COMPUTERS & SECURITY 78 (2018) 380–397

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Friend-safe evasion attack: An adversarial example that is correctly recognized by a friendly classifier☆

Hyun Kwon[a], *Yongchul Kim*[b], *Ki-Woong Park*[c], *Hyunsoo Yoon*[a], *Daeseon Choi*[d,*]

[a] *School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea*
[b] *Department of Electrical Engineering, Korea Military Academy, Seoul 01819, South Korea*
[c] *Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea*
[d] *Department of Medical Information, Kongju National University, Gongju-si 32588, South Korea*

ARTICLE INFO

Article history:
Received 16 January 2018
Revised 25 May 2018
Accepted 21 July 2018
Available online 10 August 2018

Keywords:
Deep Neural Network
Evasion Attack
Adversarial Example
Covert Channel
Machine Learning

ABSTRACT

Deep neural networks (DNNs) have been applied in several useful services, such as image recognition, intrusion detection, and pattern analysis of machine learning tasks. Recently proposed adversarial examples-slightly modified data that lead to incorrect classification-are a severe threat to the security of DNNs. In some situations, however, an adversarial example might be useful, such as when deceiving an enemy classifier on the battlefield. In such a scenario, it is necessary that a friendly classifier not be deceived. In this paper, we propose a friend-safe adversarial example, meaning that the friendly machine can classify the adversarial example correctly. To produce such examples, a transformation is carried out to minimize the probability of incorrect classification by the friend and that of correct classification by the adversary. We suggest two configurations for the scheme: targeted and untargeted class attacks. We performed experiments with this scheme using the MNIST and CIFAR10 datasets. Our proposed method shows a 100% attack success rate and 100% friend accuracy with only a small distortion: 2.18 and 1.54 for the two respective MNIST configurations, and 49.02 and 27.61 for the two respective CIFAR10 configurations. Additionally, we propose a new covert channel scheme and a mixed battlefield application for consideration in further applications.

1. Introduction

Today, deep neural networks (DNNs) (Schmidhuber, 2015) are widely used for image recognition (Simonyan and Zisserman, 2015a), speech recognition (Hinton et al., 2012), intrusion tolerance (Potluri and Diedrich, 2016), natural language processing (Collobert and Weston, 2008), and game-playing (Silver et al., 2016). The security and safety of neural networks and machine learning receive considerable attention from the security research community. Szegedy et al. (2014) presented adversarial examples in image classification; in an evasion attack, images that are transformed slightly can be incorrectly classified by a machine learning classifier, even when the changes are so

☆ A preliminary version of this paper was presented at the ICISC 2017 conference.

* Corresponding author.
E-mail addresses: khkh@kaist.ac.kr (H. Kwon), kyc6454@kma.ac.kr (Y. Kim), woongbak@sejong.ac.kr (K.-W. Park), hyoon@kaist.ac.kr (H. Yoon), sunchoi@kongju.ac.kr (D. Choi).

https://doi.org/10.1016/j.cose.2018.07.015
0167-4048/

small that a human cannot easily recognize them. Such an attack can cause a self-driving car to perform an unwanted action, provided a slight change is made to a road sign (McDaniel et al., 2016). Countermeasures against these attacks have been proposed (Kurakin et al., 2017b; Papernot et al., 2016b; Tramèr et al., 2018), and subsequently, more advanced attacks were developed to defeat the countermeasures.

Evasion attacks can be utilized in several domains, including those of military strategy. An adversarial example can be used to deceive an enemy's machine classifier. For example, a battlefield road sign can be modified to deceive an enemy's self-driving vehicle. If the battlefield is shared by enemy and friendly forces, friendly self-driving vehicles should not be deceived by the attack. Therefore, we require an evasion attack scheme that can deceive the enemy while protecting friendly forces.

In this paper, we propose an evasion attack scheme that creates an adversarial attack that is incorrectly classified by an enemy classifier and correctly recognized by a friendly classifier. The proposed scheme has two class configurations: targeted and untargeted. In the targeted class, a transformer changes the original sample to be recognized as a specific target class. In the untargeted class, the goal of the transformation is incorrect classification to any class other than the right class.

We evaluate our scheme on a standard MNIST dataset (LeCun et al., 2010), a digit recognition task (0–9), and a CIFAR10 dataset (Krizhevsky et al., 2014), with 10 color image classes. We use a defensive state-of-the-art anti-evasion distillation classifier (Papernot et al., 2016b) as an enemy classifier. In the experiment, we demonstrate an adversarial example that is both 100% successful in deceiving the enemy classifier and 100% correctly recognized by a friendly classifier. The distortion of the original sample is held similar to that of the state-of-the-art evasion attack scheme (Carlini and Wagner, 2017b) to maintain human recognition. This study is an extension of our previous work (Kwon et al., 2017). A preliminary version of this paper was presented at the ICISC (International Conference on Information Security and Cryptology) 2017 conference. For ICISC 2017, in which we focused on ideas and concepts for generating a friend-safe adversarial example. This paper makes the following contributions:

- We systematically organize the framework of the proposed scheme. For example, we describe systematic principles in more detail and verify the usefulness of the covert channel scheme while extending the CIFAR10 and MNIST datasets to allow the possibility of evaluating the security of multiple scenarios (Carlini and Wagner, 2017a). We also analyze various aspects of the proposed scheme and extend its utility to other areas. Unlike the conventional method, which uses only distortion, experimental human recognition is added for MNIST, CIFAR10, and covert channels.
- We apply our scheme to the anti-evasion classifier (Papernot et al., 2016b) and are 100% successful in deceiving it. Simultaneously, our friendly classifier achieves 100% accuracy without any modification or retraining. We learn that it is possible to achieve both objectives simultaneously while maintaining low distortion.
- We analyze distortion differences between the targeted and untargeted schemes, including the differences among targeted digits. Because the distortion is related to the possibility of unveiling a human attack, this analysis will be useful for attack planning. Additionally, the results of experiments with human recognition tests show that the distortion by the proposed method does not substantially degrade human recognition of test data.
- We propose a new covert channel scheme (Smeets and Koot, 2006) as another application, in which friend and enemy roles are reversed. The target class of an adversarial example is the hidden information transferred via the covert channel. Experimental results confirm the usefulness of the proposed covert channel.

The remainder of this paper is structured as follows: In Section 2, background and related work on machine learning attacks are presented. The problem definition is introduced in Section 3. In Section 4, the proposed friend-safe adversarial example generation scheme is introduced. Results and findings of experiments with the proposed scheme are presented in Section 5. The new covert channel scheme using the proposed method is presented in Section 6. The proposed scheme is discussed in Section 7. Finally, Section 8 concludes this paper.

2. Background and related work

Barreno et al. (2010) discussed several machine learning security issues, categorizing attacks into causative and exploratory attacks, which respectively influence learning with control over training data and exploit misclassifications without affecting training.

As an example of a causative attack, a poisoning attack with added malicious training data was proposed by Biggio et al. (2012), Mozaffari-Kermani et al. (2015), and Yang et al. (2017). Although poisoning attacks are effective, they require that the attacker access training data while they are being used to train a victim model. This assumption is unrealistic, so poisoning attacks are not considered a severe threat to machine learning applications.

For an exploratory attack, Szegedy et al. (2014) first presented the adversarial example, in which an attacker slightly transforms an image. The main goal of using an adversarial example is to cause the DNN to make a mistake by adding a small amount of noise into an original image; a human, however, cannot distinguish the difference between the original and the distorted images.

The basic method for generating adversarial examples is described in Section 2.1. The adversarial examples are classified in four different ways: target model information, distance measure, recognition of adversarial example, and method of generation, as described in Sections 2.2–2.5, which follow.

2.1. *Adversarial example generation*

The basic architecture for generating an adversarial example consists of two elements: the target model and the transformer. The transformer takes the original sample x and target class y as input data. The transformer then creates an output, a transformed example $x^* = x + w$ with noise value w added

to the original sample x. The transformed example x^* is supplied as input to the target model. The target model then provides the transformer with the class probability results for the transformed example. The transformer updates the noise values w in the transformed example $x^* = x + w$ so that the other class probabilities are higher than the original class probabilities while minimizing the distortion distances between x^* and x.

2.2. Categorization by target model information

Attacks that generate adversarial examples can also be divided into two different types according to the amount of information about the target is required for the attack: the white box attack (Carlini and Wagner, 2017b; Moosavi-Dezfooli et al., 2016; Szegedy et al., 2014) and the black box attack (Goodfellow et al., 2015; Papernot et al., 2017). The white box attack is used when the attacker has detailed information about the target model, i.e., model architecture, parameters, and probabilities for the output classes. Hence, the success rate of the white box attack reaches almost 100%.

A black box attack, on the other hand, is used when the attacker can query the target model without having the target model information. There are two well-known types of black box attack: the substitute network attack (Papernot et al., 2017) and the transferability attack (Goodfellow et al., 2015; Szegedy et al., 2014). The substitute model attack proposed in Papernot et al. (2017) is a well-known example of a black box attack. In this scheme, an attacker can create a substitute network similar to the target model by repeating the query process. Once a substitute network is created, the attacker can perform a white box attack.

The second, the transferability attack, is another example of a well-known black box attack. The work in Szegedy et al. (2014) and Goodfellow et al. (2015) introduces the concept of transferability, by which an adversarial example modified for a single local model is effective for other models that classify the same kind of data. In order to improve the transferability, the latest work (Strauss et al., 2017) has proposed an ensemble adversarial example method that uses multiple local models to attack the other models.

2.3. Categorization by distance measure

There are three ways to measure the distortion between the original sample and the adversarial example (Carlini and Wagner, 2017b; Meng and Chen, 2017). The first distance measure, L_0, represents the sum of the number of all changed pixels:

$$\sum_{i=0}^{n} |x_i - x_i^*|, \tag{1}$$

where x_i is the original ith pixel and x_i^* is the adversarial example's ith pixel. The second distance measure, L_2, represents the standard Euclidean norm, as follows:

$$\sum_{i=0}^{n} \sqrt{(x_i - x_i^*)^2}. \tag{2}$$

The third distance measure, L_∞, is the maximum distance value between x_i and x_i^*.

Therefore, as the three distance measures become smaller, the similarity of the example image to the original sample increases from a human perspective. However, there is no optimal distance measure, no perfect measure of human perceptual similarity. In this paper, L_2 is used as the distortion measure for MNIST and CIFAR10 (Section 5).

2.4. Categorization by target recognition on adversarial example

We can divide the adversarial examples (Carlini and Wagner, 2017b; Oliveira et al., 2016) into two subcategories according to the class recognized by the target model from the adversarial examples: a targeted adversarial example and an untargeted adversarial example. The first, the targeted adversarial example, is one that causes the target model to recognize the adversarial image as a particular intended class; it can be expressed mathematically as follows:

Given a target model and original sample $x \in X$, the problem can be reduced to an optimization problem that generates a targeted adversarial example x^*:

$$x^* : \underset{x^*}{\operatorname{argmin}}\, L(x, x^*) \ \text{ s. t. } \ f(x^*) = y^*, \tag{3}$$

where $L(\cdot)$ is a distance measure between original sample x and transformed example x^*, and y^* is the particular intended class. $\underset{x}{\operatorname{argmin}}\, F(x)$ is the x value at which the function $F(x)$ becomes minimal. "s. t." is an abbreviation for "such that." $f(\cdot)$ is an operation function that provides class results for the input values of the target model.

An untargeted adversarial example, on the other hand, is an adversarial example that causes the target model to recognize the adversarial image as any class other than the original class; it can be expressed mathematically as follows:

Given a target model and original sample $x \in X$, the problem can be reduced to an optimization problem that generates an untargeted adversarial example x^*:

$$x^* : \underset{x^*}{\operatorname{argmin}}\, L(x, x^*) \ \text{ s. t. } \ f(x^*) \neq y, \tag{4}$$

where $y \in Y$ is the original class.

The untargeted adversarial example has the advantage of less distortion in the original images and a shorter learning time compared to the targeted adversarial example. However, the targeted adversarial example is a more elaborate and powerful attack in that it can control the perception of the attacker's chosen class.

2.5. Categorization by adversarial example generation methods

There are four typical attacks that generate adversarial examples. The first method is the fast-gradient sign method (FGSM) (Goodfellow et al., 2015), which can find x^* through L_∞:

$$x^* = x + \epsilon \cdot \operatorname{sign}(\nabla loss_{F,t}(x)), \tag{5}$$

where F is an object function, and t is a target class. In every iteration of FGSM, the gradient is updated by ϵ from the original x, and x^* is found through optimization. This method is simple and has good performance. The second method is Iterative

FGSM (I-FGSM) (Kurakin et al., 2017a), which is an updated version of FGSM. Instead of changing the amount ϵ in every step, a smaller amount, α, is updated and eventually clipped by the same ϵ value:

$$x_i^* = x_{i-1}^* - \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla loss_{F,t}(x_{i-1}^*)). \quad (6)$$

I-FGSM provides better performance than FGSM. The third is the Deepfool method (Moosavi-Dezfooli et al., 2016), which is an untargeted attack and uses the L_2 distance measure. This method generates an adversarial example that is more efficient than FGSM and is as close as possible to the original image. To generate an adversarial example, the Deepfool method constructs a neural network and looks for x^* using the linearization approximation method. However, since the neural network is not completely linear, the adversarial example must be found through many iterations; i.e., the process is more complicated than FGSM. The fourth method is the Carlini attack (Carlini and Wagner, 2017b), which is the latest attack method and provides better performance than FGSM and I-FGSM. This method can achieve a 100% success rate even against the distillation structure (Papernot et al., 2016b), which was recently introduced in the literature. The key point of this method is to use a different objective function,

$$D(x, x^*) + c \cdot f(x^*), \quad (7)$$

instead of the conventional objective function $D(x, x^*)$, and it proposes how to find an appropriate binary c value. In addition, it suggests a method for controlling the attack success rate even if some distortion increases by reflecting the confidence value as follows:

$$f(x^*) = \max(Z(x^*)_t - \max\{Z(x^*)_t : i \neq t\}, -k), \quad (8)$$

where $Z(\cdot)$ represents the pre-softmax classification result vector and t is a target class.

In this paper, we construct the model by applying the Carlini attack, which is the most powerful of the four methods, and use L_2 with box constraint as the distortion loss function in the rescale range [0, 1], which changes the pixels of the grayscale image from full-on to full-off in the same manner as the Carlini method. This is because the Carlini method L_2 attack is superior to the L-BFGS attack (Szegedy et al., 2014) with a box constraint as it uses a better objective function.

2.6. *Related work on adversarial examples*

Some countermeasures have been proposed for the attack methods listed in Section 2.5. Biggio et al. (2014) proposed a binary classifier for detecting the adversarial example. This work covered conventional machine learning models (e.g., support vector machine) (Cortes and Vapnik, 1995) and logistic regression (Kleinbaum and Klein, 2010), but not a DNN. Goodfellow et al. (2015) proposed a new neural network activation function that is robust to adversarial examples. Applying this method necessitates changing the neural network's architecture.

Recently, Papernot et al. (2016b) proposed a defensive distillation scheme, in which an initial network and a distilled network are used. The class probability of the initial network's output is used as a label for training the distilled network. This prevents overfitting the distilled network and makes the network more robust to the adversarial example. In experiments, their scheme reduced the evasion attack success rate from 95.89% to 0.45%. However, one year later, Carlini and Wagner (2017b) showed that they could deceive a distilled network with a 100% success rate. They also showed that their scheme could be applied to both targeted and untargeted attacks.

In this manner, advanced attacks (Section 2.5) and (their countermeasures (this section) have been being proposed continuously. However, there has been no published scheme for building adversarial examples that do not affect friendly classifiers. In this paper, we use these state-of-the-art technologies in our friend-safe adversarial example scheme and its evaluation.

3. Problem definition

Szegedy et al. (2014) introduced the concept of transferability, by which adversarial examples targeting a single model gain the potential to attack other target models classifying the same kind of data.

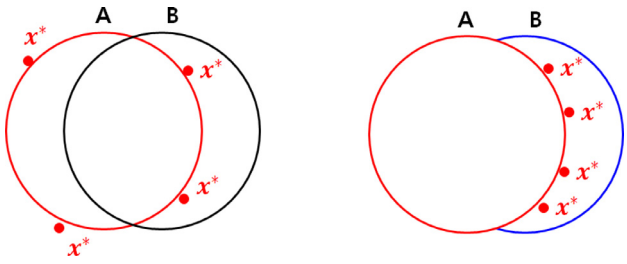
Fig. 1(a) shows an example of transferability with a single adversarial target, Model A. In Fig. 1(a), Models A and B are target models with a convolutional neural network (CNN). The circle is the decision boundary of the target model. If the image samples are within the circular boundary of the target model, the image samples are correctly classified by the target model into the original class. Alternatively, adversarial examples are generated along the circular boundary of a target model. This is because the adversarial example must be misclassified by the target model while minimizing its distance from the original sample. Each red dot in the figure is an adversarial example for target model A. Some of the adversarial examples are misclassified by target model B.

Fig. 1(b) shows an example of a friend-safe adversarial example that is correctly classified by target Model B and misclassified by target Model A. In Fig. 1(b), friend-safe adversarial examples, x^*, are within the decision boundary of target Model B but deviate from the decision boundary of target Model A. In terms of transferability, a friend-safe adversarial example is an adversarial example for target Model A that is not transferable to target Model B. Therefore, a new architecture is needed that uses both an attack target model and a protected model to generate friend-safe adversarial examples.

4. Proposed scheme

4.1. *Threat model*

The threat model of the proposed method is a neural network used in self-driving cars, drones, image classification, speech classification, and many other applications. We assume that the proposed method has white-box access to the friendly classifier and the enemy classifier and that it knows the model architecture, parameters, and probabilities of output classifications for the enemy classifier and for the friendly classifier.



(a) Adversarial example. (b) Friend-safe adversarial example.

Fig. 1 – Examples of transferability: a single adversarial target, Model A, and a friendly target, Model B. The circle is the decision boundary of the target model. The red dot is an adversarial example.

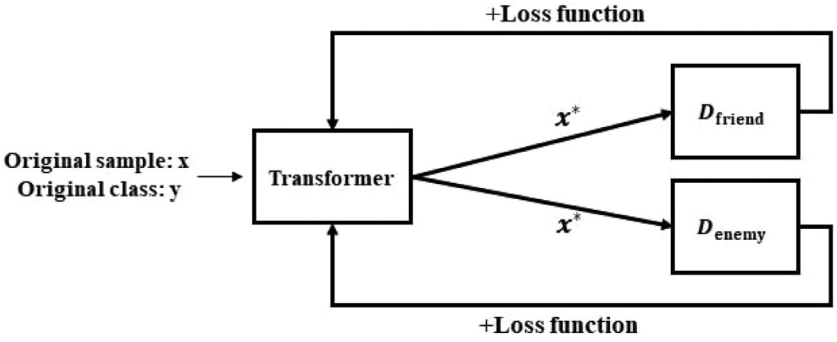


Fig. 2 – Proposed architecture.

This is a conservative and feasible assumption because it has been proven that a white box attack is possible for a black box model of the enemy classifier by constructing a substitute model. In this substitution scheme (Papernot et al., 2017), an attacker can create a substitute network that is similar to the enemy classifier by repeating the query process. Once a substitute network is created, the attacker can perform a white box attack using the substitute network. The attacker must have the ability to repeatedly query the friendly classifier and the enemy classifier. On the side of the threat models, the friendly classifier and the enemy classifier should provide feedback that includes the probabilities of the output classifications for the query. Using this feedback, the attacker can calculate the loss functions and thereby generate a friend-safe adversarial example.

4.2. *Proposed method*

To generate a friend-safe adversarial example, we propose a network architecture that consists of a transformer, a friendly discriminator D_{friend}, and an enemy discriminator D_{enemy}, as shown in Fig. 2. The transformer takes the original sample, $x \in X$, and the original class, $y \in Y$, as input and converts the original sample to the transformed example, x^*. D_{friend} and D_{enemy} are pre-trained classifiers and are not changed during transformation. They take x^* as input and provide their classification result (i.e., loss) to the transformer.

The goal of this architecture is for the transformed example, x^*, to be incorrectly classified by D_{enemy} and correctly classified by D_{friend} while minimizing the distortion from the original. There are two configurations in which the transformed example x^* is incorrectly classified by D_{enemy}: the targeted adversarial example and the untargeted adversarial example. In mathematical expressions, the operation functions of D_{enemy} and D_{friend} are denoted as $f_{\text{enemy}}(\cdot)$ and $f_{\text{friend}}(\cdot)$, respectively. Given the pre-trained D_{friend} and D_{enemy} and the original input $x \in X$, we have an optimization problem that generates the targeted adversarial example x^*:

$$x^* : \underset{x^*}{\operatorname{argmin}}\, L(x, x^*) \ \text{s.t.}\ f_{\text{friend}}(x^*) = y \ \text{and}\ f_{\text{enemy}}(x^*) = y^*, \quad (9)$$

where $L(\cdot)$ is the chosen measure of the distance between original sample x and transformed example x^*, and $y^* \in Y$ is the target class chosen by the attacker. An untargeted adversarial example x^* is generated similarly:

$$x^* : \underset{x^*}{\operatorname{argmin}}\, L(x, x^*) \ \text{s.t.}\ f_{\text{friend}}(x^*) = y \ \text{and}\ f_{\text{enemy}}(x^*) \neq y. \quad (10)$$

To achieve this goal, the procedure consists of pre-training D_{friend} and D_{enemy} and creating a transformation that generates a friend-safe adversarial example, x^*. First, D_{friend} and D_{enemy} are trained with the original sample to classify the original sample, x.

$$f_{\text{enemy}}(x) = y \in Y \ \text{and}\ f_{\text{friend}}(x) = y \in Y. \quad (11)$$

In our experiments, D_{friend} and D_{enemy} were trained to classify the original samples using MNIST and CIFAR10 with more than 99% accuracy and 91% accuracy, respectively. Second, the transformer accepts the original sample and original

class as input and produces the transformed example, x^*. For this study, we modified the transformer architecture given in Carlini and Wagner (2017b) and defined x^* as

$$x^* = \frac{\tanh(x+w)}{2}, \tag{12}$$

where w is a modifier used when optimizing with a gradient, and tanh is used to smooth the gradient as a box constraint (Carlini and Wagner, 2017b). The classification loss of x^* by D_{friend} and D_{enemy} are returned to the transformer. The transformer then calculates the total loss, $loss_T$, and generates a friend-safe adversarial example by minimizing $loss_T$ iteratively. $loss_T$ is defined as

$$loss_T = loss_{\text{distortion}} + loss_{\text{friend}} + loss_{\text{enemy}}, \tag{13}$$

where $loss_{\text{distortion}}$ is the distortion loss function, and $loss_{\text{friend}}$ and $loss_{\text{enemy}}$ are the classification loss functions of D_{friend} and D_{enemy}, respectively. $loss_{\text{distortion}}$ is the distortion loss function between the original sample x and the transformed example x^*:

$$loss_{\text{distortion}} = \sqrt{\left(x^* - \frac{\tanh(x)}{2}\right)^2}. \tag{14}$$

To satisfy $f_{\text{friend}}(x^*) = y$, $loss_{\text{friend}}$ should be minimized:

$$loss_{\text{friend}} = g^f(x^*), \tag{15}$$

where $g^f(k) = \max\{Z_f(k)_i : i \neq org\} - Z_f(k)_{org}$, and org is the original class. $Z_f(\cdot)$ and $Z_e(\cdot)$ (Carlini and Wagner, 2017b; Papernot et al., 2016a) are the probabilities of the classes being predicted by the two discriminators, D_{friend} and D_{enemy}, respectively. $f_{\text{friend}}(x^*)$ has a higher probability of predicting the original class than other classes by optimally minimizing $loss_{\text{friend}}$. $loss_{\text{enemy}}$ has two cases, those used in targeted and in untargeted adversarial examples. To satisfy $f_{\text{enemy}}(x^*) = y^*$, $y^* \in Y$, in targeted adversarial examples, $loss_{\text{enemy}}$ is defined as

$$loss_{\text{enemy}} = g^{e_t}(x^*), \tag{16}$$

where $g^{e_t}(k) = \max\{Z_e(k)_i : i \neq t\} - Z_e(k)_t$, and t is the targeted class. $f_{\text{enemy}}(x^*)$ has a higher probability of predicting the targeted class, y^*, than other classes by optimally minimizing $loss_{\text{enemy}}$. To satisfy $f_{\text{enemy}}(x^*) \neq y$ in an untargeted adversarial example,

$$loss_{\text{enemy}} = g^{e_u}(x^*), \tag{17}$$

where $g^{e_u}(k) = Z_e(k)_{org} - \max\{Z_e(k)_i : i \neq org\}$, and org is the original class. $f_{\text{enemy}}(x^*)$ has a lower probability of predicting the original class than other classes by optimally minimizing $loss_{\text{enemy}}$. The details of the procedure for generating a friend-safe adversarial example are given in Algorithm 1.

Algorithm 1 Friend-safe adversarial example generation in a transformer.

Input: original sample x, original class y, targeted class y^*, iterations r.

Targeted adversarial example generation:

$w \leftarrow 0$
$org \leftarrow y$
$t \leftarrow y^*$
$x^* \leftarrow 0$
for r step **do**
  $x^* \leftarrow \frac{\tanh(x+w)}{2}$
  $g^f(x^*) \leftarrow \max\{Z_f(x^*)_i : i \neq org\} - Z_f(x^*)_{org}$
  $g^{e_t}(x^*) \leftarrow \max\{Z_e(x^*)_i : i \neq t\} - Z_e(x^*)_t$
  $temp1 \leftarrow \sqrt{(x^* - \frac{\tanh(x)}{2})^2}$
  $temp2 \leftarrow temp1 + g^f(x^*) + g^{e_t}(x^*)$
  Update w by minimizing the gradient of $temp2$
end for
return x^*

Untargeted adversarial example generation:

$w \leftarrow 0$
$org \leftarrow y$
$x^* \leftarrow 0$
for r step **do**
  $x^* \leftarrow \frac{\tanh(x+w)}{2}$
  $g^f(x^*) \leftarrow \max\{Z_f(x^*)_i : i \neq org\} - Z_f(x^*)_{org}$
  $g^{e_u}(x^*) \leftarrow Z_e(x^*)_{org} - \max\{Z_e(x^*)_i : i \neq org\}$
  $temp1 \leftarrow \sqrt{(x^* - \frac{\tanh(x)}{2})^2}$
  $temp2 \leftarrow temp1 + g^f(x^*) + g^{e_u}(x^*)$
  Update w by minimizing the gradient of $temp2$
end for
return x^*

5. Experiment and evaluation

Through experiments, we show that the proposed scheme can generate a friend-safe adversarial example that is incorrectly classified by an enemy classifier and correctly classified by a friendly classifier, while minimizing the distortion of the original sample. We used the Tensorflow (Abadi et al., 2016) library, a widely used open source library for machine learning, on a Xeon E5-2609 1.7-GHz server.

5.1. Experimental method

In the experiment, we used MNIST (LeCun et al., 2010), a collection of handwritten digit images (0–9), and CIFAR10 (Krizhevsky et al., 2014), with 10 classes (plane, cars, birds, cats, deer, dogs, frogs, horses, boats, and trucks), as datasets. The MNIST dataset consists of 60,000 training data and 10,000 test data; the CIFAR10 dataset consists of 50,000 training data and 10,000 test data. The experimental method consisted of 1) pre-training D_{friend} and D_{enemy} and 2) transforming the friend-safe adversarial example.

First, during pre-training, D_{friend} and D_{enemy} are common CNNs (LeCun et al., 1998) in MNIST and VGG19-networks (Simonyan and Zisserman, 2015b) in CIFAR10. Their configuration and training parameters for MNIST and CIFAR10 are shown in Tables 13, 14, and 15 of the appendix. For MNIST, D_{enemy} was a distilled model (Papernot et al., 2016b), in which the classifier's output class probability is used as input to a second phase of classifier training. For MNIST, 60,000 training data were used to train D_{friend} and D_{enemy}. In the MNIST

Table 1 – A friend-safe adversarial example for each target class that was misclassified by D_{enemy} for each original sample.

Original	Targeted classes misclassified by D_{enemy}									
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
										
										
										
										

Table 2 – A friend-safe adversarial example and distortion of an original "7" for each target class, from Table 1

Original	Targeted classes misclassified by D_{enemy}									
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Rate	2.3	2.7	1.0	1.5	2.4	2.7	3.0	-	1.9	2.0

test, D_{friend} and D_{enemy} correctly classified the original MNIST samples with 99.25% and 99.12% accuracy, respectively. For CIFAR10, 50,000 training data were used to train D_{friend} and D_{enemy}. In the CIFAR10 test, D_{friend} and D_{enemy} correctly classified the original CIFAR10 samples with 91.24% and 91.13% accuracy, respectively.

Second, to generate the friend-safe adversarial example, Adam (Kingma and Ba, 2015) was used as an optimizer to minimize the total loss with a learning rate of 1×10^{-2} and an initial constant equal to 1×10^{-3}. For a given number of iterations, the transformer updates the output, x^*, and gives it to D_{friend} and D_{enemy}, from which it then receives feedback. At the end of the iterations, the transformation result, x^*, was evaluated in terms of the accuracy of D_{friend}, the attack success rate, and the amount of distortion. The accuracy of D_{friend} is the coincidence rate between the original class and the output class of D_{friend}. The attack success rate is the rate at which D_{enemy} incorrectly classifies x^*. The attack success rate has two configurations: the targeted attack success rate and the untargeted attack success rate. The targeted attack success rate is the coincidence rate between the targeted class and the class output by D_{enemy}; the untargeted attack success rate is the rate of inconsistency between the original class and the output class of D_{enemy}. In our experiments, the definition of distortion used was L_2, which is the sum of the square root of each pixel difference from the original sample, as in the Euclidean norm.

5.2. *Experimental results on MNIST*

For MNIST, the evaluation of friend-safe adversarial examples, x^*, is divided into two sections, targeted and untargeted adversarial examples.

5.2.1. *Targeted adversarial example*

Table 1 shows, for each original sample, friend-safe adversarial examples x^* generated by a transformer that were incorrectly classified as the targeted class by D_{enemy}. The number of iterations was 1,000 and the average distortion was 2.03. By human perception, the friend-safe adversarial examples x^* in Table 1 are similar to the original samples.

Table 3 – Class score of friend-safe adversarial example "7" → "0," from Table 2

Description	Original ("7")	Friend-safe adversarial example	
		D_{enemy}("0″)	D_{friend}("7″)
Image			
Class score	[0 0 0 0 0 0 0 **1** 0 0]	[**694** -225 692 -262 -319 -533 -376 **693** -142 -30]	[-1.4 1.2 7.1 -0.4 0.4 -7.4 -6.5 **10.5** -4.2 -3.8]

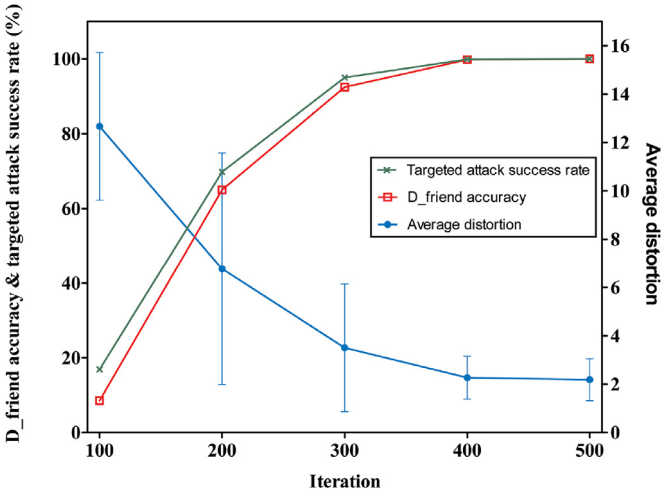


Fig. 3 – Targeted attack success rate, D_{friend} accuracy, and average distortion of 1000 friend-safe adversarial examples for each number of iterations.

Table 2. shows, for each targeted class, the average distortion of the original sample "7" corresponding to the generated adversarial examples exemplified in Table 1. The average distortion of this sample differs for each targeted class. For example, targeting class "6" results in the maximum distortion of the "7," whereas targeting class "2" produces the minimum distortion. The total average distortion of the original "7" sample is approximately 2.18. Fig. 6 in the appendix shows the average distortion of each targeted class for each original sample, found by analyzing 1,000 random friend-safe adversarial examples, which can be used in some situations for selecting targeted classes.

Table 3 shows the targeted transformation example "7" → "0," whose classification is determined by the class score. For D_{enemy}, the score of the target class "0," 694, is slightly higher than that of the original class, 693. For D_{friend}, the score of the original class, "7," is much higher than the scores of the other classes. From this result, we know that the transformation is minimized to the extent that the target class score is only slightly higher than the score of the original class, while maintaining low distortion rates.

Fig. 3 shows the targeted attack success rate, D_{friend} accuracy, and average distortion of 1000 friend-safe adversarial examples, with standard deviations indicated by the vertical bars. As the number of iterations increases, the targeted attack success rate and the D_{friend} accuracy increase, and the average distortion decreases. When the iteration count exceeds 500, the D_{friend} accuracy and the targeted attack success rate both reach 100%. At this point, the average distortion is less than 2.183. The attack success rate increases more quickly than the friendly classifier's accuracy, meaning that it requires more time to generate examples that will be correctly classified by a friendly classifier.

Table 4 – Images of the friend-safe adversarial example for the iteration counts shown in Fig. 3.

Iteration	100	200	300	400	500
Image					

Table 4 shows the iterative process of generating an example image. We think that because the image is generated from a zero matrix (black background), it is easier to make an enemy result incorrect than to make a friendly result correct.

5.2.2. *Untargeted adversarial example*

Table 5 shows the confusion matrix of the untargeted adversarial example classified by D_{enemy}, testing 100 untargeted adversarial examples per original sample. When a target class is not given, transformation mainly affects a few specific classes. Transformation is made to any class other than the original class, so minimal modification is required.

Fig. 4 shows the untargeted attack success rate, D_{friend} accuracy, and average distortion, with standard deviations

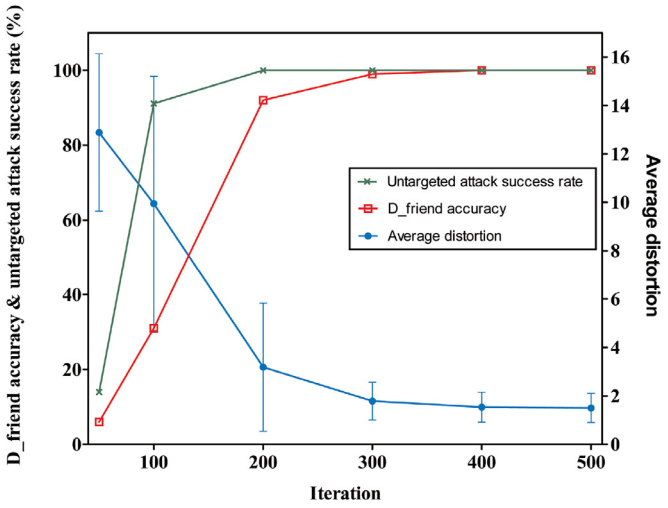


Fig. 4 – Untargeted attack success rate, D_{friend} accuracy, and average distortion of 1,000 friend-safe adversarial examples for each number of iterations.

Table 5 – Confusion matrix of D_{enemy} for untargeted class (400 iterations).

Original	Output class									
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
0	0	0	**11**	2	2	7	**24**	**15**	4	**35**
1	0	0	1	1	**46**	1	0	**11**	**40**	0
2	6	**26**	0	**29**	1	0	2	**25**	**11**	0
3	0	4	**14**	0	1	**57**	0	**19**	5	1
4	1	**13**	7	0	0	1	6	7	7	**58**
5	0	0	0	**38**	0	0	6	0	**18**	**38**
6	**16**	1	1	0	**13**	**63**	0	0	6	0
7	0	**18**	9	**21**	4	0	0	0	1	**47**
8	7	2	**13**	**42**	2	**18**	3	3	0	10
9	0	0	0	7	**37**	2	0	**30**	**24**	0

indicated by the vertical bars. As in Fig. 4, as the number of iterations increases, the untargeted attack success rate and the D_{friend} accuracy increase, and the average distortion decreases. When the iteration count exceeds 400, the D_{friend} accuracy and the untargeted attack success rate both reach 100%. At this point, the average distortion is less than 1.536. The attack success rate saturates much faster than the friendly classifier's accuracy; this difference is larger than in the targeted attack case. Hence, the lack of target restrictions allows faster successful attacks.

Table 6 shows the iteration count and distortion that are required to achieve 100% accuracy in each case of 1,000 friend-safe adversarial examples. The untargeted examples reach 100% faster than the targeted examples, and distortion in the untargeted case is also smaller than in the targeted case. In both cases, the attack success rate reaches 100% faster than the friend's accuracy. We discuss the implications of this result in Section 7.

5.3. *Experimental results on CIFAR10*

For CIFAR10, the evaluation of friend-safe adversarial examples, x^*, is divided into two sections, targeted and untargeted adversarial examples.

5.3.1. *Targeted adversarial example*

Table 7 shows, for each original sample, friend-safe adversarial examples x^* generated by a transformer that were misclassified as the targeted class by D_{enemy}. The number of iterations was 10,000, and the average distortion was 48.7. By human perception, the friend-safe adversarial examples x^* for CIFAR10 are more similar to their original samples than are those for MNIST.

Table 6 – Comparison of targeted and untargeted attacks when the attack success rate is 100% and friend accuracy is 100%. "SD" is standard deviation.

Description	Attack success rate		D_{friend} accuracy	
	Targeted	Untargeted	Targeted	Untargeted
Iterations	500	300	500	400
Maximum distortion	6.645	4.016	6.645	3.440
Minimum distortion	0.232	0.249	0.232	0.234
SD distortion	0.878	0.776	0.878	0.621
Mean distortion	2.183	1.788	2.183	1.536

Table 7 – A friend-safe adversarial example for each target class that was misclassified by D_{enemy} for each original sample: plane "0," cars "1," birds "2," cats "3," deer "4," dogs "5," frogs "6," horses "7," boats "8," and trucks "9."

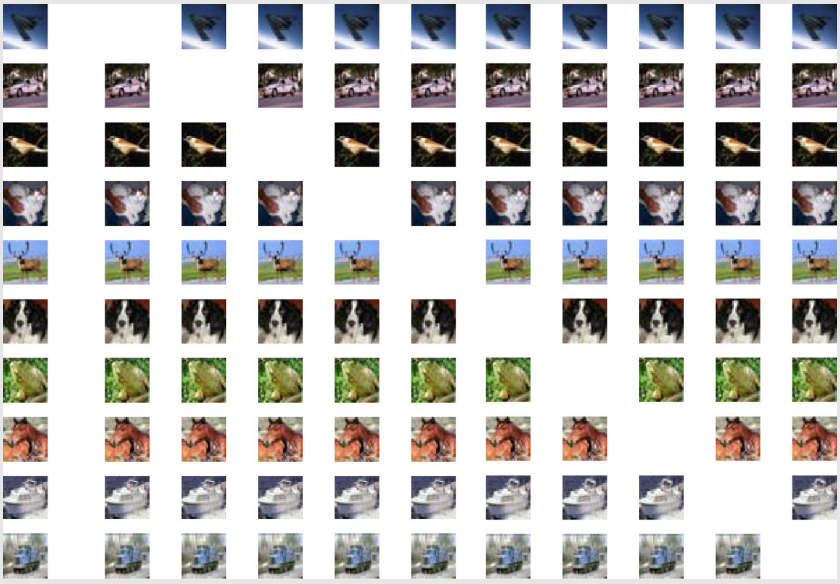
Original	Targeted classes misclassified by D_{enemy}									
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
										

Table 8 – A friend-safe adversarial example and distortion of an original sample, "horses," for each target class, from Table 7.

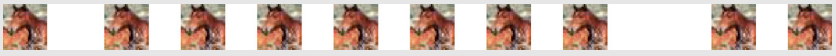
Original	Targeted classes misclassified by D_{enemy}									
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
										
Rate	42	33	8.9	25	0.1	26	18	-	47	34

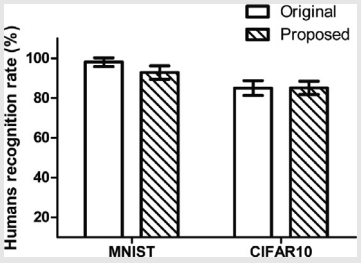
Table 8 shows, for each targeted class, the average distortion of the original sample "horses" corresponding to the generated adversarial examples exemplified in Table 7. The average distortion of this sample differs for each targeted class. For example, targeting class "8" results in the maximum distortion of the "horses" sample, whereas targeting class "4" produces the minimum distortion. The total average distortion of the original "horses" sample is approximately 26.31. The average distortion for CIFAR10 is higher that that for MNIST.

However, although the average distortion of CIFAR10 is higher, the human recognition rate for CIFAR10 is not lower than that for MNIST. Table 9 shows the human recognition rate by forty humans of 100 friend-safe adversarial examples and 100 original samples in MNIST and CIFAR10. Forty students and researchers from Kongju National University and Korea Advanced Institute of Science and Technology were tested to ascertain the human recognition rate. Their average age was 24 years, the maximum was 37 years, and the minimum was 21 years. Their average eyesight was 0.9, and the standard deviation was 0.23. The sex ratio was 29 males to 11 females. For CIFAR10, the human recognition rate for the original samples is lower than that for MNIST because humans are confused when distinguishing dogs and cats, cars and trucks, deer and horses. However, when the differences in human recognition between friend-safe adversarial examples and the original samples are compared, the performance of CIFAR10 is higher than that of MNIST because a CIFAR10 sample is a 3D color image, whereas an MNIST sample is a 1D monochrome image. Therefore, the human recognition rate should be considered not only in terms of distortion but also by the dimensionality of the data.

5.3.2. *Untargeted adversarial example*

Table 10 shows the confusion matrix of the untargeted adversarial example classified by D_{enemy}, testing 100 untargeted adversarial examples per original sample. Similar to MNIST,

Table 9 – Human recognition rates for 100 friend-safe adversarial examples and 100 original samples by forty humans with the MNIST and CIFAR10 datasets. "Proposed" is a friend-safe adversarial example; "original" is the original sample. "SD" is standard deviation.



Description	MNIST		CIFAR10	
	Original	Proposed	Original	Proposed
Mean	98.1	92.9	85.1	85.19
SD	2.21	3.37	3.717	3.66
Maximum	100	98	92	92
Minimum	88	82	80	79

Table 10 – Confusion matrix of D_{enemy} for untargeted class (6000 iterations).

Original	Output class									
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"
0	0	6	**37**	7	**14**	1	1	1	**27**	6
1	4	0	0	0	1	0	2	0	**19**	**74**
2	**18**	1	0	8	**32**	**17**	**16**	8	0	0
3	4	0	**13**	0	**15**	**44**	**13**	8	2	1
4	2	1	**37**	**15**	0	**11**	9	**23**	2	0
5	0	0	9	**61**	7	0	4	**19**	0	0
6	0	0	**18**	**67**	**14**	0	0	0	0	1
7	4	0	3	**9**	**72**	9	0	0	2	1
8	**65**	9	2	9	3	1	1	0	0	10
9	3	**63**	0	4	1	1	1	2	**25**	0

when no target class is provided, the transformation focused on some specific classes. This is because the transformation for these specific classes requires only minor modifications compared to the other classes.

Table 11 shows the iteration count and distortion that are required to achieve 100% accuracy in each case of 1,000 friend-safe adversarial examples for CIFAR10. As with MNIST, the untargeted adversarial examples reaches 100% faster than the targeted adversarial examples, and distortion in the untargeted case is smaller than in the targeted case. However, to generate a friend-safe adversarial example, CIFAR10 requires more iterations and distortion than does MNIST because of the higher dimensionality of the CIFAR10 images.

6. New covert channel scheme

As mentioned in the introduction, a friend-safe adversarial example is used as an evasion attack against an enemy in a mixed battlefield. In addition to this application, we discovered an interesting covert channel scheme, shown in Fig. 5. In this scheme, the roles of friend and enemy are reversed. The sender generates an example that is correctly recognized by a machine or human censor (i.e., enemy) and incorrectly classified by the receiver (i.e., friend). The target class is the hidden information that is transferred via the covert channel.

To evaluate the performance of the covert channel, we randomly generated 100 friend-safe adversarial examples from the MNIST and CIFAR10 datasets to test the hidden images on forty people. The results are shown in Table 12. In the case of MNIST, we needed to know which number among the remaining nine was hidden in the displayed numerical image. Similarly, for the case of CIFAR10, it was a matter of ascertaining which of the nine objects (those other than the visible object) was hidden. Although for the MNIST case the detection rate was 1.4 percentage points higher than that by random selection (11.1%), in both cases the probability of choosing one of the nine classes was close to 11.1%. The experimental results show that a new covert channel, using a friend-safe adversarial example, has the probability of fooling humans that is close to random chance.

7. Discussion

Usability of the proposed method We have shown that it is possible to generate an adversarial example that simultaneously achieves a 100% attack success rate and a 100% accuracy rate by a friendly classifier. This is possible because the enemy and friendly classifiers are different. It is impossible to generate such examples if the two are identical.

In the experiments with MNIST in Section 5, the enemy uses a distilled classifier and the friend employs a general CNN classifier. To study the possibility of generating adversarial examples with two very similar models, we tested the same classifier configuration for both the friend and enemy and provided the same training data with a different sample order. With this setup, we found the same results: a friend-safe adversarial example with a 100% attack success rate and 100% accuracy by friendly classifiers (see Figs. 7 and8 in the appendix).

Attack considerations From Table 6, we found that untargeted attacks require less distortion and are ideal when targeting is unnecessary or when minimizing distortion is important. During untargeted attacks, the attacker should estimate the probability of the to-be-recognized class from Table 5. For example, when the original class is "9," "4," "7," or "8," we have high probabilities of enemy recognition. This is useful for cases in which a specific target is not necessary and minimizing distortion is important. However, when the attacker wants to know the victim's (i.e., enemy's) classification result, the attacker can refer to Fig. 6 of the appendix and select a target class having low distortion. For example, if an attacker wants to cause the victim to recognize a road sign with the digit "9" as something other than "9," he could select "7" as the target class because "9" → "7" requires the least distortion. In this case, he knows that the victim will recognize the road sign as "7," the target class that the attacker has selected.

Table 11 – Comparison of targeted and untargeted attacks when the attack success rate is 100% and friend accuracy is 100%. "SD" is standard deviation.

Description	Attack success rate		D_{friend} accuracy	
	Targeted	Untargeted	Targeted	Untargeted
Iterations	10,000	6,000	10,000	6,000
Maximum distortion	147.32	116.5	147.32	116.5
Minimum distortion	0.002	0.002	0.002	0.002
SD distortion	26.271	26.580	26.271	26.580
Mean distortion	49.017	27.605	49.017	27.605

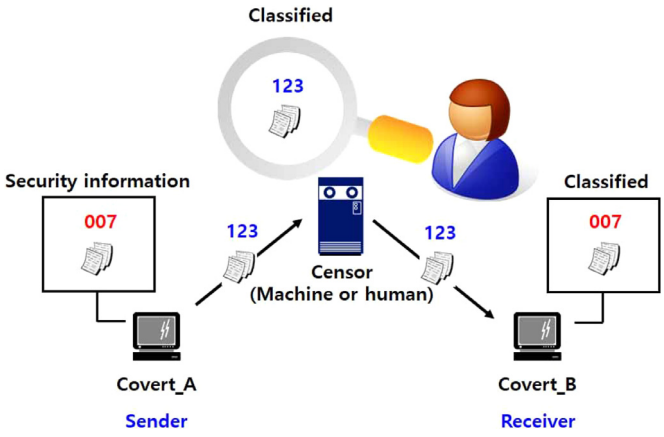
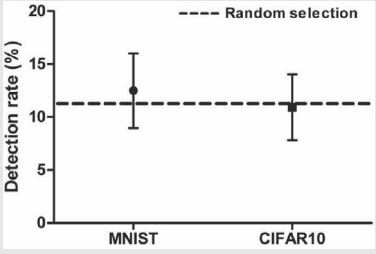


Fig. 5 – New covert channel scheme using a friend-safe adversarial example.

Table 12 – Detection rates of 100 friend-safe adversarial examples by forty humans in a covert channel. "SD" is standard deviation.



Description	MNIST	CIFAR10
Mean	12.5	10.925
SD	3.53	3.12
Maximum	19	16
Minimum	3	2

Transferability to an unknown classifier In terms of transferability to an unknown classifier, the friend-safe adversarial example has the same transferability as a conventional method that generates the adversarial example targeting one model. To verify the transferability of the friend-safe adversarial example, we tested 1000 randomized friend-safe adversarial examples of MNIST and CIFAR10 images for an unknown classifier. In the test, the friend-safe adversarial example showed the same transferability as the conventional method: 5.5% with MNIST and 27.3% with CIFAR10.

Datasets We evaluated the performance of the proposed method using the MNIST and CIFAR10 datasets. The experiment results show that the number of iterations required, the average distortion, and human recognition depend on the dataset. In terms of the number of iterations required, MNIST requires fewer iterations and generates less distortion for creating friend-safe adversarial examples than does CIFAR10. Because an MNIST sample is a 1D monochromatic image, transformers need a shorter generation process to produce an adversarial example. In terms of the average distortion, although the distortion on CIFAR10 was higher than that on MNIST, human recognition with CIFAR10 was more similar to that of the original sample than that with MNIST. This result shows that with an adversarial example generated for a 3D image such as those in CIFAR10, no problem can be detected by eye. Therefore, the characteristics of the dataset affect the distortion and the human recognition. Even though the distortion depends on the dataset, human recognition is maintained owing to the minimal amount of distortion on each dataset, as in the image domain (Carlini and Wagner, 2017b) and audio domain (Carlini and Wagner, 2018).

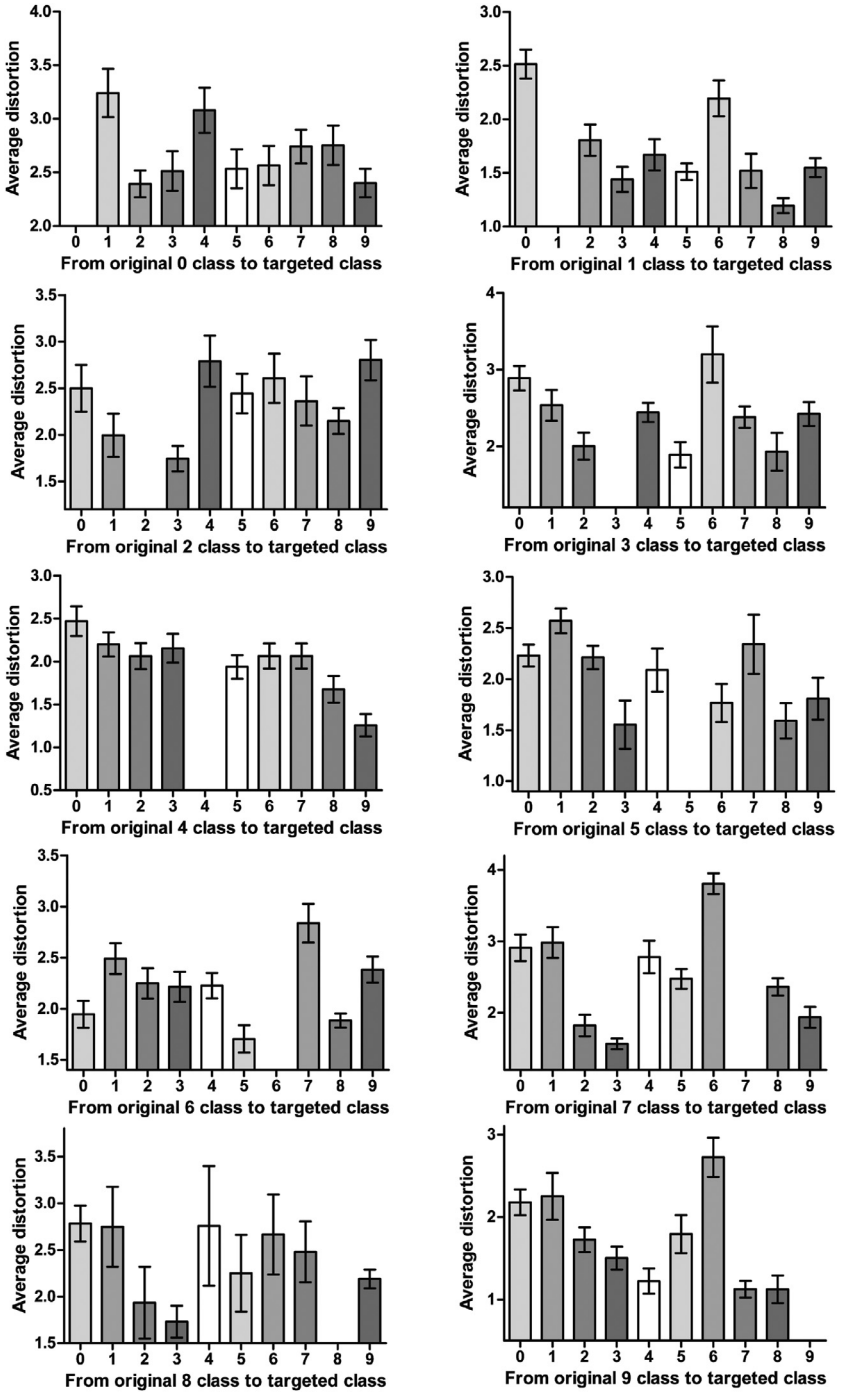


Fig. 6 – Average distortion for the targeted class for each of the original classes 0–9 in 1,000 friend-safe adversarial examples (380 iterations).

Distortion The distortion measure is L_2, the sum of the square root of each pixel difference, so there is a high probability that the distortion will increase as the size (pixels) or the dimensionality of the image increases. For example, an MNIST sample is a 1D image that has a total of 784 pixels as a matrix ($28 \times 28 \times 1$). CIFAR10 is a 3D image that has a total of 3072 pixels as a matrix ($32 \times 32 \times 3$). The experimental results in Section 5 show that the average distortion with CIFAR10 is greater than the average distortion with MNIST.

We also found that the image distortion rate is more related to the similarity between the original image and the target image than the original image itself. Table 17 in the appendix shows friend-safe adversarial examples for MNIST and CIFAR10 with maximum and minimum distortion out of 1000 friend-safe adversarial examples. In the untargeted attacks, there were similarities between truck and car, bird and cat, 3 and 5, and 9 and 5, which served to minimize the distortion. In the maximum distortions for the targeted attacks shown

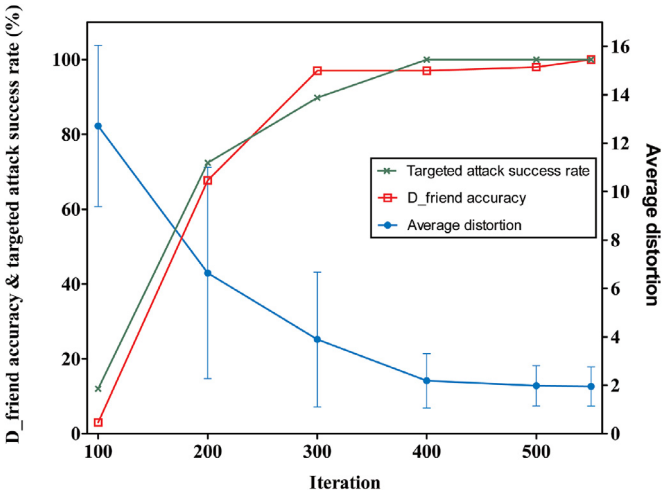


Fig. 7 – Targeted attack success rate, D_{friend} accuracy, and average distortion of friend-safe adversarial examples for each number of iterations in both models.

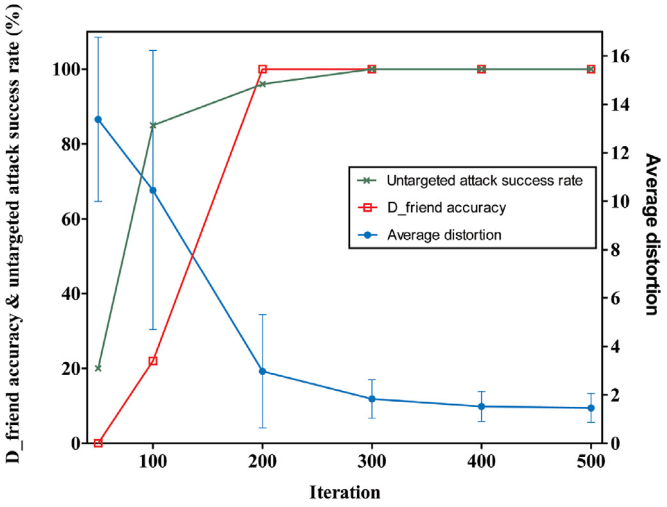


Fig. 8 – Untargeted attack success rate, D_{friend} accuracy, and average distortion of friend-safe adversarial examples for each number of iterations in both models.

in this table, the similarities between car and deer and between 1 and 0 are relatively low. This result shows that the image distortion is increased when the similarity between the original image and the target image is low. Table 18 in the appendix shows that images with maximum and minimum distortion from Table 17 are incorrectly classified as different target classes. The results of Table 18 show that a high degree of similarity between the original image and the target image can reduce distortion.

Type of models For MNIST, D_{friend} used the general CNN classifier, and D_{enemy} used the distilled classifier. A heterogeneous architecture with a different model configuration was used. In the case of CIFAR10, D_{friend} and D_{enemy} were the same general CNN classifier but used a homogeneous architecture constructed by different datasets. This demonstrates that both heterogeneous and homogeneous architectures can generate friend-safe adversarial examples.

Accuracy of models Fig. 9 in the appendix shows the average distortion (with standard deviations as indicated by the vertical bars) for each level of accuracy of D_{friend} and D_{enemy} when the proposed method generates a friend-safe adversarial example that has 100% attack success with CIFAR10. As shown in the figure, the proposed method achieves similar performance at each level of model accuracy. This is because there is a trade-off between the enemy classifier and the friendly classifier. Low-accuracy models require less distortion for the misclassification, but more distortion is needed for the friendly classifier to classify correctly. On the other hand, high-accuracy models require more distortion for the misclassification, but less distortion is required for the friendly classifier to classify correctly.

Applications If we apply the friend-safe adversarial example to other applications, we can use it for signs as well as for the newly proposed covert channel scheme. For example, a sign

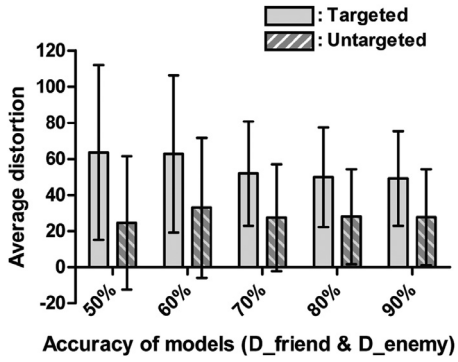


Fig. 9 – Average distortion of the friend-safe adversarial examples with CIFAR10 for each level of accuracy of D_{friend} and D_{enemy} when the attack success rate is 100% and friend accuracy is 100%.

Table 13 – D_{friend} and D_{enemy} model architecture for MNIST.

Layer type	MNIST shape
Convolution+ReLU	[3, 3, 32]
Convolution+ReLU	[3, 3, 32]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Fully connected+ReLU	[200]
Fully connected+ReLU	[200]
Softmax	[10]

Table 14 – D_{friend} and D_{enemy} model parameters.

Parameter	MNIST model	CIFAR10 model
Learning rate	0.1	0.1
Momentum	0.9	0.9
Delay rate	–	10 (decay 0.0001)
Dropout	0.5	0.5
Batch size	128	128
Epochs	50	200

that uses a friend-safe adversarial example could deceive an enemy vehicle and at the same time not deceive a friendly vehicle.

8. Conclusion

In this paper, we have proposed a method for generating a friend-safe adversarial example that will be incorrectly classified by D_{enemy} and correctly classified by D_{friend}, while minimizing distortion of the original sample. In the experimental results on MNIST data and CIFAR10 data, D_{friend} correctly classified transformed examples as the original class with 100% accuracy, and the attack success rate was 100% in both targeted and untargeted attacks when the data distortion was 2.183 and 1.536 for MNIST, and 49.017 and 27.605 for CIFAR10,

Table 15 – D_{friend} and D_{enemy} model architecture Simonyan and Zisserman (2015b) for CIFAR10.

Layer type	CIFAR10 shape
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 128]
Convolution+ReLU	[3, 3, 128]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Fully connected+ReLU	[4096]
Fully connected+ReLU	[4096]
Softmax	[10]

respectively. With regard to human recognition, the rate of human recognition of the friend-safe adversarial example was 95.9% for MNIST and 100% for CIFAR10. We discovered that distortion differs between target digit classes. This information is useful for selecting a targeted class. We also presented two applications of the proposed scheme: a mixed battlefield and a covert channel. Through experiments with the new covert channel scheme, the friend-safe adversarial example was able to fool 100% of the machines and to fool humans with a probability near that by random selection.

Future research will extend our experiments to other standard image datasets, such as ImageNet (Deng et al., 2009). Additionally, studies on evasion attacks in the voice field have been conducted by extending research in the voice and image fields (Carlini et al., 2016; Zhang et al., 2017). Thus, friend-safe adversarial examples are also applicable to evasion attacks in voice-related fields. Along with these, other applications using the friend-safe adversarial example can be applied to research. We will also work on generating a friend-safe adversarial example not through transformation but by applying a generative scheme, such as with a generative adversarial network (Goodfellow et al., 2014; Odena et al., 2017). Finally, another challenge will be to develop a countermeasure to the proposed scheme.

Acknowledgments

We thank the editors and the anonymous reviewers, who gave us very helpful comments that improved this paper. This work was supported by National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT)

Table 16 – Untargeted class of friend-safe adversarial example in D_{enemy}.

Description	Original ("1")	Friend-safe adversarial example	
		D_{enemy}("8")	D_{friend}("1")
Image			
Class score	[0 **1** 0 0 0 0 0 0 0 0]	[−441 1161 −388 −37 −93 −186 −70 −459 **1164** −245]	[−6.9 **23** −2.6 −5.2 −0.8 −3.3 −3.3 −4.3 1.1 −6.5]

Table 17 – Friend-safe adversarial examples with maximum and minimum distortions for MNIST and CIFAR10, from among 1000 friend-safe adversarial examples. "Max" is maximum; "Min" is minimum.

Description	MNIST				CIFAR10			
	Targeted		Untargeted		Targeted		Untargeted	
	Max	Min	Max	Min	Max	Min	Max	Min
Original								
Friend-safe								
Distortion	6.64	0.232	3.26	0.234	147.3	0.002	116.5	0.002
Wrong class	"0"	"5"	"5"	"5"	"deer"	"cat"	"car"	"cat"

Table 18 – Targeted friend-safe adversarial examples with maximum and minimum distortion from Table 17 are incorrectly classified as different target classes.

MNIST	Maximum	Target class	"0"	"7"	"9"
		Friend-safe			
		Distortion	6.64	1.45	1.52
	Minimum	Target class	"5"	"2"	"3"
		Friend-safe			
		Distortion	0.232	2.67	2.08
CIFAR10	Maximum	Target class	"deer"	"ship "	"truck"
		Friend-safe			
		Distortion	147.3	57.07	98.2
	Minimum	Target class	"cat"	"plane"	"car"
		Friend-safe			
		Distortion	0.002	35.1	38.7

(2016R1A4A1011761 and 2017R1A2B4006026) and an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2016-0-00173, Security Technologies for Financial Fraud Prevention on Fintech).

Appendix

Tables 13–18.

REFERENCES

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, et al. Tensorflow: a system for large-scale machine learning, 16; 2016. p. 265–83.

Barreno M, Nelson B, Joseph AD, Tygar J. The security of machine learning. Mach Learn 2010;81(2):121–48.

Biggio B, Fumera G, Roli F. Security evaluation of pattern classifiers under attack. IEEE Trans Knowl Data Eng 2014;26(4):984–96.

Biggio B, Nelson B, Laskov P. Poisoning attacks against support vector machines. In: Proceedings of the 29th international conference on international conference on machine learning. Omnipress; 2012. p. 1467–74.

Carlini N, Mishra P, Vaidya T, Zhang Y, Sherr M, Shields C, Wagner D, Zhou W. Hidden voice commands. In: Proceedings of the USENIX security symposium; 2016. p. 513–30.

Carlini N, Wagner D. Adversarial examples are not easily detected: Bypassing ten detection methods. Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM; 2017a.

Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proceedings of the IEEE symposium on security and privacy (SP). IEEE; 2017b. p. 39–57.

Carlini N, Wagner D. Audio adversarial examples: Targeted attacks on speech-to-text. Deep Learning and Security Workshop 2018.

Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference on machine learning. ACM; 2008. p. 160–7.

Cortes C, Vapnik V. Support vector machine. Mach Learn 1995;20(3):273–97.

Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2009. CVPR 2009. IEEE; 2009. p. 248–55.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Proceedings of the advances in neural information processing systems; 2014. p. 2672–80.

Goodfellow I, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. International conference on learning representations, 2015.

Hinton G, Deng L, Yu D, Dahl GE, Mohamed A-r, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Process Mag 2012;29(6):82–97.

Kingma D, Ba J. In: Proceedings of the international conference on learning representations (ICLR). Adam: a method for stochastic optimization; 2015.

Kleinbaum DG, Klein M. Introduction to logistic regression. In: Logistic regression. Springer; 2010. p. 1–39.

Krizhevsky, A., Nair, V., Hinton, G., 2014. The cifar-10 dataset. online: http://www.cs.toronto.edu/kriz/cifar.html.

Kurakin A, Goodfellow I, Bengio S. In: Proceedings of the ICLR workshop. Adversarial examples in the physical world; 2017a.

Kurakin A, Goodfellow IJ, Bengio S. Adversarial machine learning at scale. Proceedings of the international conference on learning representations (ICLR), 2017b.

Kwon H, Yoon H, Choi D. Friend-Safe Adversarial Examples in an Evasion Attack on a Deep Neural Network. International Conference on Information Security and Cryptology. Springer, Cham, 2017.

LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.

LeCun Y, Cortes C, Burges CJ. Mnist handwritten digit database 2010;2. AT&T Labs Available http://yann.lecun.com/exdb/mnist.

McDaniel P, Papernot N, Celik ZB. Machine learning in adversarial settings. IEEE Secur Privacy 2016;14(3):68–72.

Meng D, Chen H. Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. ACM; 2017. p. 135–47.

Moosavi-Dezfooli S-M, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2574–82.

Mozaffari-Kermani M, Sur-Kolay S, Raghunathan A, Jha NK. Systematic poisoning attacks on and defenses for machine learning in healthcare. IEEE J Biomed Health Inform 2015;19(6):1893–905.

Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans. Proceedings of the ICML, 2017.

Oliveira GL, Valada A, Bollen C, Burgard W, Brox T. Deep learning for human part discovery in images. In: Proceedings of the IEEE international conference on robotics and automation (ICRA). IEEE; 2016. p. 1634–41.

Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A. Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia conference on computer and communications security. ACM; 2017. p. 506–19.

Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proceedings of the 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE; 2016a. p. 372–87.

Papernot N, McDaniel P, Wu X, Jha S, Swami A. Distillation as a defense to adversarial perturbations against deep neural networks. In: Proceedings of the IEEE symposium on security and Privacy (SP). IEEE; 2016b. p. 582–97.

Potluri S, Diedrich C. Accelerated deep neural networks for enhanced intrusion detection system. In: Proceedings of the IEEE 21st international conference on emerging technologies and factory automation (ETFA). IEEE; 2016. p. 1–8.

Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw 2015;61:85–117.

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al. Mastering the game of go with deep neural networks and tree search. Nature 2016;529(7587):484–9.

Simonyan K, Zisserman A. In: Proceedings of the international conference on learning representations. Very deep convolutional networks for large-scale image recognition; 2015.

Simonyan K, Zisserman A. In: Proceedings of the ICLR 2015. Very deep convolutional networks for large-scale image recognition; 2015.

Smeets M, Koot M. Covert channels. RPI University of Amsterdam MSc in System and Network Engineering; 2006.

Strauss T, et al. Ensemble methods as a defense to adversarial perturbations against deep neural networks. arXiv:1709.03423 2017.

Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. Proceedings of the international conference on learning representations, 2014.

Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: attacks and defenses. Proceedings of the international conference on learning representations (ICLR), 2018.

Yang C, et al. Generative poisoning attack method against neural networks. arXiv:1703.01340 2017.

Zhang G, Yan C, Ji X, Zhang T, Zhang T, Xu W. Dolphinattack: inaudible voice commands. In: Proceedings of the ACM SIGSAC conference on computer and communications security. ACM; 2017. p. 103–17.

Hyun Kwon received the B.S degree in mathematics from Korea Military Academy, South Korea, in 2010. He also received the M.S. degree in School of Computing from Korea Advanced Institute of Science and Technology (KAIST) in 2015. He is currently working toward the Ph.D. degree at School of Computing, KAIST. His research interests include information security, computer security, and intrusion tolerant system.

Yongchul Kim received the B.E. degree in electrical engineering from the Korea Military Academy, South Korea, in 1998, the M.S. degree in electrical engineering from the University of Surrey, U.K., in 2001, and the Ph.D. degree in electrical and computer engineering with the department of electrical and computer engineering, North Carolina State University, in 2007. He has been a professor in the department of electrical engineering at the Korea Military Academy. His research interests include WiMAX and wireless relay networks.

Ki-Woong Park received the B.S. degree in computer science from Yonsei University, South Korea, in 2005, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST) in 2007, and the Ph.D. degree in electrical engineering from KAIST in 2012. He received a 2009–2010 Microsoft Graduate Research Fellowship. He worked for National Security Research Institute as a senior researcher. He has been a professor in the department of computer and information security at Sejong University. His research interests include security issues for cloud and mobile computing systems as well as the actual system implementation and subsequent evaluation in a real computing system.

Hyunsoo Yoon received the B.E. degree in electronics engineering from Seoul National University, South Korea, in 1979, the M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST) in 1981, and the Ph.D. degree in computer and information science from the Ohio State University, Columbus, Ohio, in 1988. From 1988 to 1989, he was a member of technical staff at AT&T Bell Labs. Since 1989 he has been a faculty member of School of Computing at KAIST. His main research interest includes wireless sensor networks, 4G networks, and network security.

Daeseon Choi received the B.S. degree in computer science from Dongguk University, South Korea, in 1995, the M.S. degree in computer science from Pohang Institute of Science and Technology (POSTECH), South Korea, in 1997, and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2009. He is currently a professor at department of medical information, Kongju National University, South Korea. His research interests include information security and identity management.