=16) (ours)	8*16!	82.61%±0.27%	8.75%±0.36%	**86.75%±0.58%**	**4.90 ± 0.58%**

Table 3. Comparison of adversarial robustness to adaptive attacks on CIFAR10. PGD 0.031 perturbation is used on CIFAR10 dataset.

Repeat Number	Key space	100	500	1000
Block shuffling (location block size=4)	12!	78.72%±3.01%	78.23%±2.67%	77.94%±2.75%
Pixel Intensity (intensity block size=16) (ours)	16!	5.17%±0.54%	5.11%±0.57%	5.24%±0.56%

Table 4. Attack Success Rate under random key guess attack with various repeat numbers.

Chapter 6. Discussion

6.1 Discussion

Adaptive attack for adversarial encryption defense

We designed an adversarial adaptive attack framework for the encryption defense method. Many of the previous gradient masking methods were easily attacked because the non-differentiable function of their methods were similar to the identity function. Thus, we propose that it is always necessary to evaluate if the gradient masking layer is similar to the identity function. Similarly, the transfer attack is also possible for investigating if the non differentiable layer causes the similar distribution of the identity function as the adversarial examples has the ability of transferability. Especially for the encryption method, we suggest that the brute-force key guess should be used to evaluate the defense method because their algorithms and network might be publicly known. Thus, it is important to make key of the defense method to be guessed difficult. Based on this idea, we considered to make key space larger than that of the previous methods.

Learnability of the proposed method

To investigate the effects of encryption on the learnability of the model, we varied the value of the key and the block size of the encryption methods. The pixel location shuffling method was already shown to have low standard accuracy if it uses a large block size by

Aprilpyone et al. (2020). For the pixel intensity method, we evaluated the largest block size 256 for affine encryption and showed that it is not stable to use this method as an adversarial defense. It is inferred that if the full block size is used, then the correlation coefficient of the image can be low or high depending on the key value so it affects the learnability of the model depending on the value of the key. However, when we set the block size as 16 and we randomly transformed the intensity of a pixel intensity in the block size, the correlation coefficient of the image does not change significantly and it shows the low variation depending on the value of a key. We concluded that if an image is not encrypted in the local block size in location or in intentiy, it results in the high variation in the learnability of the model. Thus, it is important to find the block size which does not sacrifice or results in high variation of the standard accuracy, but also does not result in the similar distribution of the original image.

Adversarial Robustness of the proposed method

An adaptive attack scenario for encryption methods suggests that encryption methods that have large key space but do not sacrifice the learnability of the model can be used as an adversarial defense. The experiment of adaptive attacks supports the idea. The block shuffling method suggested by Aprilpyone et al. (2020) shows strong robustness against the identity function (white box) attack and the transfer attack as the encryption results in considerably different input distribution from the original image. However, it was exhibited that this method

can be easily attacked by the brute force key guess attack. It is ferred that it was easy to guess the key and make vital adversarial perturbations to this attack. If we upgraded this method using the upgraded method and made the key space larger than that of the original method, then the attack success rate becomes much lower. This result supports that making the encryption that has large key space is important. The pixel intensity shuffling method or upgraded pixel intensity shuffling method have larger key space than those of the location block shuffling method, and our proposed method exhibit the strong robustness against the key guess attack. We infer that shuffling the intensity locally makes the model more robust under small changes.

It is also important to make an image encryption function not to work as an identity function to be robust against the transfer attack and the white box attack. Experimental results show that both of the location block shuffling method and the pixel intensity shuffling method result in high robustness under the transfer attack and the identity function attack. We conclude that they are robust against these attacks as their input distribution to the base network are considerably different from the original input distributions to the base network.

6.2 Limitations and Future Work

We have shown that the proposed method is robust against

gradient based adversarial attack. However, an attacker may perform gradient-free attacks without secret keys when the output of a model is available. Although gradient-free attacks have high computational complexity and they are notoriously slow to attack without any gradient information, but it is also required to make the network robust under any possible attacks.

Explaining the behavior of the deep networks remain challenging due to hierarchical non-linear nature in a black-box fashion and the lack of interpretability raises a severe issue about the trust od deep models, in high-statkes prediction applications, such as autonomous driving, health care, criminal justice, and financial servies (Xuhong et al., 2021). However, the proposed method makes network more difficult to be interpreted. As a result, more studies are required to interpret the distribution of the input and the learnability of the model.

Chapter 7. Conclusion

Motivated by adding encryption layer to the network to defend gradient based adversarial attack, we have demonstrated the new encryption methods that transforms pixel intensity with key that has larger key space than that of the previous studies and demonstrated its robustness under adaptive attacks that we designed.

We presented an attack scenario for general encryption defense methods. Firstly, an identity function (white-box) attack is possible to investigate that the non-differentiable encryption layer works similar to an identity function. Seoncdly, transfer attack is available to evaluate if the defense nework can be attacked easily only by the base network without defense. Lastly, we designed a new adaptive key-guess attack, which brute-forcefully guesses the key of the defense encryption method. It is shown that the method of the previous studies fails under this attack and the proposed method shows robustness to the adaptive attacks. This implies that having larger key guess is important to be robust under adaptive attacks when encryption is used for an adversarial defense.

Furthermore, We find that the learnability is related to the correlation coefficients of images. We analysed the strength of encryption including entropy, NPCR, UACI, and correlation coefficient to investigate why the learnability is different depending on the value of key and the size of the block, and exhibited that the

correlation coeffiecient makes the key effect on learnability.

References

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, July 2018. URL https://arxiv.org/abs/1802.00420.

Olga Taran, Shideh Rezaeifar, and Slava Voloshynovskiy. Bridging machine learning and cryptography in defence against adversarial attacks. in Proceedings of the European Conference on Computer Vision (ECCV), 2018. URL https://arxiv.org/abs/1809.01715

MaungMaung AprilPyone, Hitoshi Kiya. Encryption Inspired Adversarial Defense for Visual Classification. arXiv:2005.07998, 2020.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia conference on computer and communications security, pp. 506–519. ACM, 2017.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean

Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=H1uR4GZRZ.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

MaungMaung AprilPyone, Hitoshi Kiya. An Extension of Encryption-Inspired Adversarial Defense with Secret Keys against Adversarial Examples. 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand, 2020, pp. 1369-1374.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In International Conference on Machine Learning (ICML), 2019b.

Musheer Ahmad, Mohammad Najam Doja, Mirza Mohd Sufyan Beg. Security analysis and enhancements of an image cryptosystem based on hyperchaotic system. J King Saud Univ-ComputInf Sci.

Leslie. N. Smith, Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. arXiv:1708.07120, 2017.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed precision training. In International Conference on Learning Representations.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

Yash Sharma, Pin-Yu Chen. Attacking the madry defense model with L1-based adversarial examples. arXiv preprint arXiv:1710.10733, 2017.

Xuhong Li and Haoyi Xiong and Xingjian Li and Xuanyu Wu and Xiao Zhang and Ji Liu and Jiang Bian and Dejing Dou. Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond. arXiv preprint arXiv 2103.10689, 2021.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018. URL https://openreview.net/pdf?id= S18Su--CW.

Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. 2019. A taxonomy and survey of attacks against machine learning. Elsevier Computer Science Review 34 (2019), 100199.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958, 2014.