Evaluating Adversarial Robustness of Secret Key-Based Defenses

ZIAD TARIQ MUHAMMAD ALI, AMEER MOHAMMED, AND IMTIAZ AHMAD
Department of Computer Engineering, Kuwait University, Safat 13060, Kuwait

Corresponding author: Ameer Mohammed (e-mail: ameer.mohammed@ku.edu.kw).

ABSTRACT The vulnerability of neural networks to adversarial attacks has inspired the proposal of many defenses. Key-based input transformation techniques are the recently proposed methods that make use of gradient obfuscation to improve the adversarial robustness of models. However, most gradient obfuscation techniques can be broken by adaptive attacks that consider the knowledge of the new defense; thus, defenses that rely on gradient obfuscation require a thorough evaluation to identify their effectiveness. Block-wise transformation and randomized diversification are the two recently proposed key-based defenses that claim adversarial robustness. In this study, we developed adaptive attacks and used preexisting attacks against key-based defenses to show that they are still vulnerable to adversarial attacks. Our experiments demonstrate that for a block-wise transformation defense on the CIFAR-10 dataset with the block size of 4, our work can reduce the accuracy of pixel-shuffling to 7.45%, bit-flipping to 4.20%, and feistel-based encryption to 9.45%, as compared to previous work that claims high adversarial robustness. In addition to block-wise transformation, we reduced the accuracy of the randomized diversification method by 25.30% on CIFAR-10.

INDEX TERMS Adversarial Machine Learning, Gradient Obfuscation, Input Transformation, Key-based Transformation, Neural Networks

I. INTRODUCTION

Machine learning classifiers are vulnerable to carefully crafted input perturbations that are designed to fool them into misclassifying the data. This weakness of neural networks was first observed by [1] who showed that noise calculated by maximizing the loss function and reapplying it to the input can cause the neural network to misclassify the input even though the perturbation is visually indistinguishable. Such inputs were later termed adversarial examples because an adversary can use them to degrade the performance of the classifiers, thus, making the classifiers difficult to deploy for security-critical applications.

The vulnerability of neural networks to adversarial examples has attracted many proposals for defense. The majority of defenses are based on input preprocessing [2], adversarial retraining [3], [4], and detection [5], [6]. Most input preprocessing techniques were later shown to be ineffective by [7] owing to the phenomenon of gradient obfuscation, which causes the unavailability of useful gradients. Many defenses that cause gradient obfuscation can be defeated by adaptive attacks, as instead of increasing true robustness against adversarial attacks they make launching the attack difficult. Adaptive attacks are attacks in which the adversary has knowledge of the defense and can often circumvent the defense by designing attacks suited to particular defenses. The most important quality of a defense system is the ability to be adversarially robust against all attacks. If even one attack can degrade the performance, then the defense can be considered nonrobust.

Despite the ineffectiveness of input preprocessing defenses, they were explored and proposed because they are inexpensive and only require the application of a simple transformation to the images before the training process. They use gradient obfuscation to make it difficult for an adversary to find adversarial examples. The key question, however, is whether introducing gradient obfuscation can render finding useful gradients computationally intractable. Recently, key-based input transformation techniques such as block-wise transformation [8] and randomized diversification [9] have been proposed, under which the defender uses a secret key to transform the input images before evaluation to enhance adversarial robustness. The idea is that as long as the attacker does not have access to the secret key, the attack will be difficult or expensive to launch. The large key space ensures that the brute-forcing of the entire key space is computationally intractable considering the transformation.
and prediction costs. We refer the reader to [8] for details of the key space of the block-wise transformations. Nevertheless, this does not eliminate the possibility of development of adaptive attacks that eventually approximate gradients to compromise the performance of such defenses. Moreover, block-wise transformation achieves high accuracy against white-box attacks as long as the key is hidden. However, if the attack is launched with the secret key, the underlying adversarial robustness is 0%, which motivates the development of adaptive attacks.

A. MOTIVATION

Defenses that cause gradient obfuscation have been shown to be ineffective at increasing adversarial robustness [7], [10]. Yet, defense methods that intentionally cause gradient obfuscation were proposed [8], [9], to increase adversarial robustness of models as they are inexpensive in comparison with other defense methods such as adversarial retraining. The authors in the previous works failed to use existing attacks to compromise the adversarial robustness of the proposed secret-key-based models, and consequently, claimed that the defenses are adversarially robust. Because defenses against adversarial attacks are critical as machine learning has applications in various fields, a proper evaluation of defenses is important to eliminate defenses that will not be secure during deployment. Therefore, we evaluate recently proposed secret-key-based defenses that use gradient obfuscation to claim adversarial robustness.

B. CONTRIBUTIONS

In this work, we evaluate the existing key-based defenses, such as block-wise transformation [8] and randomized diversification [9] using adaptive attacks to show that these defenses are still vulnerable to adversarial examples. We propose a technique that decreases the test accuracy under the \( l_\infty \) (max-norm) distance metric for all types of block-wise transformations proposed in the previous work. Similarly, we use a stronger transferability attack to decrease the test accuracy of the randomized diversification defense under the \( l_\infty \) distance metric. Our evaluation further emphasizes that gradient obfuscation based defenses are not secure under the \( l_\infty \) distance metric [7], [10]. The vulnerability of secret-key defenses to adversarial attacks also suggests that they are not suitable for real-time applications as the vulnerability is inherent to the defenses due to gradient obfuscation. Our contributions can be summarized as follows:

1) Attacking and evaluating block-wise transformation methods proposed in [8] and managing to reduce their accuracy by launching attacks that are adapted to their defenses.
2) Evaluating the secret key-based randomized diversification method proposed in [9] and managing to reduce its accuracy under their assumed gray-box threat model.
3) Demonstrating through experiments that secret key based defense techniques that rely on gradient obfuscation are not adversarially robust.

C. ORGANIZATION

The rest of this paper is organized as follows: Section II presents the background on preexisting attacks and threat models. Section III provides a summary of related work. Section IV presents the proposed methodology of attacks. Section V provides the experimental results of the attacks. Section VI provides discussion on the mathematical objectives. Finally, concluding remarks are presented in Section VII.

II. BACKGROUND

Depending on the amount of information an adversary possesses, they may decide to launch attacks that can be categorized as either white-box, gray-box, black-box, or adaptive. Below we describe the classification of attacks depending on the information an adversary has:

1) White-box Attacks: In such attacks, we assume that the adversary has complete knowledge of the model, such as the model parameters, model architecture, training dataset, and learning process. These attacks are usually gradient-based attacks that are applied directly to the model.
2) Gray-box Attacks: Gray-box attacks assume partial knowledge of the model. In a specific case, the work in [9] assumes knowledge of the training dataset, model architecture, and defense used. The only thing the adversary does not have access to are the model parameters.
3) Black-box Attacks: These attacks assume no knowledge of the model. Instead, they either use the property of transferability by [11] and [3] in which adversarial examples that fool one model tend to fool other models with different architecture and training process or use other attacks by using the outputs of the model, such as hard label based attack by [12] or zero-order optimization attacks (ZOO) by [13]. For transferability, typical gradient-based attacks are used on one model and then transferred to a substitute model, resulting in a drop in accuracy of the substitute model.
4) Adaptive Attacks: Some defenses against adversarial attacks introduce nondifferentiable components into the classifier, while others introduce differentiable components. To evaluate defenses correctly, [7] proposed use of adaptive attacks that assume knowledge of the defense. Such defenses require randomized attacks to reduce the performance of defenses. Adaptive attacks can also assume that an adversary has complete access to model information.

A. FAST GRADIENT SIGN METHOD (FGSM)

This method was introduced in [14] and is optimized for the \( l_\infty \) distance metric. This attack was originally developed to support the hypothesis that the adversarial phenomenon
occurs because of the sufficiently high dimensionality of linear and linearly behaving classifiers. Hence, the goal of this attack is to be fast, by generating a simple perturbation, which causes the input to cross the decision boundary. Its nontargeted variant works by maximizing the loss function value between an input instance $x$ and its ground truth label $y_{\text{true}}$. The loss function value is maximized by calculating the gradient of the loss function with respect to the input and adding the sign of the gradient with a chosen step-size as a perturbation to the input instance. For example, the perturbation $\delta$ is calculated using the following equation:

$$\delta = \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y))$$

(1)

where $L$ is the loss function used to train model, $\theta$ represents the parameters such as weights and biases, $x$ is the input instance, and $y$ is the label which can be $y_{\text{true}}$ or $y_{\text{target}}$.

$\nabla_x$ is the gradient of the input $x$ with respect to the cost function. $\text{Sign}$ is an activation function that gives $\text{sign}(z) := \{1, -1\}$. It determines if features in $x$ should be increased or decreased to maximize the loss function $L$. $\epsilon$ controls the percentage of perturbation added to the input. The final adversarial instance is given as:

$$x^{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y))$$

(2)

where the input instance and the adversarial instance are close according to the $l_\infty$ distance metric given by $\|x^{\text{adv}} - x\|_\infty \leq \epsilon$. By using this distance metric, we put a bound that no input feature such as a pixel value could be changed more than $\epsilon$.

**B. PROJECTED GRADIENT DESCENT (PGD)**

This attack is a multistep version of FGSM and is the most widely used attack for adversarial retraining. This attack was proposed in [3]. PGD applies random projection before applying adversarial steps to the input. It is given by the equation:

$$x_n^{\text{adv}} = x$$

(3)

and proceeds with the following equation iteratively:

$$x_{n+1}^{\text{adv}} = \text{Proj}_{x+\delta}(x_n^{\text{adv}} + \alpha \cdot \text{sign}(\nabla_x L(\theta, x_n^{\text{adv}}, y)))$$

(4)

**C. BACKWARD PASS DIFFERENTIABLE APPROXIMATION (BPDA)**

This algorithm was used in [7] to break defenses that cause obfuscated gradients. Such defenses make it difficult to find useful gradients by introducing a nondifferentiable preprocessing layer. However, the authors showed that approximating the gradients for such defenses is sufficient to break them. Such defenses use a classifier $f(.)$ and apply a preprocessing function $g(.)$ to input $x$ such that $x \approx g(x)$. To calculate the gradients, they use forward propagation with $f(g(x))$ and replace $g(.)$ with the identity function on the backward pass. In other words, it means that they approximate $\nabla_x f(g(x))$ by evaluating $\nabla_x f(x)$ at point $g(x)$, which means they replace $x$ with $g(x)$ to overcome the difficulty of back-propagation and then approximate the gradients.

This attack results in breaking of most defenses that use nondifferentiable preprocessors and can be coupled with Expectation Over Transformation (EOT) for a stronger attack.

**D. EXPECTATION OVER TRANSFORMATION (EOT)**

The defenses that use randomized preprocessing can be backpropagated with the transformation function $t$ that they use. To launch effective attack against such defenses, it is important to launch an attack that strongly captures their randomization. To do that, [7] devised EOT attack that optimizes $E_t - T (t(x))$. It uses the average of gradients over multiple transformations as $E_t - T \nabla_x f(t(x))$ to break defenses.

**E. SQUARE ATTACK**

Square attack is a score-based state of the art black-box attack developed in [15]. It uses random search to calculate the adversarial directions, and does not rely on gradient information. Such attacks are useful for evaluating defenses that rely on gradient obfuscation. Moreover, Square attack is also used as a part of AutoAttack [16] which has become a standard for evaluating all recently proposed defenses.

**III. RELATED WORK**

Learning on encrypted images was first proposed in [17] to enable training while preserving privacy. The technique uses secret key-based block-wise transformation on images so that they cannot be interpreted by humans but can easily and effectively train classifiers on the transformed images. Although training with this technique resulted in a slight drop in test accuracy compared with natural training, the models still achieved satisfactory performance. Moreover, to counter the effects of adversarial attacks, [9] proposed a secret key-based randomized diversification defense. Their work showed that this technique can be used effectively on three different datasets: MNIST [18], F-MNIST [19], and CIFAR-10 [20]. However, they only evaluated their defense against Carlini & Wagner (CW) attack [21] but not against stronger attacks proposed such as projected gradient descent (PGD) resulting in a weak evaluation. Subsequent works [22] and [8] used the earlier proposed block-wise transformation technique to counter the effects of adversarial attacks on machine learning. The results suggest that such a technique can be used as an effective defense against white-box attacks, as it successfully causes gradient obfuscation as long as the key is hidden. Block-wise transformation defense has been extended to ensemble voting models [23], transfer learning-based models [24], adversarial detector [25], vision transformers [26], and privacy-preserving DNNs [27], [28], [29], [30], which highlight the importance and various applications of this defense method. Furthermore, [31] proposed secret key-based transformation defense in which the images are transformed using a key-based matrix transform.
A. DEFENSE 1: BLOCK-WISE TRANSFORMATION

Block-wise transformation applies a secret filter to images. In [8], three major block-wise transformations are used: pixel shuffling, bit flipping, and format-preserving encryption. Pixel shuffling transformations first divide the image into blocks, and in each block, shuffle the locations of the pixel values across the image channels using a secret key. The same shuffling permutation key is used for each block of the image. Bit flipping works in a similar manner, but instead of shuffling the values in a block, it subtracts the values of the pixels in the block of an image from 255 based on whether the bit of the secret key block is 0 or 1. Pixel shuffling uses a permutation vector as a secret key whereas bit flipping uses secret key block of 0’s and 1’s generated from a uniform distribution. Finally, format-preserving encryption (FFX) works similarly to bit flipping, but instead of flipping the pixel value, it encrypts the pixel values from the range 0-255 to 0-999 based on the secret key block. The format is preserved, as the range consists of only three digits. Moreover, we outline key takeaways from our evaluation and the evaluation of previous work:

1) Adversary’s Capabilities

The previous work in [8] applied white-box and black-box attacks on the defense to claim adversarial robustness. The attacks failed to considerably reduce the accuracy of the model as they were launched on top of the adversary’s key. They also assume that an adversary has no access to the secret key or model outputs with respect to the secret key.

However, we believe that to properly evaluate a defense, a stronger assumption is required, which is more realistic and aligned with the standard notion of security. The stronger assumption includes providing adversary access to the model’s outputs with respect to the secret key. An adversary with access to model outputs with respect to the secret key is not unreasonable, as it is in line with similar security notions of cryptographic primitives. Therefore, our utilized black-box attack, assumes oracle access to the model, whereas the white-box and transferability attacks only need to have adversarial examples evaluated during the testing time without the need for an oracle. Moreover, the adversary has no access to the secret key used to train the model. The secret key is used only for evaluating the adversarial robustness of the model.

2) Previous Evaluation on Existing Attacks

As demonstrated in previous works [8], the authors were unable to compromise the defense using the existing white-box attacks such as elastic-net attack (EAD) [32], PGD and CW. Such attacks were performed on top of a single key. Furthermore, the authors applied an inverse transformation attack which transforms the image using a random key and applies an attack algorithm to find the adversarial noise. The previous work [8] suggests two methods for white-box attacks: one uses inverse transformation method and the other simply transforms the input with the attacker’s key, and consequently applies an attack algorithm such as PGD. The authors did not provide exact details on how PGD attacks are used in their work for the second method. They only mentioned that the input is first transformed with the attacker’s key, but details regarding how adversarial noise is added to the input are not provided. Therefore, owing to the unavailability of the code of the work in [8], in our evaluation of the defense, we use the inverse transformation attack based on a single iteration of 20-step BPDA-PGD algorithm for bit flipping and pixel shuffling, and consider this to be the evaluation of previous work as it gives similar results.

In case of FFX encryption, the authors point out that the inverse transformation causes significant changes to the input and do not consider it further. For this reason, instead of applying an inverse transformation attack on FFX encryption, we report the accuracy under a single iteration of the BPDA-FGSM attack and consider it as an evaluation of previous work. Nevertheless, our main goal is to show that the test accuracy of the models can be significantly reduced under the \( l_\infty \) threat model, which previous work has failed to demonstrate. Moreover, the authors mention that since inverse transformation attack is not possible on FFX encryption because it causes large changes to the image when it is inversely transformed. In our evaluation, we demonstrate that such an attack is not required. Simply applying the gradient-based noise from the model on clean images iteratively is sufficient to bring the accuracy down.

B. DEFENSE 2: RANDOMIZED DIVERSIFICATION

The defense in [9] uses an ensemble of classifiers to protect against attacks in gray-box threat models. The models in their multi-channel (ensemble) defense use preprocessed images for training and classification. The images are first processed by transformations such as discrete cosine transform (DCT), followed by key-based processing which could be sign flipping or a permutation, and finally an inverse of DCT. The defense claims robustness against attacks in the gray-box settings; therefore, we assume their threat model to attack this defense.

IV. PROPOSED ATTACK METHODOLOGY

A. ADVISER’S CAPABILITY

We launch various attacks based on the white-box and black-box strategies on the block-wise transformation defense. Our white-box attack generates various potential adversarial instances, each based on a different random key for each test example and attempts to find an adversarial example. We report the test accuracy under per example attack success rate as suggested in [33] for our white-box attack by assuming that each key transformation is a different attack on every example. We also used the transferability attack, which is based on PGD, in which adversarial instances are transferred from the surrogate model to the defended model [3]. These attacks require no oracle access and only require the testing examples to be evaluated during testing time. Moreover, the white-box attack we used is a weak attack because it approximates useful gradients and is based on a
weak assumption that model outputs are unavailable to the adversary. On the other hand, the black-box attack assumes oracle access to the model (outputs with respect to the secret key), and thus, is a stronger attack. However, both attacks are able to significantly compromise the defenses.

Randomized diversification uses a gray-box threat model. Therefore, we launch a transferability attack by assuming the attacker has access to model architecture.

**B. UTILIZED ADAPTIVE ATTACKS**

Adaptive attacks refer to attacks that are aware of the defense used. We use two main white-box adaptive attacks suited for attacking block-wise image transformations. The frameworks of the white-box adaptive attacks applied on block-wise transformation defense are explained in Figure 1 and Figure 2. We use BPDA attacks iteratively to make them successful on block-wise transformations. The motivation to apply the BPDA attacks iteratively stems from the observation that applying an inverse transformation attack using a single key may not approximate the true gradients of every input image. However, using a few different keys at random should be able to get us close to our goal of finding true gradients for a high number of input images to compromise the defense. As will be shown in experimental results, brute-forcing the entire key space is not necessary. We only had to try a maximum of 1,000 random keys to significantly compromise the different block-wise transformations. Moreover, for proper evaluation of the defense, we use square attack as a black-box adaptive attack and transferability attack as a gray-box adaptive attack along with aforementioned white-box adaptive attacks.

1) Iterative Backward Pass Differentiable Approximation (I-BPDA)

The standard BPDA works over a transformed input as its goal is to approximate correct gradients over the nondifferentiable transformation. In some cases, identity functions are used to transform the input; however, we find that in the case of block-wise transformation, we can transform the input using a random key and approximate the correct gradients using a regular PGD or FGSM attack over the transformed input. However, evaluation of the defense by previous work is not able to use BPDA to break the defense. Therefore, we propose an iterative backward pass differentiable approximation. We refer to iterative version of BPDA as I-BPDA, in which we transform the clean image with various random keys until we find one that manages to create an adversarial sample. The exact steps for this procedure are explained in Algorithm 1. We utilized this attack particularly for pixel shuffling and bit flipping.

**Algorithm 1 I-BPDA**

| Require: x (input batch), ε (distance), θ (weights), I (iterations), TM (transformation method) |
|-------------------------------------------------|
| 1: for all iterations in I do |
| 2: \(x_t \leftarrow \text{block-wise-transform}(x, key_{random}(t))\) |
| 3: if \(TM == FFX\) then |
| 4: \(grad \leftarrow \text{gradients}(x_t, model(\theta))\) |
| 5: \(adv_x \leftarrow x + (\epsilon \cdot \text{sign}(grad))\) |
| 6: else |
| 7: \(adv_t \leftarrow \text{FGSM/PGD}(x_t, model(\theta))\) |
| 8: \(adv_x \leftarrow \text{block-wise-inverse-transform}(adv_t, key_{random}(t))\) |
| 9: end if |
| 10: end for |
| 11: return \(adv_x\) |
Algorithm 2 I-BPDA+EOT

Require: $x, \epsilon, \theta, I, EOT\!$  
1: for all iterations in I do  
2: \hspace{1em} $grad \leftarrow 0$  
3: \hspace{1em} for all iterations in EOTI do  
4: \hspace{2em} $x_t \leftarrow$ block-wise-transform$(x, key_{random}(I))$  
5: \hspace{2em} $grad \leftarrow grad + \text{gradients}(x_t, model(\theta))$  
6: \hspace{1em} end for  
7: \hspace{1em} $grad \leftarrow \text{average}(grad)$  
8: \hspace{1em} $adv_x \leftarrow x + (\epsilon \cdot \text{sign}(grad))$  
9: \hspace{1em} end for  
10: return $adv_x$

2) Iterative Backward Pass Differentiable Approximation + Expectation Over Transformation (I-BPDA + EOT)  
This attack combines BPDA and EOT. It is used by calculating the average of the gradients generated using BPDA with multiple random keys on input and applying the averaged gradients on either the transformed input or clean input depending on the use case. The exact steps of this procedure are described in Algorithm 2. We utilized this attack particularly for FFX encryption. Although, both proposed algorithms can be used on all block-wise transformations, we observed that I-BPDA+EOT performed better on the FFX defense whereas I-BPDA was the stronger attack on pixel shuffling and bit flipping.

3) Transferability Attack  
We used the property of transferability to attack all defenses considered in this study and report the effectiveness of each defense under this attack. Our version of the transferability attack works by generating adversarial examples using PGD on a naturally trained surrogate model or an adversarially trained surrogate model [3], and transfers them to a defended model. For each defense, we report an attack that is more successful from either transferability from the natural model or the adversarially trained model [3].

4) Square Attack  
Black-box attacks are useful for evaluating defenses that use gradient obfuscation. However, these tend to be more expensive. We use the square attack to evaluate block-wise transformation and use it to compromise the accuracy beyond what could be achieved by the white-box attacks.

5) I-BPDA + Square Attack  
We report the performance of the model on an attack that combines both I-BPDA and square attack to measure the true robustness of the model under the $l_\infty$ distance as suggested in [16]. Both attacks are used independently on test examples and per example accuracy is reported under the two attacks.

C. ATTACKING KEY-BASED TRANSFORMATIONS  
Here, we describe the procedure for applying the key-based input preprocessing methods such as block-wise transformation and randomized diversification, along with the procedure to attack these defenses. The transformations in this section are based on previous works in [8] and [9].

1) Pixel Shuffling  
Because pixel shuffling simply changes the locations of pixels in a block, the transformation can easily be reversed. Therefore, the attack shown in Figure 2 was used. BPDA requires a nondifferentiable transformation to approximate the correct gradients. This allows for the application of a key based transformation based on a random key as an approximation function. The attack calculates the gradients over the transformation using FGSM, and finally inverse transforms the image to obtain an adversarial image, which is then evaluated with the secret key that was previously used in training. Moreover, we use the square attack with 10 restarts.
Table 1: Test accuracy of pixel shuffling defense under our evaluation with $\epsilon = 8/255$, $l_{\infty}$ distance metric and I-BPDA (FGSM) as adaptive attack. Strongest attacks are in bold.

| Model                  | Natural | BPDA-PGD-20 [8] | Transfer Attack | I-BPDA Attack Iterations | I-BPDA (Ours) | Square Attack [15] | I-BPDA + Square |
|------------------------|---------|----------------|-----------------|--------------------------|---------------|-------------------|------------------|
| Pixel Shuffling (M=2)  | 93.79%  | 88.53%         | 67.84%          | 1.000                    | 32.78%        | 4.60%             | 0.82%            |
| Pixel Shuffling (M=4)  | 90.32%  | 89.22%         | 62.80%          | 1.000                    | 70.56%        | 12.57%            | 7.54%            |
| Pixel Shuffling (M=8)  | 78.31%  | 77.28%         | 55.65%          | 1.000                    | 53.47%        | 14.06%            | 2.59%            |
| Pixel Shuffling (M=16) | 53.48%  | 52.20%         | 43.66%          | 1.000                    | 16.64%        | 19.41%            | 0.02%            |

and 1,000 iterations on pixel shuffling and finally report the performance under combined attacks.

2) Bit Flipping

Bit flipping is similar to pixel shuffling because it can also be reversed without causing any major changes to the input in the case of unavailability of correct secret key. Therefore, we used the attack in Figure 2. The nondifferentiable approximation component in this case can be bit-flipping transformation with a random key. We can follow a similar procedure for the attack used for pixel shuffling. We use the square attack on bit flipping with 10 restarts and 1,000 iterations.

3) FFX Encryption

The procedure for attacking FFX encryption transformation is different from bit flipping and pixel shuffling. The reason is that FFX encrypts a number from range 0-255 to 0-999 and changing the pixel value in the range 0-999 can result in major changes if the number is decrypted back to the 0-255 range, and this can be greater than $\epsilon$ limit set for the perturbation. Therefore, instead of applying the gradients on the transformed image, in each step for either FGSM or PGD we apply the gradients on the clean image to make it adversarial. The framework of the attack is illustrated in Figure 1. Consequently, PGD or FGSM based on BPDA we apply the gradients on the clean image to make it adversarial. The framework of the attack is illustrated in Figure 1. Consequently, PGD or FGSM based on BPDA can be applied and BPDA can be coupled with EOT. Furthermore, we used square attack on FFX encryption with 1 restart and 1,000 iterations. A lower number of restarts is used as FFX encryption is a slightly more expensive transformation. Therefore, we only use the square attack on the correctly predicted remaining examples from the white-box attacks.

4) Randomized Diversification

This defense assumes a gray-box setting. Therefore, we cannot apply a white-box attack. However, because their gray-box scenario assumes that model architecture information is available to the adversary, we can use the property of transferability to attack their defense. For this defense, we simply used a transferability attack based on PGD from a naturally trained model. It works by generating adversarial examples on a surrogate model and transfers them to the multichannel aggregator.

V. EXPERIMENTAL RESULTS

We used a simple ResNet-18 model by [34] and trained the model on the CIFAR-10 dataset with stochastic gradient descent (SGD) as an optimizer, with an initial learning rate of 0.01, which was divided by 10 after 75th, 90th and 100th epoch for a total of 120 epochs. We also used common data augmentation techniques such as horizontal flipping and random cropping. The code we used to train our models was available in [35]. Moreover, we report the results under the strongest attack that worked on each defence. We evaluated the block-wise transformations method using the test set of CIFAR-10. For the transferability attack, we used a simple ResNet-18 model, trained it on clean CIFAR-10 data, and generated adversarial examples on it by applying PGD attack on the clean test set. The test accuracy of the model for the clean test set was 94.98%. For the PGD and FGSM attacks on CIFAR-10, we used an $\epsilon = 8/255$ for the $l_{\infty}$ distance metric.

For randomized diversification, we used the code and models made publicly available by [9]. Moreover, none of the attacks that we launched took more than 24 hours; they were efficient enough to be easily launched with an entire testing set.

A. EVALUATION METRICS

In general, we measure the attack success rate (ASR) on block-wise transformation and randomized defenses as:

$$ASR = \frac{1}{N} \sum_{i=1}^{N} 1(f(x_i^{adv}) \neq y_i) \quad (5)$$

where $N$ is the number of test images, $f(\cdot)$ is the model, 1 is one if condition is true, $y_i$ is the true label of the input and $x_i^{adv}$ is the adversarial instance generated by some attack algorithm such as the square or the transferability attack for...
Table 2: Test accuracy of bit flipping defense under our evaluation with $\epsilon = 8/255$, $l_\infty$ distance metric and I-BPDA (PGD) as adaptive attack. Strongest attacks are in bold.

| Model                  | Natural | BPDA-PGD-20 [8] | Transfer Attack | I-BPDA Attack Iterations | I-BPDA (Ours) | Square Attack [15] | I-BPDA + Square |
|------------------------|---------|-----------------|-----------------|--------------------------|---------------|---------------------|-----------------|
| Bit Flipping (M=2)     | 94.55%  | 81.37%          | 39.95%          | 100                      | 12.55%        | 3.62%               | 0.22%           |
| Bit Flipping (M=4)     | 93.00%  | 86.75%          | 68.47%          | 50                       | 29.52%        | 8.66%               | 4.20%           |

the $i$th input. Moreover, the test accuracy (ACC) is given as:

$$ACC = (1 - ASR) \times 100$$  \(6\)

whereas, we measure the attack success rate of our proposed white-box attacks on block-wise transformation as:

$$ASR = \frac{1}{N} \sum_{i=1}^{N} \left( f(x_{adv}^{i}) \neq y_i \lor \ldots \lor f(x_{adv}^{N}) \neq y_N \right)$$  \(7\)

where $x_{adv}^{i}$ is the adversarial instance generated by our proposed adaptive white-box attacks using key $k_1$, and $m$ is the number of iterations of the attack algorithm.

### B. PIXEL SHUFFLING

We implemented pixel shuffling using the code in [17] as used originally by [22] and [8]. For all four values of block size $M$, we report the accuracy under I-BPDA attack coupled with single step FGSM algorithm. The results are reported in Table 1. Adaptive + square attacks refers to the use of both attacks to find an adversarial counterpart for a given image. According to the results, the test accuracy under I-BPDA and square attack is significantly reduced i.e. under 8% for each block size. The accuracy was significantly lower than that reported in the original work in [8]. For comparison with previous work, we used BPDA-PGD-20 which refers to the inverse transformation attack using PGD with 20 steps. We added it to the table to compare previous work with our contribution. As shown, BPDA-PGD-20 does not cause the accuracy of the model to drop significantly, and this is because the attack fails to find useful gradients. However, the attacks we used were successful which suggests that they diminish the effects of gradient obfuscation. Moreover, the adaptive attack iterations correspond to the number of different keys used to generate potential adversarial examples for each image.

### C. BIT FLIPPING

Because the code for bit flipping was not publicly available, we implemented it based on the algorithm for bit flipping. We report the test accuracies for two block sizes, 2 and 4. Transformations with the block size 8 and 16 were already vulnerable according to the original work. In our implementation of attacks on bit flipping, we used an I-BPDA (PGD) attack, in which at each step of PGD we transform the input with a unique random key and inversely transform before the next step. The results are reported in Table 2. BPDA-PGD-20 refers to the inverse transformation attack using PGD with 20 steps, and is based on previous work.

According to the results, the performance of the model was significantly reduced under our evaluation. According to the evaluation method used in [8], the accuracy of the model under inverse transformation attack (BPDA-PGD-20) was 81.37% for the block size $M=2$ and 86.75% for the block size $M=4$, respectively. In comparison, the attacks we used resulted in accuracy of 0.22% for $M=2$ and 4.20% for $M=4$, rendering this defense transformation ineffective. As previously reported, BPDA-PGD-20 does not cause the accuracy of the model to drop significantly.

### D. FFX ENCRYPTION

We also had to implement FFX encryption transformation as the code was not available. We use iterative BPDA + EOT with 30 different keys by averaging the gradients for attacking FFX encryption. We also use transferability and square attack. The accuracy of FFX encryption defense drops to 3.46%, 9.45%, 12.25%, and 13.11% for the block size of 2, 4, 8, and 16, respectively. The results are reported in Table 3.

### E. RANDOMIZED DIVERSIFICATION

Because randomized diversification claims robustness in gray-box setting that assumes that the model architecture is available to the adversary, we launched a transferability attack on both MNIST and CIFAR-10 datasets. We used the publicly available code, and in their code they have CIFAR-10 and MNIST pretrained DCT domain models available. Whereas we train models with same architecture that they used for pretrained models, and generate adversarial examples using PGD and then transfer them to the pretrained models. To make a fair comparison with previous work [9], we used the first 1,000 inputs from both datasets for evaluation.

For CIFAR-10, we use an $\epsilon = 8/255$ for $l_\infty$ distance metric and for MNIST we use $\epsilon = 0.3$ for $l_\infty$ distance metric. The results are reported in Table 4. Based on the results, the accuracy dropped around 20% for CIFAR-10 model which is lower than the accuracy reported in the original work under the Carlini & Wagner (CW) attack. Under CW attack the
Table 3: Test accuracy of FFX encryption defense under our evaluation with $\epsilon = 8/255$, $l_{\infty}$ distance metric and I-BPDA+EOT (FGSM) as adaptive attack. Strongest attacks are in bold.

| Model               | Natural | BPDA-FGSM [8] | Transfer Attack | I-BPDA Attack Iterations | I-BPDA (Ours) | I-BPDA + Square |
|---------------------|---------|---------------|-----------------|--------------------------|---------------|-----------------|
| FFX encryption (M=2)| 92.43\% | 84.26\%       | 73.5\%          | 63                       | 29.07\%       | 3.46\%          |
| FFX encryption (M=4)| 92.25\% | 89.82\%       | 72.23\%         | 53                       | 54.00\%       | 9.45\%          |
| FFX encryption (M=8)| 92.21\% | 90.63\%       | 71.27\%         | 85                       | 37.65\%       | 12.25\%         |
| FFX encryption (M=16)| 91.50\% | 90.15\%       | 70.17\%         | 100                      | 30.66\%       | 13.11\%         |

Table 4: Test accuracy of randomized diversification with DCT permutation and 3 sub-bands method under our evaluation with PGD as attack type and $l_{\infty}$ as distance metric. Strongest attacks are in bold.

| Dataset  | Channels (I) | Natural | CW [21]         | Transfer Attack (Ours) |
|----------|--------------|---------|-----------------|------------------------|
| MNIST    | 3            | 99.50\% | 92.00\%         | 44.00\%                |
| MNIST    | 5            | 99.40\% | 93.00\%         | 44.20\%                |
| CIFAR-10 | 3            | 79.20\% | 75.00\%         | 52.40\%                |
| CIFAR-10 | 5            | 79.30\% | 76.10\%         | 54.00\%                |

accuracy only drops by 3-4\% based on our experiments, and the results reported in previous work. Moreover, the accuracy dropped around 50\% for the MNIST models, which is significantly lower compared to the CW attack used in previous work [9]. Since this is a gray-box setting, our goal is not to break the defense completely but partially, which we do so successfully. Moreover the robust accuracy obtained using Madry’s adversarial retraining performs significantly better under black-box attacks compared to randomized diversification on both MNIST and CIFAR-10 datasets which shows that the defense is ineffective.

VI. DISCUSSION

Based on the results of our experiments, block-wise transformation and randomized diversification are not secure as they cause gradient obfuscation which is a weak form of defense against adversarial attacks. Breaking these defenses required the development of strong attacks which can estimate useful gradients. In this section, we provide an explanation for the success of our attacks on block-wise transformation defense. Consider $t_k$ to be a block-wise transform with random key $k$, then the transformed input is given as:

$$x_k = t_k(x) \quad (8)$$

then the objective used for creating an adversarial example using an inverse transformation attack in the previous work [8] is given as:

$$x_{k, inv}^{adv} = inv_{k_i}(x_{k_i} + \epsilon \text{ sign}(\nabla_{x_{k_i}} L(\theta, x_{k_i}, y))) \quad (9)$$

where $inv_{k_i}$ is the inverse block-wise transformation with the same random key $k_i$, and $y$ is the true label of the input instance. This attack is based on a single random key $k_1$ and is not able to approximate useful gradients successfully. On the other hand, our objective is an accumulation of (9) based on different random keys. It finds an adversarial instance as:

$$x_{k}^{adv} \in \{f((t_i(x_{k, inv}^{adv}))) \neq y\}_i \quad (10)$$

where $t_i$ is the transformation with the secret key that was used to train the model, $i$ is the iterations of the attack algorithms, namely I-BPDA and I-BPDA+EOT. This shows that our attack creates multiple adversarial instances of an input by using various random keys and only selects the one that manages to fool the model. Moreover, one random key might approximate useful gradients better than a different random key, which makes using different random keys effective for the success of the attack. Furthermore, the time complexity of the inverse transformation attack proposed by previous work is $O(Nc_f(x)c_{inv}(x))$, where $c_f(x)$ is the block-wise transformation cost for a single image, $c_{inv}(x)$ is the block-wise inverse transformation cost for a single image, and $N$ is the number of images. In comparison, the time complexity of our proposed algorithm is $O(NKc_f(x)c_{inv}(x))$, where $K$ is the number of keys used. Although our proposed algorithm...
is expensive by a factor of $K$, it performs significantly better than the previously proposed inverse transformation attack.

VII. CONCLUSION

Key-based input transformation is a recently proposed technique for defending against adversarial attacks. The goal of key-based input transformation is to introduce a secret key into the model that cannot be guessed by an adversary. This secret ensures gradient obfuscation and does not allow an adversary to find useful gradient directions. However, all gradient obfuscation techniques can be defeated by adaptive attacks, which are attacks that consider the defense and are aware of which defense is being used. Therefore, in our evaluation we developed new adaptive attacks and used existing attacks that are able to break the defenses and significantly reduce the accuracy of block-wise transformation defense under white-box and black-box settings. It is important to note that using adaptive white-box attacks that are based on a weak assumption results in weaker attacks. However, using a strong assumption that allows an adversary access to the model outputs with respect to secret key results in stronger attacks that manage to break the block-wise transformation defense. Our work can reduce the accuracy for all 3 block-wise transformation methods, namely pixel-shuffling, bit-flipping and FFX encryption. Moreover, the second key-based input transformation defense called randomized diversification also claims robustness in gray-box settings. Our evaluation of randomized diversification shows that its accuracy can be reduced under a strong PGD-based transferability attack. Our work offers evaluations that reduce the accuracy of key-based input transformation defenses and shows that they cannot be used to secure machine learning systems.

VIII. ACKNOWLEDGMENTS

We thank Heba Elhadayd for proofreading the manuscript.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[2] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, “Countering adversarial images using input transformations,” in International Conference on Learning Representations, 2018.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladiu, “Towards deep learning models resistant to adversarial attacks,” in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.

[4] A. Kurakin, I. J. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” arXiv preprint arXiv:1607.02533, 2016.

[5] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” arXiv preprint arXiv:1704.01155, 2017.

[6] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, “A survey on adversarial attacks and defences,” CAAI Transactions on Intelligence Technology, vol. 6, no. 1, pp. 25–45, 2021.

[7] A. Athalye, N. Carlini, and D. A. 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, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 274–283.

[8] M. Aprilpyone and H. Kiya, “Block-wise image transformation with secret key for adversarially robust defense,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2709–2723, 2021.

[9] O. Taran, S. Rezaeifard, T. Holotyak, and S. Voloshynovskiy, “Defending against adversarial attacks by randomized diversification,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 11 226–11 233.

[10] F. Tramer, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example defenses,” arXiv preprint arXiv:2002.08347, 2020.

[11] N. Papernot, P. McDaniel, and I. Goodfellow, “Transferability in machine learning: from phenomena to black-box attacks using adversarial samples,” arXiv preprint arXiv:1605.07277, 2016.

[12] M. Cheng, T. Le, P. Chen, H. Zhang, J. Yi, and C. Hsieh, “Query-efficient hard-label black-box attack: An optimization-based approach,” in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

[13] 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 Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, 2017, pp. 15–26.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[15] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient hard-label black-box adversarial attack via random search,” in European Conference on Computer Vision. Springer, 2020, pp. 484–501.

[16] F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” in International Conference on Machine Learning. PMLR, 2020, pp. 2206–2216.
[17] M. Tanaka, “Learnable image encryption,” in 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). IEEE, 2018, pp. 1–2.

[18] Y. LeCun. (1998) The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Online; accessed 5-May-2021.

[19] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” arXiv preprint arXiv:1708.07747, 2017.

[20] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images.” Citeseer, 2009.

[21] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 39–57.

[22] M. Maung, A. Pyone, and H. Kiya, “Encryption inspired adversarial defense for visual classification,” in 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020, pp. 1681–1685.

[23] A. MaungMaung and H. Kiya, “Ensemble of key-based models: Defense against black-box adversarial attacks,” in 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE). IEEE, 2021, pp. 95–98.

[24] M. AprilPyone and H. Kiya, “Transfer learning-based model protection with secret key,” in 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021, pp. 3877–3881.

[25] T. Osakabe, M. Aprilpyone, S. Shiot, and H. Kiya, “Adversarial detector with robust classifier,” arXiv preprint arXiv:2202.02503, 2022.

[26] T. Chuman and H. Kiya, “Security evaluation of block-based image encryption for vision transformer against jigsaw puzzle solver attack,” arXiv preprint arXiv:2202.00806, 2022.

[27] H. Ito, Y. Kinoshita, M. Aprilpyone, and H. Kiya, “Image to perturbation: An image transformation network for generating visually protected images for privacy-preserving deep neural networks,” IEEE Access, vol. 9, pp. 64 629–64 638, 2021.

[28] T. Chuman and H. Kiya, “Block scrambling image encryption used in combination with data augmentation for privacy-preserving dnnls,” in 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). IEEE, 2021, pp. 1–2.

[29] M. AprilPyone and H. Kiya, “A protection method of trained cnn model with a secret key from unauthorized access,”,”” APSIPA Transactions on Signal and Information Processing, vol. 10, p. e10, 2021.

[30] H. Kiya, A. MaungMaung, Y. Kinoshita, I. Shoko, and S. Shiot, “An overview of compressible and learnable image transformation with secret key and its applications,” arXiv preprint arXiv:2201.11006, 2022.

[31] Y. Qin and C. Yue, “Key-based input transformation defense against adversarial examples,” in 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, 2021, pp. 1–10.

[32] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, “Ead: elastic-net attacks to deep neural networks via adversarial examples,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018.

[33] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, “On evaluating adversarial robustness,” arXiv preprint arXiv:1902.06705, 2019.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[35] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, “Improving adversarial robustness requires revisiting misclassified examples,” in International Conference on Learning Representations, 2019.
IMTIAZ AHMAD received the B.Sc. degree in Electrical Engineering from the University of Engineering and Technology at Lahore, Pakistan, the M.Sc. degree in Electrical Engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and the Ph.D. degree in Computer Engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1988, and 1992, respectively. Since 1992, he has been with the Department of Computer Engineering, Kuwait University, Kuwait, where he is currently a Professor. His research interests include the design automation of digital systems, parallel and distributed computing, machine learning and software-defined networks.

***