A Study on Compressible and Learnable Image Encryption Methods for Untrusted Cloud Environments

July, 2020
Warit Sirichotedumrong
Tokyo Metropolitan University
# Table of Contents

1 **Introduction**  
1.1 Background ................................................................................. 4  
1.2 Aim of this thesis ................................................................. 6  
1.3 Organization ........................................................................... 7  
1.4 Notations .................................................................................. 9  

2 **Issues of Image Encryption for Untrusted Cloud Environments**  
2.1 Encryption-then-Compression Systems ......................................... 12  
2.2 Privacy-Preserving Deep Neural Networks ...................................... 15  
2.3 Contributions of this Thesis ....................................................... 19  

3 **Compressible Image Encryption for EtC Systems with JPEG Compression**  
3.1 Introduction .............................................................................. 21  
3.2 Related Work ........................................................................... 22  
3.3 EtC systems Using Grayscale-based Image Generation Methods .... 32  
3.4 Experiments and Discussion ...................................................... 38  
3.5 Summary .................................................................................... 47  

4 **Learnable Image Encryption for Privacy-Preserving Deep Neural Networks**  
4.1 Introduction .............................................................................. 51  
4.2 Related Work ........................................................................... 52  
4.3 Pixel-based Image Encryption for Privacy-Preserving Deep Neural Networks ........................................................................ 54  
4.4 Experiments and Discussions ...................................................... 61  
4.5 Summary .................................................................................... 63  

5 **Learnable Image Encryption with Independent Encryption Keys**  
5.1 Introduction .............................................................................. 65
5.2 Related Work ....................................................... 66
5.3 Pixel-based Image Encryption with Independent Encryption Keys ... 66
5.4 Robustness against Ciphertext-only Attacks .............................. 69
5.5 Experiments and Discussions .......................................... 70
5.6 Summary .................................................................. 82

6 Conclusion .................................................................. 84
  6.1 Future Works .......................................................... 85

References .................................................................... 88
Chapter 1

Introduction

1.1 Background

With the widespread of distributed systems for information processing, such as social networking and cloud computing, multimedia data is not only transmitted but also computed in cloud environments. As cloud providers are not trusted in general, it is necessary for clients to control data security issues such as data privacy, data leakage, and unauthorized data access by themselves. While many studies on secure, efficient, and flexible communications have been reported [1–4], full encryption with provable security (like RSA and AES) that generates a ciphertext is the most secure option for securing multimedia data. Moreover, their applications are limited due to the difficulty of computation and compression in the encrypted domain that cloud environments require.

Although some of the conventional encryption methods allow us to compute data in the encrypted domain, they do not consider not only compressing encrypted images with the JPEG standard but also applying encrypted images to machine learning algorithms, including deep learning ones. Therefore, many multimedia applications have been seeking a trade-off in security to enable other requirements, such as low processing demand, bit-stream compliance, and signal processing and learning in the encrypted domain. Several perceptual encryption schemes have been developed to achieve these trade-offs [5–11]. In contrast to information theory-based encryption, images encrypted by the perceptual encryption methods can be directly applied to some image processing algorithms.

A lot of perceptual image encryption methods [3–30] have been proposed for securing image data. However, some of them do not consider the compression to the encryption schemes. A hybrid compression-encryption algorithm [11] has been proposed to jointly consider both compression and encryption. In this algorithm, an image is compressed...
and encrypted at the same time. Moreover, various compression-friendly algorithms [3, 11–13, 25, 26] have been studied although they do not consider applying them to applications with JPEG compression, such as Social Network Services (SNS), and Cloud Photo Storage Services (CPSS).

Unlike file storage services and email, some cloud providers, such as SNS, Photo Sharing Services, and Google Photo, allow us to transmit or share images under free of charge or high reliable data management. Because of such scenarios, image encryption prior to image compression, which is known as Encryption-then-Compression (EtC) systems, is required in certain practical scenarios such as secure image transmission through an untrusted channel provider, as illustrated in Fig. 1.1. In [3, 17, 18], the compression methods for EtC systems have been proposed, but they are not applicable to EtC systems with JPEG compression. Due to such a situation, several perceptual encryption schemes [19–22] have been proposed for EtC systems under the use of the JPEG standard, but they do not have enough robustness against image recompression forced by cloud providers, such as social network service (SNS) ones.

Some of the perceptual schemes [19, 20, 24, 27] were demonstrated to be applied to traditional machine learning (ML) algorithms, such as support vector machine (SVM), k-nearest neighbors (KNN), and random forest, even under the use of the kernel trick [31], but they cannot be applied to deep learning algorithms yet. To apply encrypted images to deep learning algorithms, a learnable image encryption scheme [28] has been also proposed. The learnable image encryption scheme [28] applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to deep neural networks (DNNs). However, it cannot avoid the influence of data augmentation in the encrypted domain, and the performance is degraded compared with that of using plain images.

To address these issues with compressible and learnable encryption, this thesis
Chapter 1 Introduction

presents three perceptual image encryption schemes. The first image encryption aims to generate compressible encrypted images that are applicable to JPEG compression. This encryption has strong robustness against various attacks, while maintaining high compression performance. The second encryption allows us not only to train a model for DNNs with encrypted images, but also to use images without visual information on plain images as test ones. In addition, this learnable encryption enables us to directly carry out some data augmentation operations in the encrypted domain. The third one, which is an extension of the second one, makes the sensitive management of security keys unnecessary. In other word, independent encryption keys can be assigned to every image by using the third encryption. Therefore, training and testing images for DNNs can be encrypted by using different keys under the use of this encryption.

1.2 Aim of this thesis

Most perceptual encryption schemes have not given much consideration on compressible and learnable image encryption, namely, an encrypted image has to be compatible to international compression standards, such as JPEG standard, or be able to apply to deep learning algorithms. To address these problems, the aims of this thesis are concluded as follows.

First, this thesis presents a novel compressible image encryption scheme for EtC systems with JPEG compression that considers the robustness against image recompression forced by SNS providers, and the robustness against ciphertext-only attacks (COA), such as jigsaw puzzle solver and brute-force attacks. Prior to image encryption steps, the proposed encryption method firstly converts an RGB color image into YCbCr color image, and then, generates a grayscale-based image that has one color channel. In addition, the proposed grayscale-based image generation methods allow us to perform color sub-sampling as JPEG compression, although an encrypted image has only one-color channel. In addition, a novel JPEG quantization table, which is especially designed for the proposed grayscale-based image, has been considered to enhance compression performance of JPEG compression.

Next, it presents a learnable encryption method that is not only to apply images without visual information to DNNs, but also to consider data augmentation in the encrypted domain. Data augmentation aims to enlarge the number of data points used for training and it enables avoiding overfitting of DNNs. Therefore, it is necessary for data augmentation to be carried out in cloud servers in order to reduce communication costs, namely, data augmentation has to be done in the encrypted domain. The proposed learnable encryption, which is a pixel-based encryption method, demonstrates the capability of applying image encryption to privacy-preserving DNNs with independent
keys as well as considering data augmentation in the encrypted domain.

In addition to the pixel-based image encryption, while the conventional privacy-preserving DNNs generally employ a common security key, the use of independent keys, where training and testing images are encrypted under the use of different keys, is established in this thesis. Therefore, the pixel-based image encryption with independent keys allows the use of privacy-preserving DNNs without key management.

1.3 Organization

As illustrated in Fig. 1.2, this thesis consists of six chapters as follows.

Chapter 1 describes the background and challenges of this research field, the purpose of this research, and the structure of the thesis.

Chapter 2 addresses the issues of image encryption for untrusted cloud environments
which describe the problems of EtC systems and privacy-preserving DNNs. In addition, the contributions of this thesis are summarized.

Chapter 3 describes a novel grayscale-based block scrambling image encryption scheme using YCbCr color space that has been proposed to not only to provide almost the same security level as the conventional grayscale-based encryption scheme, but also to improve the compression performance for EtC systems with JPEG compression. In addition, the scheme allows us to consider the color sub-sampling operation which can improve the compression performance, although the encrypted images have no color information. The grayscale-based scheme firstly converts an RGB color image into YCbCr color image, and then, generates a grayscale-based image using the proposed grayscale-based image generation method, which consists of two types: grayscale-based image without color sub-sampling and grayscale-based image with color sub-sampling. Then, encryption steps are performed to each block to generate an encrypted image. In addition, a novel JPEG quantization table, which is especially designed for the proposed grayscale-based image, has been considered to enhance compression performance of JPEG compression. The experimental results showed the signification improvement of the proposed scheme in terms of the compression performance, robustness against image recompression forced by the providers, and robustness against COA.

Chapter 4 focuses on a learnable image encryption scheme with a common security key, which is a pixel-based image encryption method, has been proposed to not only to apply images without visual information to DNNs, but also to consider data augmentation in the encrypted domain. In addition, an adaptation network, which is added prior to DNNs, has been presented to enhance the classification performance of DNNs by obtaining the representations of each pixel before passing through the conventional DNNs. As a result, the proposed scheme relaxes the use of image encryption on DNNs in terms of signal processing in the encrypted domain and classification performance. The experimental results showed that the pixel-based image encryption outperforms the conventional encryption methods, including the block-based one for EtC systems, in terms of the classification performance, although data augmentation is carried out in the encrypted domain.

Chapter 5 addresses a new idea of privacy-preserving DNNs using the pixel-based image encryption that protects visual information and considers the use of independent encryption keys for training and testing images. Namely, all training images and testing images are independently encrypted by using different encryption keys. Therefore, there is no need to manage encryption keys. In addition, the pixel-based image encryption with independent keys allows us to train a DNN model with encrypted images and then test it with plain images. Several experiments were conducted to confirm the effectiveness of learnable image encryption with independent keys in terms of image classification accuracy and robustness against COA. The experimental results demon-
Chapter 1 Introduction

It was demonstrated that the pixel-based method with independent encryption keys can maintain the classification performance and provide higher robustness against COA than the pixel-based one with a common security key. Moreover, the results proved that the pixel-based method with different keys was able to classify plain images as well as encrypted images.

Chapter 6 concludes this thesis and then presents prospects of further developments.

1.4 Notations

The following notations are used throughout this thesis:

- A full color image in RGB color space with $U \times V$ pixels is denoted by $I_{RGB}$ and is composed of red ($i_R$), green ($i_G$), and blue ($i_B$) color channels.
- $n$ denotes the number of pixels of $I_{RGB}$.
- A pixel of $i_R$, $i_G$, or $i_B$ is denoted by $p$.
- $p_R$, $p_G$, and $p_B$ denote pixel values of $i_R$, $i_G$, and $i_B$, respectively.
- $I_{YCbCr}$ denotes a color image in YCbCr color space.
- $I_{YCbCr}$ consists of three individual channels which can be represented by $i_Y$, $i_{Cb}$, and $i_{Cr}$.
- $i'_Cb$ and $i'_Cr$ denote the sub-sampled chrominance.
- $I_g$ denotes a grayscale-based image which is generated from $I_{YCbCr}$.
- An image encryption algorithm is represented by $Enc(.)$.
- An encrypted image $I_e$ is written as $I_e = Enc(I)$.
- $K_g$ denotes a set of secret keys used for generating grayscale-based encrypted images, where $K_g = \{K_1, K_2, K_3\}$.
- $I_{ec}$ denotes an encrypted JPEG image produced from $I_e$.
- $\hat{I}_{ec}$ denotes an encrypted JPEG image downloaded from SNS providers.
- $\hat{I}_e$ represents an encrypted image decompressed from $\hat{I}_{ec}$.
- $\hat{I}_g$ denotes a grayscale-based image decrypted from $\hat{I}_e$. 


Chapter 1 Introduction

- A color image in YCbCr color space, which is decomposed from $\hat{I}_g$, is $\hat{I}_{YCbCr}$.
- $\hat{I}$ denotes a color image in RGB color space transformed from $\hat{I}_{YCbCr}$.
- $T$ denotes a set of plain images for training, which consists of $g$ images, where $T = \{I_{t_1}, I_{t_2}, \ldots, I_{t_g}\}$.
- A set of plain images for testing is represented by $Q$ which includes $h$ testing images so that $Q = \{I_{q_1}, I_{q_2}, \ldots, I_{q_h}\}$.
- $X_T$ and $X_Q$ denote sets of input images used to train and test a model, respectively.
- A secret key $K$ denotes a set of keys used for image encryption.
- $K_T$ denotes a secret key set used for encrypting $T$, namely, $K_T = \{K_{t_1}, K_{t_2}, \ldots, K_{t_g}\}$. For example, $Enc(I_{t_1})$ is obtained by encrypting $I_{t_1}$ with $K_{t_1}$.
- A set of secret key $K_Q$ is utilized for encrypting $Q$ where $K_Q = \{K_{q_1}, K_{q_2}, \ldots, K_{q_h}\}$. For example, $Enc(I_{q_h})$ is obtained by encrypting $I_{q_h}$ with $K_{q_h}$.
- $f_d(\cdot)$ denotes a data augmentation process.
Chapter 2

Issues of Image Encryption for Untrusted Cloud Environments

This chapter addresses issues of image encryption for untrusted cloud environments, including the problems of encryption-then-compression (EtC) systems and privacy-preserving DNNs.

2.1 Encryption-then-Compression Systems

The traditional way of securely transmitting images is to use a Compression-then-Encryption (CtE) system, namely, an image is compressed before image encryption. However, image encryption prior to image compression is required in certain practical scenarios such as secure image transmission through an untrusted channel provider. Meanwhile, it is known that almost all social network service (SNS) providers manipulate images uploaded by users, e.g., rescaling image resolution and recompressing with different parameters, for decreasing the data size of images [32, 33]. As a result, the quality of images recompressed by SNS providers is degraded by image manipulation on social media. To avoid the distortion, the following requirements have to be satisfied.

- The encrypted images are compatible with the JPEG standard.
- The compression efficiency for the encrypted images is almost the same as that for the original ones under the JPEG standard.
- The encrypted images have robustness against various attacks [34–36].

Due to such situations, block scrambling-based image encryption schemes [18–23] were proposed for EtC systems, in which a user wants to securely transmit an image.
Chapter 2 Issues of Image Encryption for Untrusted Cloud Environments

Figure 2.1: EtC system.

Figure 2.2: Sub-sampling in JPEG encoder and interpolation of chroma components in JPEG decoder

$I$ to an audience or a client via, Social Network Services (SNS) or Cloud Photo Storage Services (CPSS) providers, as shown in Fig. 2.1. The privacy of the image to be shared can be controlled by the user unless the user does not give the secret key $K_g$ to the providers, although the image is generally recompressed by the providers. In contrast, in CtE systems, the disclosure of non-encrypted images is required before the recompression.

Let $B_x$ and $B_y$ denote integers and are divisor of $U$ and $V$, respectively. Although images encrypted by using the color-based image encryption scheme are compatible with the JPEG standard and have almost the same compression performance as non-encrypted ones when using $B_x = B_y = 16$, there is a limitation that the scheme cannot achieve. The possible smallest block size of the scheme is $16 \times 16$ to avoid the effect of color sub-sampling. As shown in Fig. 2.2, if 4:2:0 color sub-sampling is applied to an encrypted image, each $8 \times 8$-block in the sub-sampled chroma components consists of four $4 \times 4$-blocks from different $8 \times 8$-block, which have low correlation among the blocks.
Chapter 2 Issues of Image Encryption for Untrusted Cloud Environments

Figure 2.3: Example of the decrypted images with the color-based scheme including block distortion ($B_x = B_y = 8, Q_f = 100$, and 4:2:0 sub-sampling)

As a result, block distortion is generated due to the interpolation of the sub-sampled chroma components with discontinuous pixel values. An example of the decrypted images including block distortion, where the JPEG quality factor ($Q_f$) of uploaded images was equal to 100, is shown in Fig. 2.3. If the block size is smaller than $16 \times 16$ pixels, such as $8 \times 8$ pixels, the compression performance decreases, and some block distortion is generated. This is because when color sub-sampling is applied to the chroma components ($C_b$ and $C_r$) of a color image in a JPEG encoder, the interpolation is carried out to the sub-sampled chroma components ($C'_b$ and $C'_r$) to reconstruct the spatial resolution as that of the original image in a decoder.

In addition, images encrypted by using the conventional color-based image encryption [19, 20] has not enough robustness against jigsaw puzzle solver attacks [34–36], as
Chapter 2 Issues of Image Encryption for Untrusted Cloud Environments

illustrated in Fig. 2.4.

2.2 Privacy-Preserving Deep Neural Networks

The spread of deep neural networks (DNNs) has greatly contributed to solving complex tasks for many applications [37–39], such as for computer vision, biomedical systems, and information technology. Deep learning utilizes a large amount of data to extract representations of relevant features, so the performance in applications is significantly improved [40, 41]. In addition, DNNs have been deployed in security-critical applications, such as facial recognition, biometric authentication, and medical image analysis.

Recently, it is very popular for data owners to utilize cloud servers to compute and process a large amount of data instead of using local servers. This is because cloud environment provides the flexibility and cost-saving computation. However, since cloud servers are semi-trusted, data privacy, such as personal information and medical records, may be revealed in cloud computing. Therefore, it is necessary to protect data privacy in the cloud environment, and privacy-preserving DNNs have become an urgent challenge. This thesis focuses on protecting data privacy by encrypting data before uploading to the cloud environment.

Various perceptual encryption methods have been proposed that generate images without visual information [3–30], although information theory-based encryption (like RSA and AES) generates a ciphertext. In contrast to information theory-based encryption, images encrypted by the perceptual encryption methods can be directly applied to some image processing algorithms.

Even though some perceptual encryption methods [19, 20, 24, 27] can be applied to traditional machine learning (ML) algorithms, such as support vector machine (SVM), k-nearest neighbors (KNN), and random forest, even under the use of the kernel trick [31], these methods have never been applied to DNNs. Tanaka’s scheme [28] is the first perceptual encryption for privacy-preserving DNNs that uses encrypted images for both training and testing DNN models. This scheme [28] applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to DNNs. However, Tanaka’s scheme [28] cannot avoid the influence of data augmentation in the encrypted domain.

To avoid the performance degradation, image encryption methods for privacy-preserving DNNs should meet the following requirements.

• **Visual information protection**: to protect an individual, the time, and the location of a taken photograph.

• **Lightweight computation**: to train and test privacy-preserving DNNs with the
same computational cost as the ordinal DNNs with plain images.

- **Low damage to DNNs**: to maintain the performance of DNNs as the ordinal DNNs with plain images.

- **Data augmentation in encrypted domain**: to carry out data augmentation on encrypted images.

- **Security**: to provide robustness against COA.

This thesis aims to propose image encryption that satisfies all requirements mentioned above.

Figure 2.5 illustrates the training frameworks used for image classification used in this thesis. Encrypted images that have no visual information are utilized for training and testing DNNs.

### 2.2.1 Visual Information Protection

Security mostly refers to protection from adversarial forces. This thesis focuses on protecting visual information that allows us to identify an individual, the time, and the
location of the taken photograph. Semi-trusted cloud providers and unauthorized users are assumed to be adversaries.

A lot of perceptual encryption methods have been proposed for protecting the visual information of images [8, 11, 19, 20, 24, 25, 27–31, 42–45]. Perceptual image encryption generates visually protected images, which have pixel values, but information theory-based encryption (like RSA and AES) generates a ciphertext. Therefore, the encrypted images can be directly applied to some image processing algorithms.

For example, encryption methods [8, 42] have been proposed not only for visually protecting privacy and security but also for matching and searching images in the encrypted domain.

Compressible encryption methods have been also proposed that consider both security and efficient compression so that they can be adapted to cloud storage and network sharing [11, 19, 20, 24, 27, 43–45]. Some of them [19, 20, 24, 27] can be applied to traditional ML algorithms, such as SVM, KNN, and random forest, even under the use of the kernel trick [31]. However, these methods have never been applied to DNNs.

There are two encryption methods [28–30] that use encrypted images for both training and testing DNN models. One, the first perceptual encryption for privacy-preserving DNNs, is Tanaka’s scheme [28]. Tanaka’s scheme applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to DNNs. The other is a pixel-based encryption method that directly applies encrypted images to DNNs [29, 30].

Images encrypted by using perceptual encryption are illustrated in Fig 2.6(b), (c), (d), and (e), where Fig 2.6(a) is the original one. It was illustrated that encrypted images have no visual information. In addition, images encrypted by the perceptual encryption methods have pixel values and can be directly applied to some image processing algorithms.

### 2.2.2 Security Problems with Machine Learning

The security issues with ML are classified into three classes in terms of the goals of an attack [46–53]: reliability of results, model protection, and data protection. For reliability of results, some adversaries aim to confuse users by misclassifying the results of ML, such as imperceptible adversarial perturbation, called ”adversarial examples” [51–53]. Adversarial examples cause DNNs to misclassify with high confidence or force them to classify a targeted label. Although various methods have been proposed to defend against adversarial examples [54–56], there is no robust model yet.

Model protection means to protect model parameters including hyper-ones and learned-ones [46, 47]. Model extraction attacks [46, 47], which aim to extract an equivalent or near-equivalent ML model, are vulnerable to ML models.
Figure 2.6: Examples of images. (a) Original image \((U \times V = 96 \times 96)\). (b) Image encrypted by block-based encryption [19, 20] (Block size = 4 \times 4). (c) Image encrypted by pixel-based image encryption [25]. (d) Image encrypted by grayscale-based image encryption [23] (Block size = 4 \times 4). (e) Image encrypted by block-based encryption [28] (Block size = 4 \times 4).

Alternatively, data protection means to avoid a threat to training and testing data from adversaries. Namely, the data that contains confidential information, such as personal information and medical records, has to be protected. For data protection, there are three issues that have to be considered. One is membership inference [48], which has been proposed to identify whether a data record was trained by a model. The second is model inversion attacks [49, 50] that aim to obtain the trained data from the trained model. The other is untrusted cloud environments which are also vulnerable for data privacy because cloud servers are generally semi-trusted. Thus, data privacy may be revealed during computation carried out in the cloud server.

Various encryption methods, such as homomorphic encryption (HE) [57–65] and perceptual encryption [11, 19, 20, 24, 27–31, 43–45], have been proposed not only to protect privacy of data but also to be available for cloud computing.

This thesis focus on data protection for training and testing DNN models. Moreover, even if membership inference attacks can identify whether or not an image encrypted by perceptual encryption methods was utilized to train the model because the proposed method protects the visual information of images, an individual, the time, and location of a taken photograph are not exposed. Hence, protecting visual information allows us to provide robustness against membership inference attacks and model inversion attacks.
2.3 Contributions of this Thesis

The contributions of this thesis are summarized as follows (See Table 2.1).

- It addresses the issues of image encryption for EtC systems with JPEG compression and privacy-preserving DNNs.

- This thesis proposes a novel grayscale-based image encryption scheme that is not only to enhance compression performance but also to provide robustness against image recompression forced by SNS providers.

- A novel JPEG quantization table, which is especially designed for grayscale-based images, has been described, and its effectiveness has been confirmed in terms of the compression performance.

- It proposes a novel pixel-based image encryption scheme for privacy-preserving DNNs. This scheme enables us to apply images without visual information to DNNs.

- In addition to the pixel-based encryption scheme, this thesis also delivers a new idea for privacy-preserving DNNs using the pixel-based image encryption that protects visual information and considers the use of independent encryption keys for training and testing images.

- It provides several experiments to confirm the effectiveness of all proposed image encryption schemes, including the grayscale-based image encryption scheme, and the pixel-based one.
Chapter 3

Compressible Image Encryption for EtC Systems with JPEG Compression

3.1 Introduction

Image encryption prior to image compression is required in certain practical scenarios such as secure image transmission through an untrusted channel provider. Encryption-then-Compression (EtC) systems [19–24] are used in such scenarios. In this chapter, we focus on EtC systems with JPEG compression, although the traditional way of transmitting images is to use a Compression-then-Encryption (CtE) systems.

As described in Chapter 2, block scrambling-based image encryption schemes were proposed for EtC systems, in which a user wants to securely transmit an image to an audience or a client via Social Network Services (SNS) and Cloud Photo Storage Services (CPSS) as shown in Fig. 2.1. In the color-based image encryption scheme [18–22], a full color image $I_{RGB}$ with $U \times V$ pixels is divided into non-overlapping blocks each with $B_x \times B_y$ pixels; then block scrambling-based encryption steps are applied to the divided blocks. As a result, the encrypted image consists of three color channels.

Although images encrypted by using the color-based image encryption scheme are compatible with the JPEG standard and have almost the same compression performance as non-encrypted ones when using $B_x \times B_y = 16 \times 16$, there is a limitation that the scheme cannot achieve. To avoid the effect of color sub-sampling the possible smallest block size is $16 \times 16$.

In this chapter, a grayscale-based image encryption that has almost the same compression performance as non-encrypted images as well as the color-based scheme is described. In addition, a new quantization table, which is designed for grayscale-based
images, is explained. It is noteworthy that a grayscale-based image differs from a grayscale image. A grayscale image is an image with a range of shades of gray without color information, where the darkest possible shade is black, and the lightest possible shade is white. In contrast, a grayscale-based image is artificially generated from three color channels of a color image, and pixels in the image have color information. Therefore, the original color image can be reconstructed from the grayscale-based image. Although the conventional grayscale-based image encryption scheme [23, 24], which is an extension of the color-based EtC systems [19–22], has been proposed to enhance the robustness against several attacks, such as brute-force and jigsaw puzzle solver attacks [34–36], and also avoid the effect of color subsampling, the compression performance is degraded compared with the color-based image encryption scheme [19–22]. This is because a grayscale-based image is generated from RGB components of a full-color image, so the correlation between RGB color channels is not used.

Consequently, this chapter aims to propose a novel grayscale-based encryption scheme that has almost the same compression performance as non-encrypted images. A novel JPEG quantization table which is especially designed for grayscale-based images is also considered. In experiments, the effectiveness of the grayscale-based image encryption is evaluated and discussed in various metrics, including JPEG compression performance, robustness against image recompression, and robustness against ciphertext-only attacks. The JPEG compression performance of images encrypted by the grayscale-based image encryption is evaluated in terms of rate-distortion curves, which is the relation between peak signal-to-noise ratio (PSNR) values and bits per pixel. Moreover, encrypted images are uploaded to Facebook and Twitter, and then downloaded to evaluate the robustness against image recompression carried out by SNS providers.

3.2 Related Work

3.2.1 JPEG Compression

JPEG Encoding

This thesis focuses on JPEG compression which is a lossy image compression method since it is one of the most widely used image compression standards, and most SNS providers [32, 33] support JPEG. JPEG encoding of color images consists of six steps (See Fig. 3.1):

1) Perform the color transformation from RGB color space to YCbCr color space.

2) Sub-sample the Cb and Cr components to reduce the spatial resolution.
3) Divide the image into 8-by-8 blocks.

4) Apply the 2-D discrete cosine transform (DCT) to each block.

5) Carry out block-based quantizing to brightness and chroma components with luminance table (Y-table) and chrominance table (CbCr-table), respectively.

6) Carry out entropy coding using Huffman coding.

The quality of compressed images varies based on the change of quality factor $Q_f$ ($1 \leq Q_f \leq 100$), which is a parameter that controls the quantization tables. $Q_f = 100$ gives the best quality, and $Q_f = 1$ provides the worst quality.

For a given $Q_f$, the elements of associated quantization matrix $T_q(i, j)$, which also referred to as the quantization steps, are obtained using the following relation, where $1 \leq i \leq 8$ and $1 \leq j \leq 8$.

$$T_q(i, j) = \left\lfloor \frac{50 + S \times D(i, j)}{100} \right\rfloor,$$  \hspace{1cm} (3.1)

where $\lfloor . \rfloor$ denotes the rounding function, and $D(i, j)$ are the elements of default quantization matrices, as illustrated in Fig. 3.2. Note that $S$ is a parameter that is varied by $Q_f$ and can be calculated as follows.

$$S = \begin{cases} 200 - 2Q_f, & Q_f \geq 50 \\ \frac{5000}{Q_f}, & otherwise \end{cases}.$$  \hspace{1cm} (3.2)

Let $DCT_m(i, j)$ be the DCT coefficient of the $m^{th}$ block at the position $(i, j)$, where $1 \leq i \leq 8$ and $1 \leq j \leq 8$. The quantized DCT coefficients ($B_m(i, j)$) are computed with

$$B_m(i, j) = \left\lfloor \frac{DCT_m(i, j)}{T_q(i, j)} \right\rfloor.$$  \hspace{1cm} (3.3)
In addition, there are three sub-sampling ratios in the JPEG standard, referred to as 4:2:0 (reduction by a factor of 2 in both horizontal and vertical directions), 4:2:2 (reduction by a factor of 2 in horizontal direction), and 4:4:4 (no sub-sampling). The JPEG image of a color image is generated by performing steps 3) to 6) on the brightness component Y and sub-sampled chroma components Cb and Cr independently.

**JPEG Decoding**

To obtain decompressed images from JPEG images, the following steps of JPEG decoding are carried out (See Fig. 3.3).

1) Carry out entropy decoding using Huffman decoding.

2) Divide the image into 8-by-8 blocks.

3) Apply the inverse discrete cosine transform (DCT) to each block.

4) Carry out block-based dequantizing with a quantization matrix.

5) Interpolate the Cb and Cr components.

6) Perform the color transformation from YCbCr color space to RGB color space.

In JPEG decoding, since Cb and Cr components are interpolated, block distortion may be generated if an image consists of chroma components, as demonstrated in Chapter 2. Therefore, images encrypted by using the color-based image encryption [18–
22] suffer from this interpolation because of the sub-sampled chroma components with discontinuous pixel values. In contrast, images encrypted the grayscale-based image encryption [23] have robustness against interpolation because the encrypted images have no chroma component.

3.2.2 Image Manipulation Specifications on Social Network Services

Table 3.1 summarizes the relationship between uploaded and downloaded JPEG images of typical SNS and CPSS providers in terms of sub-sampling ratios, the quality factor ($Q_f$), and the maximum resolutions [32]. $Q_{fu}$ and $Q_{fd}$ denote the uploaded and downloaded quality factors, respectively.

Providers do not resize uploaded images when the resolution of uploaded images is less than or equal to the maximum resolution that each provider decided. For example, if the resolution of an image uploaded to Twitter is not larger than $4096 \times 4096$ pixels, the uploaded image is not resized.

The quality of images downloaded from SNS and CPSS providers is generally degraded due to recompression forced by the providers. As shown in Table 3.1, a color JPEG image uploaded to Facebook is always recompressed into the new JPEG image with 4:2:0 color sub-sampling. Therefore, images encrypted by the color-based scheme are affected by the recompression, even when they were compressed with 4:4:4 sub-sampling. In comparison, when grayscale-based images are uploaded to Facebook, the color sub-sampling is not carried out, although they are recompressed with different quality factors from those of uploaded images. Images encrypted by using the proposed scheme are not affected by the effect of color sub-sampling thanks to the use of grayscale-based images [23, 24].
Table 3.1: Relationship between uploaded JPEG files and downloaded ones in terms of sub-sampling ratios and the maximum resolutions. Providers do not resize uploaded images when their resolutions are less than or equal to the maximum resolutions, e.g. the maximum resolutions of Twitter and Tumblr are 4096 × 4096 and 1280 × 1280, respectively [29].

<table>
<thead>
<tr>
<th>Provider (Maximum resolution)</th>
<th>Uploaded JPEG file</th>
<th>Downloaded JPEG file</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sub-sampling ratio</td>
<td>$Q_{fu}$</td>
</tr>
<tr>
<td></td>
<td>low</td>
<td>No recompression</td>
</tr>
<tr>
<td></td>
<td>high</td>
<td>$Q_{fd}$</td>
</tr>
<tr>
<td>Twitter (Up to 4096 × 4096 pixels)</td>
<td>4:4:4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:2:0</td>
<td>1, 2, ..., 84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No recompression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85, 86, ..., 100</td>
</tr>
<tr>
<td></td>
<td>(Grayscale)</td>
<td>1, 2, ..., 84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No recompression</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85, 86, ..., 100</td>
</tr>
<tr>
<td></td>
<td>(Grayscale)</td>
<td></td>
</tr>
<tr>
<td>Facebook (HQ, Up to 2048 × 2048 pixels)</td>
<td>4:4:4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:2:0</td>
<td>1, 2, ..., 100</td>
</tr>
<tr>
<td></td>
<td>(Grayscale)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>71, 72, ..., 85</td>
</tr>
<tr>
<td>Facebook (LQ, Up to 960 × 960 pixels)</td>
<td>4:4:4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:2:0</td>
<td>1, 2, ..., 100</td>
</tr>
<tr>
<td></td>
<td>(Grayscale)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>66, ..., 90</td>
</tr>
<tr>
<td>Google Photos (HQ, Up to 16 Megapixels)</td>
<td>4:4:4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:2:0</td>
<td>1, 2, ..., 100</td>
</tr>
<tr>
<td></td>
<td>(Grayscale)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>66, ..., 90</td>
</tr>
<tr>
<td>Tumblr (Up to 1280 × 1280 pixels)</td>
<td>4:4:4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:2:0</td>
<td>1, 2, ..., 100</td>
</tr>
<tr>
<td></td>
<td>(Grayscale)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No recompression</td>
</tr>
<tr>
<td>Google+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flickr</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


3.2.3 Color-based Image Encryption

In the previous works [18–22], a full-color image \( I_{RGB} \) with \( U \times V \) pixels is divided into non-overlapping blocks each with \( B_x \times B_y \); then four steps of block scrambling-based encryption are applied to the divided blocks as follows (See Fig. 3.5).

1) Randomly permute the divided blocks by using a sequence of random integers generated by a secret key \( K_1 \).

2) Rotate and invert each block randomly (See Fig. 3.4) by using a random integer generated by a key \( K_2 \).

3) Apply negative-positive transformation to each block by using a random binary integer generated by a key \( K_3 \), where \( K_3 \) is commonly used for all color components. In this step, a transformed pixel value in the \( i \)-th block \( B_i, p' \), is calculated using

\[
p' = \begin{cases} 
p & (r(i) = 0) 
p \oplus (2^L - 1) & (r(i) = 1) \end{cases},
\]

where \( r(i) \) is a random binary integer generated by \( K_3 \), \( p \in B_i \) is the pixel value of the original image with \( L \) bit per pixel, and \( \oplus \) is the bitwise exclusive-or operation. The value of occurrence probability \( P(r(i)) = 0.5 \) is used to invert bits randomly.

4) Shuffle three color components in each block by using an integer randomly selected from six integers generated by a key \( K_4 \) as shown in Table 3.2.
Chapter 3 Compressible Image Encryption for EtC Systems with JPEG Compression

Table 3.2: Permutation of color components for a random integer. For example, if the random integer is equal to 2, the red component is replaced by the green one, and the green component is replaced by the red one while the blue component is not replaced.

<table>
<thead>
<tr>
<th>Random Integer</th>
<th>Three Color Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
</tr>
<tr>
<td>0</td>
<td>R</td>
</tr>
<tr>
<td>1</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>G</td>
</tr>
<tr>
<td>3</td>
<td>G</td>
</tr>
<tr>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
</tr>
</tbody>
</table>

3.2.4 Grayscale-based Image Encryption

A conventional grayscale-based image encryption has been proposed to avoid the effect of color sub-sampling by encrypting $I_{RGB}$ into the encrypted image ($I_e$) which consists of only one color channel [23]. In order to generate $I_e$, four processing steps are carried out as follows (See Fig. 3.6).

1) Split $I_{RGB}$ into three RGB channels, $i_R$, $i_G$, and $i_B$ and concatenate all channels to generate a conventional grayscale-based image ($I_g^{RGB}$) with $3(U \times V)$ pixels as shown in Fig. 3.7.

2) Divide $I_g^{RGB}$ into blocks each with $B_x \times B_y$ and then randomly permute the divided blocks by using a random integer generated by a secret key $K_1$.

3) Rotate and invert each block randomly by using a random integer generated by a key $K_2$.

4) Apply negative-positive transformation to each block by using a random binary integer generated by a key $K_3$. This step is carried out by using the same process as Step 3 of the color-based image encryption in Section 3.2.3.

Since images encrypted by using the grayscale-based image encryption contain only one color channel, the encrypted images can avoid the effect of color sub-sampling. Therefore, the scheme allows us to use $8 \times 8$ as the block size, which is smaller than that with the color-based one, as shown in Fig. 3.8(c). Moreover, the number of blocks
is larger because the block size is smaller, and the number of pixels of encrypted images is larger. As a result, the scheme enhances the security against various attacks [23] and also provides the robustness against color sub-sampling due to the use of grayscale-based images. However, the compression performance of the conventional grayscale-based scheme decreases, because the correlation between RGB color channels is not used.

### 3.2.5 Security against Cipherertext-only Attacks

**Brute-force Attack**

In the conventional color-based encryption scheme, if an image with $U \times V$ pixels is divided into blocks with $B_x \times B_y$ pixels, the number of blocks $n_b$ is given by

$$n_b = \left\lfloor \frac{U}{B_x} \right\rfloor \times \left\lfloor \frac{V}{B_y} \right\rfloor,$$  \hspace{1cm} (3.5)
where $\lfloor \cdot \rfloor$ is a function that rounds down to the nearest integer. The four block scrambling-based processing steps are then applied to the divided image.

The key space of the block scrambling (Step 1) $N_{Scr}(n_b)$, which is the number of permutation of $n_b$ blocks, is given by

$$N_{Scr}(n_b) = n_b!,$$  \hspace{1cm} (3.6)

Similarly, the key spaces of other encryption steps are given as

$$N_{Rot}(n_b) = 4^n_b, \quad N_{Inv}(n_b) = 4^n_b, \quad N_{Rot&Inv}(n_b) = 8^n_b$$

$$N_{NP}(n_b) = 2^n_b, \quad N_{CS}(n_b) = (3P_3)^{n_b} = 6^n_b$$  \hspace{1cm} (3.7) \hspace{1cm} (3.8)

where $N_{Rot}(n_b)$ and $N_{Inv}(n_b)$ are the key spaces of the block rotation and block inversion, and $N_{Rot&Inv}(n_b)$ is the key space of the encryption combining them (Step 2). Note that $N_{Rot&Inv}$ is the key space considering the collision between block rotation and inversion. Namely, rotating pieces 180 degrees is the same operation as inverting them horizontally and vertically. $N_{NP}(n_b)$ and $N_{CS}(n_b)$ are the key spaces of the negative-positive transformation (Step 3) and color component shuffling (Step 4) respectively. Consequently, the key space of images encrypted by using all the proposed encryption steps, $N_A(n_b)$, is represented by

$$N_A(n_b) = N_{Scr}(n_b) \cdot N_{Rot&Inv}(n_b) \cdot N_{NP}(n_b) \cdot N_{CS}(n_b)$$

$$= n_b! \cdot 8^n_b \cdot 2^n_b \cdot 6^n_b.$$  \hspace{1cm} (3.9)

In comparison, since an image encrypted by using the conventional grayscale-based image encryption scheme is generated from $I^{RGB}_{g}$ with $3(U \times V)$ pixels, the number of blocks of the grayscale-based scheme ($n_g$) is three times larger than that of the color-based scheme. $n_g$ is given by

$$n_g = 3n_b.$$  \hspace{1cm} (3.10)
Unlike the color-based scheme, color shuffling is not carried out by the conventional grayscale-based encryption scheme. Thus, the key space of the grayscale-based image encryption is calculated by

\[
N_B(n_b) = N_{Scr}(3n_b) \cdot N_{Rot&Inv}(3n_b) \cdot N_{NP}(3n_b)
\]

\[
= 3n_b! \cdot 8^{3n_b} \cdot 2^{3n_b} \gg N_A(n_b),
\]

where \(n_b\) is the number of blocks calculated from \(I\) with \(U \times V\) pixels. Although the color shuffling is not applied to the scheme, the number of blocks is larger, as shown in (3.11). Therefore, the grayscale-based image encryption enhances the robustness against brute-force attacks.

**Jigsaw Puzzle Solver Attack**

The extended jigsaw puzzle solvers (JPS) for block scrambling-based image encryption [34–36] have been proposed to assemble encrypted images including rotated, inverted, negative-positive transformed, color component shuffled blocks. It has been shown that assembling encrypted images becomes difficult when the encrypted images are under the following conditions [34–36].

- Number of blocks is large.
- Block size is small.
- Encrypted images include JPEG distortion.
- Encrypted images have less color information.

Since most conventional JPS utilize color information to assemble puzzles, reducing the number of color channels in each pixel makes assembling encrypted images much more difficult. Thus, the grayscale-based encryption scheme has a higher security level than that of the color-based scheme because it provides a large number of blocks, the small block size, and less color information.

The following three evaluation metrics [66, 67] are used to evaluate the robustness against JPS.

**Direct comparison** \((D_c)\): represents the ratio of the number of pieces of an assembled image which are in the correct position. \(D_c\) for image \(I_d\), namely, \(D_c(I_d)\) is calculated as

\[
D_c(I_d) = \frac{1}{n} \sum_{i=1}^{n} d_c(i),
\]

\[
d_c(i) = \begin{cases} 
1, & \text{if } I_d(i) \text{ is in the correct position} \\
0, & \text{otherwise}
\end{cases}
\]
where \( I_d(i) \) represents the position of a piece \( i \) in image \( I_d \)

**Neighbor comparison** \( (N_c) \): is the ratio of the number of correctly joined blocks. \( N_c \) for image \( I_d \), namely, \( N_c(I_d) \) is calculated as

\[
N_c(I_d) = \frac{1}{N_{bou}} \sum_{k=1}^{N_{bou}} n_c(k),
\]

\[
n_c(k) = \begin{cases} 
1, & \text{if } b_k \text{ is joined correctly} \\
0, & \text{otherwise} 
\end{cases}
\]

where \( N_{bou} \) is the number of boundaries among pieces in \( I_d \), and \( b_k \) is the \( k \)-th boundary in \( I_d \). For an image with \( u \times v \) blocks, there are \( N_{bou} = 2uv - u - v \) boundaries in the image.

**Largest component** \( (L_c) \): is the ratio of the number of the largest joined blocks that have correct adjacencies to the number of blocks in an image. \( L_c \) for image \( I_d \), namely, \( L_c(I_d) \) is calculated as

\[
L_c(I_d) = \frac{1}{n} \max_j \{ l_c(I_d, j) \}, j = 1, 2, \ldots, m
\]

where \( l_c(I_d, j) \) is the number of blocks in the \( j \)-th partial correctly assembled area, and \( m \) is the number of partial correctly assembled areas.

In the measures, \( D_c, N_c, L_c \in [0, 1] \), a larger value means higher compatibility.

### 3.3 EtC systems Using Grayscale-based Image Generation Methods

This section proposes new grayscale-based image generation methods using YCbCr color space which aim not only to enhance security, but also to improve the compression performance of encrypted images.

#### 3.3.1 Encryption Procedure

To generate an encrypted image \( (I_e) \), the following steps are carried out (See Fig. 3.9).

1) Transform an RGB color image \( (I_{RGB}) \) with \( U \times V \) pixels into an image in YCbCr color space \( (I_{YCbCr}) \) as in [68]. Note that \( I_{YCbCr} \) consists of three individual channels which can be represented by \( i_Y, i_{Cb}, \) and \( i_{Cr} \).
Chapter 3 Compressible Image Encryption for EtC Systems with JPEG Compression

Figure 3.8: Examples of images encrypted by using the color-based, conventional grayscale-based, and proposed grayscale-based schemes. \( n \) is the number of divided blocks.

2) Generate a grayscale-based image \( (I_g) \) from \( I_{YCBCr} \). There are two types of \( I_g \) generated from \( I_{YCBCr} \):

(a) \( I_g \) without color sub-sampling: As shown in Fig. 3.10(a), \( i_Y, i_{Cb}, \) and \( i_{Cr} \) are combined into \( I_{YCBCr} \) with \( 3(U \times V) \) pixels.

(b) \( I_g \) with 4:2:0 color sub-sampling: To provide higher compression rate than \( I_g \) without color sub-sampling, as shown in Fig. 3.10(b), \( i_{Cb} \) and \( i_{Cr} \) are subsampled to produce the sub-sampled chrominance \( (i'_{Cb} \) and \( i'_{Cr} \)) using the same method as in [69]. Then, \( i_Y, i'_{Cb}, \) and \( i'_{Cr} \) are combined into \( I_g \) with \( \frac{3}{2}(U \times V) \) pixels.

3) Divide \( I_g \) into blocks each with \( B_x \times B_y \) and then randomly permute the divided blocks by using a random integer generated by a secret key \( K_1 \).

4) Rotate and invert each block randomly by using a random integer generated by a key \( K_2 \).

5) Apply negative-positive transformation to each block by using a random binary integer generated by a key \( K_3 \).
An example of images encrypted by using the proposed grayscale-based scheme is shown in Figs. 3.8(d) and (e).

### 3.3.2 Decryption Procedure

As shown in Fig. 2.1, to reproduce a decrypted image ($\hat{I}$) from an encrypted JPEG image ($\hat{I}_{ec}$), JPEG decompression is performed to $\hat{I}_{ec}$. As a result, the decompressed image ($\hat{I}_e$) is generated. Then, the following steps are carried out to decrypt $\hat{I}_e$ using the corresponding secret key $K_g$ (See Fig. 4.9).

1) Divide $\hat{I}_e$ into blocks, each with $B_x \times B_y$. 

Figure 3.9: Encryption Procedure

Figure 3.10: Grayscale-based image generation methods where the color space transformation is carried out as in [68].
2) Apply inverse negative-positive transformation to each block with key $K_3$.  
3) Inversely rotate and invert each block with key $K_2$.  
4) Assemble blocks based on key $K_1$ to produce the grayscale-based image ($\hat{I}_g$).  
5) Reconstruct the color image in YCbCr color space ($\hat{I}_{YCbCr}$) from $\hat{I}_g$.  
6) Transform $\hat{I}_{YCbCr}$ to RGB color space.  
7) Integrate RGB components to generate $\hat{I}$.  

There are two methods used for reconstructing $\hat{I}_{YCbCr}$ from $\hat{I}_g$ in Step 5:  

- **$I_g$ without sub-sampling**: When $I_g$ was generated according to Fig. 3.10(a), $\hat{I}_g$ is separated into three color channels. Then, the three channels are integrated into $\hat{I}_{YCbCr}$, as shown in Fig. 3.12(a).  
- **$I_g$ with 4:2:0 sub-sampling**: When $I_g$ was generated according to Fig. 3.10(b), $\hat{I}_g$ is separated into one luminance ($\hat{i}_Y$) and two sub-sampled chrominance components ($\hat{i}'_{Cb}$ and $\hat{i}'_{Cr}$), as shown in Fig. 3.12(b). Then, $\hat{i}'_{Cb}$ and $\hat{i}'_{Cr}$ are interpolated to increase the spatial resolution as in [69]. Eventually, the luminance and interpolated chrominance components are integrated into $\hat{I}_{YCbCr}$.  

![Figure 3.11: Decryption procedure](image-url)
3.3.3 Compression of Grayscale-based Image Encryption

The grayscale-based image encryption focuses on JPEG lossy encryption, although the JPEG standard supports both lossy and lossless compressions, and the block scrambling-based image encryption schemes are applicable to lossless compression as discussed in [22]. This is because most JPEG compression applications, especially SNS and CPSS providers, employ lossy compression.

JPEG software, such as Independent JPEG Group (IJG) software [69], generally utilizes two default quantization tables to quantize \( i_Y, i_{Cb}, \) and \( i_{Cr} \) of \( I_{YCbCr} \) called the luminance quantization table (Y-table), and the chrominance quantization table (CbCr-table), as illustrated Fig. 3.13. In addition, according to the JPEG structure, a quantization table is stored in the header of a JPEG file; therefore, JPEG standard
allows us to use a custom table.

Although image-dependent quantization tables can be designed by users [70], grayscale-based images do not correspond to luminance or chrominance. Therefore, a new quantization table called G-table is proposed to improve the compression performance of \( I_g \).

Figure 3.13 shows the summary of quantization tables used for JPEG compression. In JPEG compression, all pixel values in each block of \( I_g \) are mapped from \([0, 255]\) to \([-127, 128]\) by subtracting 128, then each block is transformed using Discrete Cosine Transform (DCT) to obtain DCT coefficients.

The DCT coefficients are employed to generate G-table. Let \( DCT_m(i, j) \) be the DCT coefficient of the \( m \)th block at the position \((i, j)\) where \(1 \leq i \leq 8\) and \(1 \leq j \leq 8\). Considering every block of \( I_g \), the absolute value of \( DCT_m(i, j) \) is calculated, and the arithmetic mean of \( |DCT_m(i, j)| \) is expressed by

\[
c(i, j) = \frac{1}{n_g} \sum_{m=1}^{n_g} |DCT_m(i, j)|
\]  

where \( I_g \) consists of \( n_g \) blocks.

As a set of grayscale-based images which consists of \( N_I \) images is utilized to determine G-table, we define \( c_k(i, j) \) as \( c(i, j) \) of the \( k \)th image and calculate the average of every \( c(i, j) \) from \( N_I \) grayscale-based images. The average \( \bar{c}(i, j) \) is calculated as follow.

\[
\bar{c}(i, j) = \frac{1}{N_I} \sum_{k=1}^{N_I} c_k(i, j)
\]

To obtain G-table, \( q(i, j) \) represents the quantization step size at \((i, j)\) and is derived from the ratio between \( \bar{c}(1, 1) \) and \( \bar{c}(i, j) \). The step size can be calculated by

\[
q(i, j) = \left\lfloor \frac{\bar{c}(1, 1)}{\bar{c}(i, j)} \right\rfloor + \epsilon
\]

where \( \epsilon \) is set to 16 for adjusting the Y-table step size at \((1, 1)\) as for IJG software [69].
To design G-table for grayscale-based images, 1338 images with $512 \times 384$ pixels from Uncompressed Color Image Database (UCID) [71] were employed. Grayscale-based images were generated from all images in the dataset. Then, the grayscale-based images were compressed by using IJG software [69] to obtain the DCT coefficients. To design G-table for $I_g$ without sub-sampling, the DCT coefficients were calculated whereas $n_g = 9216$, $N_I = 1338$, and $\epsilon = 16$. On the other hand, G-table for $I_g$ with 4:2:0 sub-sampling was designed by using $n_g = 4608$. As a result, two types of G-table designed for $I_g$ without sub-sampling and with 4:2:0 sub-sampling were designed as shown in Fig. 3.14.

### 3.4 Experiments and Discussion

#### 3.4.1 Compression Performance

To evaluate the compression performance of the grayscale-based image encryption scheme, we utilized 30 images from CSIQ dataset ($512 \times 512$) [72]. All images in the dataset were encrypted by using the scheme with $B_x = B_y = 8$. Then, all encrypted images were compressed with specific quality factors, $Q_f \in [70, 100]$, using the JPEG standard from IJG software [69]. Rate-distortion (RD) curves, which are the average peak signal-to-noise ratio (PSNR) values of all images per bits per pixel ($\text{bpp}$), were used for evaluating the compression performance. Note that $\text{bpp}$ is the ratio between JPEG file size and the number of pixels, where the number of pixels is the resolution of an original image. For example, $\text{bpp}$ of a grayscale-based encrypted image without sub-sampling is the ratio between JPEG file size of the encrypted one and the number of pixels of the corresponding original color one. Two types of G-table in Fig. 3.14 were utilized to quantize the DCT coefficients of grayscale-based images, while the IJG standard tables were employed for images encrypted by the color-based encryption scheme and non-encrypted ones.

Figure 3.15 shows RD curves of JPEG compressed images without any encryption and with encryption. As shown in Fig. 3.15(a), when images were compressed without sub-sampling, the images encrypted by using the proposed grayscale-based scheme had higher PSNR values than those with the conventional grayscale-based encryption scheme, and moreover, had almost the same RD curves as the non-encrypted ones and the color-based scheme.

For 4:2:0 color sub-sampling, the conventional grayscale-based scheme cannot consider 4:2:0 color sub-sampling because the conventional grayscale-based images consist of RGB color components. As shown in Fig. 3.15(b), when 4:2:0 sub-sampling was carried out, the proposed scheme provided slightly higher PSNR values, compared with
Table 3.3: Parameters used for uploaded JPEG images

<table>
<thead>
<tr>
<th>Type</th>
<th>Color Sub-sampling</th>
<th>Block size</th>
<th>Quantization Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-encrypted</td>
<td>4:4:4 (no sub-sampling) (non-encrypted)</td>
<td>4:2:0</td>
<td>IJG standard table</td>
</tr>
<tr>
<td>Color-based</td>
<td>4:4:4 (no sub-sampling)</td>
<td>16 x 16</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>4:4:4 (no sub-sampling)</td>
<td>8 x 8</td>
<td>G-table (4:4:4)</td>
</tr>
<tr>
<td>Grayscale-based</td>
<td>4:2:0</td>
<td></td>
<td>G-table (4:2:0)</td>
</tr>
</tbody>
</table>

non-encrypted images and the color-based scheme.

In order to clearly compare the difference between the conventional scheme and the proposed one, RD curves in a low bpp range were plotted in Fig. 3.15(c). The result showed that the proposed scheme can provide higher PSNR values than the conventional one. In addition, the proposed scheme with 4:2:0 sub-sampling is useful for the low bpp range. Therefore, the proposed scheme enables us to maintain almost the same image quality as non-encrypted images.

Moreover, other conventional encryption methods are considered in terms of compression performance. There are various encryption methods which can maintain an image format after encrypting as well as the proposed grayscale-based scheme. However, they are not suitable to EtC systems with JPEG compression, because they do not consider using JPEG compression. Figures 3.16(d) and (e) illustrate decrypted images after compressing and decompressing encrypted ones, where Fig. 3.16(a) is the original one. The image quality of decrypted images heavily decreased due to JPEG compression as shown in Figs. 3.16(d) and (e), because they do not consider using JPEG compression as well as most other conventional encryption methods. The proposed scheme can maintain the high quality of images as shown in Fig. 3.16(b) and (c). Note that Fig. 3.16(e) is a grayscale image [26], so the PSNR value is not listed.

3.4.2 Robustness against Recompression

An experiment was carried out for evaluating robustness against image recompression and color sub-sampling forced by providers. In the experiment, CSIQ dataset and a set of parameters for JPEG compression in Table 3.3 were utilized. According to Fig. 2.1,
Chapter 3 Compressible Image Encryption for EtC Systems with JPEG Compression

Figure 3.15: R-D curves of uploaded JPEG images

the following procedure was carried out.

1) Generate encrypted image $I_e$ from an original image $I$.
2) Compress encrypted image $I_e$ with $Q_{fu}$.
3) Upload encrypted JPEG image $I_{ec}$ to SNS providers.
4) Download recompressed JPEG image $\hat{I}_{ec}$ from the providers.
5) Decompress encrypted JPEG image $\hat{I}_{ec}$.
6) Decrypt decompressed image $\hat{I}_e$.
7) Compute the PSNR value between the original image $I$ and decrypted image $\hat{I}$.

To compare PSNR values in 7), original image $I$ was also compressed without any encryption, then uploaded and downloaded. The downloaded images were decompressed, and then, the average PSNR values of 30 images per $Q_{fu}$ were calculated.
Since Twitter and Facebook recompress uploaded JPEG images under their conditions as shown in Table 3.1, encrypted images were uploaded to both SNS providers to evaluate the robustness against image recompression.

Figure 3.17 depicts examples of the procedures for evaluating the robustness against recompression forced by providers where an uploaded image is encrypted by the proposed scheme with 4:2:0 sub-sampling. The encrypted image is robust against color sub-sampling and can avoid the interpolation of the sub-sampled chroma components with discontinuous pixel values, as illustrated in Fig. 2.2.

In the experiment, suppose that the image resolution is less than or equal to the maximum resolution of each provider as well as in [20, 22, 32]. Images encrypted by using the proposed scheme were uploaded to Twitter and Facebook, and then downloaded, as well as images encrypted using color-based scheme [18–22] with $B_x = B_y = 16$, and non-encrypted images to confirm the effectiveness of the proposed grayscale-based scheme.

Figure 3.18 shows the quality of images downloaded from Twitter, where the values

![Decrypted images after compressing and decompressing encrypted ones](Q_f = 90)
Figure 3.17: Examples of encryption and decryption procedures when 4:2:0 color subsampling is carried out in the grayscale-based image generation method. An encrypted image has robustness against color sub-sampling forced by providers; therefore, the decrypted image does not include any block artifacts.

in horizontal axis are $bpp$ values of uploaded images. Although the color-based scheme could maintain almost the same PSNR values as non-encrypted images, the proposed grayscale-based scheme offered slightly higher PSNR values in both sub-sampling ratios, as shown in Fig. 3.18(a) and (b). This is because color JPEG images are affected by 4:2:0 color sub-sampling carried out by Twitter while grayscale-based images can avoid the effect of color sub-sampling.

Figure 3.19 shows the quality of images downloaded from Facebook, where the horizontal axis refers to $bpp$ values of uploaded images. As shown in Fig. 3.19(a), when images were compressed with 4:4:4 sub-sampling, the quality of images encrypted by the color-based scheme were heavily degraded compared with the non-encrypted ones. In comparison, the images encrypted by the proposed grayscale-based scheme provided higher PSNR values than those encrypted by the color-based scheme and non-encrypted images. This is because the proposed scheme is not affected by 4:2:0 sub-sampling carried out by Facebook, although color JPEG images are affected by 4:2:0 sub-sampling. When images were compressed with 4:2:0 sub-sampling, the proposed scheme provided higher image quality than the color-based image encryption, as shown in Fig. 3.19(b). However, when $bpp > 1$, PSNR values of images encrypted by the proposed scheme are slightly lower than non-encrypted ones, but the decrypted images did not include any block artifacts. This is because every grayscale JPEG image uploaded to Facebook is recompressed to the new grayscale JPEG image with $Q_{fd} = 71$ while Facebook recompresses JPEG color JPEG images with $Q_{fd} \in [71, 85]$, as shown in Table 3.1. Note that the PSNR values of images encrypted by the color-based scheme were much lower than those of the proposed scheme and non-encrypted ones. The reason is that the images encrypted by the color-based scheme included block artifacts due to the influence of
Figure 3.18: R-D curves of downloaded JPEG images from Twitter

Consequently, the proposed scheme is shown to have robustness against image re-compression and color sub-sampling caused by providers as the conventional grayscale-based scheme [23, 24]. As shown in Fig. 3.20(b), the decrypted image with the proposed scheme did not include any block artifacts, although 4:2:0 color sub-sampling was carried out in the grayscale-based image generation method.

### 3.4.3 Security against Ciphertext-only Attacks

Since grayscale-based image encryption generates encrypted images that have only one color channel, it provides the higher security level than the color-based image encryption scheme in both brute-force and jigsaw puzzle solver attacks as well as the conventional grayscale-based encryption.
Robustness against Brute-force Attack

In $I_g$ without color sub-sampling, the number of pixels of $I_g$ is the same as the conventional one. This means that the number of blocks is also identical. As a result, the key space of this type is equal to that with the conventional one in (3.11).

In contrast, $I_g$ with 4:2:0 sub-sampling has $\frac{3}{2}(U \times V)$ pixels, so $n_g = \frac{3}{2}n$. As a result, the key space of this type ($N_{P_{sub}}(n)$) is given by

$$N_{P_{sub}}(n) = \frac{3}{2}n! \cdot 8^{\frac{3}{2}n} \cdot 2^{\frac{3}{2}n} \gg N_A(n).$$

(3.18)

The robustness against brute-force attack was evaluated in terms of the key space of encrypted images [73] where the number of pixels of the original images ($N_p$) is $256 \times 144$. The key space of the color-based scheme and the conventional scheme were calculated using 3.9 and 3.11, respectively. On the other hand, the key space of the proposed scheme without sub-sampling and with sub-sampling were calculated by using 3.11 and 3.18 respectively. As shown in Table 3.4, the key space of the proposed scheme without color sub-sampling is equal to that of the conventional one and greatly higher than the color-based scheme. Even if the key space of the proposed scheme with color
Robustness against Jigsaw Puzzle Solver Attacks

The robustness against jigsaw puzzle solver attacks was evaluated in terms of the difficulty of assembling encrypted images.

A. Experimental Conditions

The jigsaw puzzle solver (JPS) was implemented in MATLAB2017a on a PC with a 3.6GHz processor and a main memory 16Gbytes (Processor: Intel Core i7-7700 3.6GHz, OS: Ubuntu 16.04 LTS).

The robustness against JPS was evaluated in terms of $D_c, N_c, and L_c$, as described in Section 3.2.5, where a larger value means higher compatibility.

B. Experimental Results
Table 3.4: Key space of the color-based, conventional, and proposed scheme ($N_p = 256 \times 144$)

<table>
<thead>
<tr>
<th>Encryption type</th>
<th>Number of pixels</th>
<th>$B_x \times B_y$</th>
<th>Key Space</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-based scheme [19, 20]</td>
<td>$N_p$</td>
<td>$16 \times 16$</td>
<td>$144! \cdot 2^{720} \cdot 3^{144}$</td>
<td>3</td>
</tr>
<tr>
<td>Conventional scheme [23]</td>
<td>$3N_p$</td>
<td>$8 \times 8$</td>
<td>$1728! \cdot 2^{6912}$</td>
<td>1</td>
</tr>
<tr>
<td>Grayscale-based scheme without sub-sampling</td>
<td>$3N_p$</td>
<td>$8 \times 8$</td>
<td>$1728! \cdot 2^{6912}$</td>
<td>1</td>
</tr>
<tr>
<td>Grayscale-based scheme with sub-sampling</td>
<td>$\frac{3}{2}N_p$</td>
<td>$8 \times 8$</td>
<td>$864! \cdot 2^{3456}$</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 3.21: Running time of assembling encrypted images from resized Ultra-eye dataset ($256 \times 144$)

Table 3.5 illustrates the robustness against the extended jigsaw puzzle solver attack [34–36]. The scores for assembling both types of the proposed scheme were much lower than those of the color-based scheme and equal to those with conventional one. This is because images encrypted with the proposed scheme have a large number of encrypted blocks and no color information in the blocks.

Figures 3.22(g), (h), (i), (j), and (k) are images assembled from Figs. 3.22(b), (c), (d), (e), and (f), respectively, while Fig. 3.22(a) is the original one. Comparing Fig. 3.22(g) and (h), it obviously shows that the difficulty of assembling encrypted images depends on the block size. Since most conventional jigsaw puzzle solvers employ color information for assembling puzzles, reducing color channels of each pixel makes assembling puzzles harder. As shown in Fig. 3.22(i) and (j), it is more difficult to assemble the im-
Table 3.5: Security evaluation of the color-based, conventional, and proposed scheme against the extended jigsaw puzzle solver [34–36].

<table>
<thead>
<tr>
<th>Encryption type</th>
<th>Color channel</th>
<th>Block size</th>
<th>$D_c$</th>
<th>$N_c$</th>
<th>$L_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color-based scheme [19, 20]</td>
<td>RGB</td>
<td>16 × 16</td>
<td>0.035</td>
<td>0.202</td>
<td>0.223</td>
</tr>
<tr>
<td>Conventional scheme [23]</td>
<td>Grayscale</td>
<td>8 × 8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.002</td>
</tr>
<tr>
<td>Grayscale-based scheme with sub-sampling</td>
<td>Grayscale</td>
<td>8 × 8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.002</td>
</tr>
<tr>
<td>Grayscale-based scheme without sub-sampling</td>
<td>Grayscale</td>
<td>8 × 8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Figure 3.21 shows the running time to assemble encrypted images by using the jigsaw puzzle solver [34–36], where the average time of 20 images from resized Ultra-eye dataset [73] were plotted. We compared the running time to assemble images encrypted with the color-based scheme ($B_x = B_y = 16$), the conventional scheme ($B_x = B_y = 8$) and the proposed one ($B_x = B_y = 8$).

As shown in Fig. 3.21, although the images encrypted by using the proposed grayscale-based scheme ($B_x = B_y = 8$) with 4:2:0 sub-sampling were solved in 10.33 minutes, the scores of assembled images were very low as $L_c = 0.002$ (See Table 3.5). It obviously takes more time to assemble encrypted images than that of images encrypted by the color-based scheme. The reason is that the proposed scheme can offer a smaller block size, the larger number of blocks, and less color information. Moreover, the images encrypted by the proposed method without any sub-sampling were solved in 45.84 minutes which is almost the same as that of conventional one. As a result, the proposed scheme can enhance security against ciphertext-only attacks in terms of both computational complexity and the accuracy of assembled results.

3.5 Summary

In this chapter, a novel grayscale-based block scrambling image encryption scheme using YCbCr color space has been proposed to not only to provide almost the same security level as the conventional grayscale-based encryption scheme, but also to improve the compression performance for EtC systems with JPEG compression. In addition, the proposed grayscale-based scheme allows us to consider the color sub-sampling operation that can improve the compression performance, although the encrypted images have no
The grayscale-based scheme firstly converts an RGB color image into YCbCr color image, and then, generates a grayscale-based image using the proposed grayscale-based image generation method, which consists of two types: grayscale-based image without color sub-sampling and grayscale-based image with color sub-sampling. Then, four encryption steps, including scrambling, inverting, rotating, and negative-positive transformation, are performed to each block to generate an encrypted image with one color channel. In addition, a novel JPEG quantization table, which is especially designed for the proposed grayscale-based image, has been proposed to enhance compression performance of JPEG compression.

In the experiments, the compression performance of the proposed grayscale-based scheme is determined in terms of rate-distortion (RD) curves, which are the average PSNR values of all images per bits per pixel (bpp). The experimental results showed that the proposed scheme has better performance than the conventional one in terms of the compression performance as well as the robustness against image recompression forced by the providers. Furthermore, the proposed scheme was confirmed to provide higher robustness against COA than the conventional color-based image encryption and have almost the same robustness as the conventional grayscale-based encryption.
Figure 3.22: Assembled images by using the extended jigsaw puzzle solver [34–36]
Chapter 4

Learnable Image Encryption for Privacy-Preserving Deep Neural Networks

4.1 Introduction

With the widespread of distributed systems for information processing, such as social networking and cloud computing, multimedia data are not only transmitted but also computed in cloud environments. As cloud providers are not trusted in general, it is necessary for clients to control data security issues such as data privacy, data leakage, and unauthorized data access by themselves. Deep neural networks (DNNs) have greatly contributed to solving complex tasks for many applications [37–39], such as for computer vision, biomedical systems, and information technology. However, there are security issues when using deep learning in cloud environments to train and test data, such as data privacy, data leakage, and unauthorized data access. Therefore, privacy-preserving DNNs have become an urgent challenge.

Even though some perceptual encryption methods [19,20,24,27] can be applied to traditional machine learning (ML) algorithms, such as support vector machine (SVM), k-nearest neighbors (KNN), and random forest, even under the use of the kernel trick [31], these methods have never been applied to DNNs. As described in Chapter 2, a conventional learnable image encryption scheme [28] has been proposed for privacy-preserving DNNs for image classification. Namely, images encrypted by using the conventional method can be applied to train and test DNNs in which an adaption network is added prior to DNNs to avoid the influence of image encryption. However, the accuracy of image classification is lower than that of using plain images. To solve complex tasks, a large amount of data is necessary to train DNNs, and, moreover, data
augmentation in the encrypted domain cannot be applied to the conventional method. Consequently, this chapter focuses on a learnable image encryption scheme with a common security key, which is a pixel-based image encryption method. The pixel-based image encryption method has been proposed to not only to apply images without visual information to DNNs but also to consider data augmentation in the encrypted domain. In addition, a novel adaptation network, which is added prior to DNNs, has been presented to enhance the classification performance of DNNs by obtaining the representations of each pixel before passing through the well-known DNNs. As a result, the pixel-based scheme relaxes the use of image encryption on DNNs in terms of signal processing in the encrypted domain and classification performance. The experimental results showed that the pixel-based image encryption outperforms the conventional encryption methods, including the block-based one for EtC systems, in terms of the classification performance, although data augmentation is carried out in the encrypted domain.

4.2 Related Work

Privacy-preserving machine learning methods with homomorphic encryption (HE) [57, 61–65] have been studied. One is CryptoNet [64], which can apply HE to the influence stage of DNNs. CryptoNet has very high computational complexity, so a dedicated low computer convolution core architecture for CryptoNet was proposed and implemented with CMOS technology [65]. In CryptoNet, all activation functions and the loss function must be polynomial functions. Therefore, it cannot be applied to state-of-the-art DNNs. Moreover, CryptoNet assumes that the weights in a neural network have been trained beforehand; therefore, CryptoNet is not robust against model inversion attacks [49,50].

In comparison, an approach with HE was proposed for privacy-preserving weight transmission for multiple owners who wish to apply a machine learning method over combined data sets [57, 61–63]. In this approach, since the gradients are encrypted by using HE, model information is not leaked. Privacy-preserving weight transmission can provide robustness against model extraction attacks. However, this approach cannot be applied to network training in the encrypted domain.

Alternatively, there is a perceptual image encryption [28] that has been proposed to apply to privacy-preserving DNNs for image classification, but there are several issues that need to be overcome. Details are given in the next section.


4.2.1 Conventional Learnable Image Encryption

According to [28], a conventional learnable image encryption scheme has been proposed for protecting visual information of training and testing images for privacy-preserving DNNs. This scheme is known as Tanaka’s scheme [28], which applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to DNNs.

As depicted in Fig. 4.1, The 8-bit pixel values in $B_x \times B_y$ blocks are separated into upper and lower 4-bit to form the 6-channel blocks. The intensities of pixel values are randomly reversed and shuffled. The 6-channel blocks are reformed to 3-channel blocks.

In addition to encryption steps, the adaptation network, which consists of the first convolution layer ($B_x \times B_y$ kernel and $B_x \times B_y$ stride), several network-in-network style layers and sub-pixel convolution (pixel shuffle), has been proposed to reduce the
influence of image encryption. After the adaptation network, any network can be followed, as illustrated in Fig. 4.2.

However, Tanaka’s scheme cannot avoid the influence of data augmentation in the encrypted domain. As illustrated in Fig. 4.3, the pixel permutation sequence of all encrypted blocks is changed when horizontal flipping is carried out in the encrypted domain. Therefore, the adaptation network is affected due to such situation. In addition, an encrypted image has some visual information from the original image, as shown in Fig. 4.4(c), compared with the conventional pixel-based image encryption [25] in Fig. 4.4(d).

### 4.3 Pixel-based Image Encryption for Privacy-Preserving Deep Neural Networks

This section proposes a pixel-based image encryption for privacy-preserving DNNs which is not only to apply images without visual information to DNNs but also to consider data augmentation in the encrypted domain.

#### 4.3.1 Framework of Privacy-Preserving DNNs

Figure 4.5 illustrates the training frameworks used for image classification used in this chapter.

- **Data augmentation in plain domain:** As shown in Fig. 4.5(a), data augmentation \((f_d(\cdot))\) is first carried out to a set of training images \((T)\) with labels, which consists of \(g\) images, so \(f_d(T)\) is obtained. Then, a client \(u\) encrypts \(f_d(T)\) by using a set of secret keys \((K_T)\) to protect the visual information. \(\text{Enc}(f_d(T))\) with labels is sent to a cloud server to train DNNs. As a result, \(X_T = \text{Enc}(f_d(T))\).
Figure 4.4: Examples of images. (a) Original image ($U \times V = 96 \times 96$). (b) Image encrypted by block-based encryption [19,20] (Block size = $4 \times 4$). (c) Image encrypted by block-based encryption [28] (Block size = $4 \times 4$). (d) Image encrypted by pixel-based image encryption [25].

- **Data augmentation in encrypted domain:** A client $u$ encrypts $T$ with labels to protect the visual information by using $K_T$, as shown in Fig. 4.5(b). Then, $\text{Enc}(T)$ with labels is uploaded to a cloud server. Eventually, data augmentation is carried out to $\text{Enc}(T)$; therefore, $X_T = f_d(\text{Enc}(T))$.

Like Tanaka’s scheme [28], all training and testing images are encrypted by using only one secret key, i.e. $K_{t_1} = K_{t_2} = \ldots = K_{t_q} = K_{q_1} = \ldots = K_{q_h} = K$.

After a DNN model is trained by using encrypted images, the client $u$ encrypts $Q$ by using the corresponded key $K$ and sends $\text{Enc}(Q)$ to a server, as shown in Fig. 4.6. Hence, $X_Q = \text{Enc}(Q)$.

Then, the server solves a classification problem with an image classification model trained in advance and then returns the classification results to the client.

Note that the server has no secret key, so clients are able to control the privacy of images by themselves even when the classification process is done in the server. In addition, as demonstrated later, the pixel-based image encryption method allows us to carry out data augmentation in the encrypted domain.
4.3.2 Pixel-based Image Encryption

Encryption Procedure

To generate an encrypted training image Enc(I_{t_j}) by using K_{t_j} from I_{t_j}, j \in \{1, 2, \ldots, g\}, three steps are carried out, as shown in Fig. 4.7. Note that K_{t_j} = \{K^{t_j}_{NP}, K^{t_j}_{CS}\} consists of a set of secret keys for negative-positive transformation (NP) and a key for color shuffling (CS), which are denoted by K^{t_j}_{NP} and K^{t_j}_{CS}, respectively.

1) Divide a color image I_{t_j} with U \times V pixels into pixels.

2) Individually apply NP to each pixel of the three RGB color channels, \(i^t_{R}, i^t_{G}, \) and \(i^t_{B}\), by using a random binary integer generated by \(K^{t_j}_{NP} = \{K^{t_j}_{R}, K^{t_j}_{G}, K^{t_j}_{B}\}\), which consists of \(K^{t_j}_{R}, K^{t_j}_{G}, \) and \(K^{t_j}_{B}\) used for encrypting \(i^t_{R}, i^t_{G}, \) and \(i^t_{B}\), respectively. In this step, a transformed pixel value of the \(i\)-th pixel, \(p'_c\), is calculated using
Chapter 4 Learnable Image Encryption for Privacy-Preserving Deep Neural Networks

Figure 4.6: Frameworks of model testing for image classification where $X_Q = Enc(Q)$ is applied to trained model.

Figure 4.7: Encryption procedure of pixel-based image encryption

$$p'_c = \begin{cases} 
  p_c & (r(i) = 0) \\
  p_c \oplus (2^L - 1) & (r(i) = 1) 
\end{cases},$$

where $r(i)$ is a random binary integer generated by $K_{NP}^{t_j}$, $c \in \{R, G, B\}$, and $p_c$ is the pixel value of $I_{t_j}$ with $L$ bits per pixel. The value of the occurrence probability $P(r(i)) = 0.5$ is used to invert bits randomly [24].

3) (Optional) Shuffle three color components of each pixel by using an integer randomly selected from six integers generated by a key $K_{CS}^{t_j}$ as shown in Table 3.2.

For generating encrypted test images $Enc(I_{q_l})$, $l \in \{1, 2, \ldots, h\}$, the same encryption steps are carried out as for training images by using $K_{q_l}$.

Images encrypted by using the proposed method are illustrated in Fig. 4.8, where Fig. 4.4(a) is the original image. It is proved that the visual information of the images was protected as well as in Fig. 4.4(d). Moreover, the proposed encryption and a DNN model are independent; therefore, encrypted images can be applied to any DNNs.
Figure 4.8: Examples of images encrypted by proposed encryption method, where Fig. 4.4(a) is original. (a) Image encrypted by NP. (b) Image encrypted by NP and CS.

![Encryption Diagram](image)

Figure 4.9: Decryption procedure of pixel-based image encryption

**Decryption Procedure**

As shown in Fig. 4.9, to reproduce decrypted images $Dec(Enc(I_{t_j}))$ by using $K_t$ from $I_{t_j}$, $j \in \{1, 2, \ldots, g\}$, three decryption steps are carried out as follows.

1) Divide an encrypted image $Enc(I_{t_j})$ with $U \times V$ pixels into pixels

2) Inversely shuffle three color components of each pixel by using an integer randomly selected from six integers generated by a key $K_{CS}^{t_j}$.

3) Individually apply inverse NP to each pixel of the three RGB color channels by using a random binary integer generated by $K_{NP}^{t_j} = \{K_{R}^{t_j}, K_{G}^{t_j}, K_{B}^{t_j}\}$, which consists of $K_{R}^{t_j}$, $K_{G}^{t_j}$, and $K_{B}^{t_j}$ used for decrypting $Enc(i_{t_j}^{R})$, $Enc(i_{t_j}^{G})$, and $Enc(i_{t_j}^{B})$, respectively.

### 4.3.3 Robustness against Brute-force Attack

The robustness of the pixel-based image encryption against brute-force attack is discussed in terms of key space analysis.
Chapter 4 Learnable Image Encryption for Privacy-Preserving Deep Neural Networks

If $I$ with $U \times V$ pixels is divided into pixels, the number of pixels $n$ is given by

$$n = U \times V. \quad (4.2)$$

The key spaces of negative-positive transformation ($N_{NP}$) and color component shuffling ($N_{CS}$) are represented by

$$N_{NP}(n) = 2^{3n}, N_{CS}(n) = (3P_3)^n = 6^n. \quad (4.3)$$

Consequently, the key space of images encrypted by using the encryption scheme, $N(n)$, is represented by the following.

$$N(n) = N_{NP}(n) \cdot N_{CS}(n) = 2^{3n} \cdot 6^n \quad (4.4)$$

In contrast, in Tanaka’s method [28], $I$ with $U \times V$ pixels is divided into blocks each with $4 \times 4$ pixels, and each block is split into upper 4-bit and lower 4-bit images to generate 6-channel image blocks. Then, the intensities of randomly selected pixels are reversed. Eventually, the pixels in each block are shuffled with the same pattern.

The key space of Tanaka’s method [28], $N_{tanaka}$, is given by

$$N_{tanaka} = 96! \cdot 2^{96}. \quad (4.5)$$

$N(n)$ is equal to $N_{tanaka}$ when $n$ is approximately equal to 106.4. Therefore, the proposed encryption has a larger key space than Tanaka’s method if $U \times V$ is more than $11 \times 11$ pixels.

### 4.3.4 Adaptation Network

This chapter presents an adaptation network for DNNs that consists of convolutional layers without pre-trained weights. The adaptation network aims to adapt images encrypted by the pixel-based encryption to be compatible with DNNs. As in [28], the adaptation network is added prior to a DNN model, and the DNN model with the adaptation network is trained as end-to-end learning. Namely, the weights of the DNN model with the adaptation network are updated during the training process. Since the encryption method is a pixel-based one, the adaptation network consists of simple $1 \times 1$-convolutional layers.

Figure 4.10 illustrates the adaptation network where $C^{M_i}_i$ is $i$-th convolutional layer of the adaptation network with kernel size and stride equal to (1,1), and $M_i$ is the number of feature maps of the $i$-th convolutional layer.

In accordance with NP and CS of the pixel-based encryption, there are 48 possible patterns of each pixel due to the encryption scheme, so that each pixel has to be
adjusted before using with DNNs. $C_i^{M_b}$ learns the patterns of each encrypted pixel of the encrypted images and then the feature representations of each encrypted pixel are obtained. As a result, the output of adaptation network can be used with any DNNs.

### 4.3.5 Data Augmentation in the Encrypted Domain

To solve complex tasks, a large amount of data is necessary to train DNNs. Data augmentation aims to enlarge the number of data points used for training and enables us to avoid the overfitting of DNNs. Many data augmentation techniques have already been proposed, e.g., horizontal/vertical flip, random crop, random rotation, cutout, and random erasing [74]. Data augmentation is required to be done in both clients and servers when training DNNs due to the following reasons. First, it is necessary for the servers to enlarge the number of training data if there is not enough training data for model training. The second reason is that data augmentation in servers can reduce the communication cost. However, in general, conventional privacy-preserving methods have to perform data augmentation in clients, namely, data augmentation in servers is not available [28].

In this chapter, some data augmentation techniques are demonstrated to be applied to images encrypted by the proposed method, as shown in Fig 4.5(b). Here, the following well-known techniques are utilized for data augmentation in the encrypted domain:

- **Horizontal flip**: flips original images horizontally. Therefore, the number of original images is doubled by horizontal flipping.

- **Shifting**: shifts the pixel locations of original images by four pixels on both the horizontal and vertical axes. Hence, the number of original images is increased fourfold.
Chapter 4 Learnable Image Encryption for Privacy-Preserving Deep Neural Networks

In general, data augmentation is randomly performed every batch generation during model training. In this thesis, to avoid the randomness of data augmentation, every possible data augmentation technique is carried out. As a result, the number of images is increased eightfold by horizontal flipping and shifting. For example, if \( T \) consists of 50K images, the total number of images of \( f_d(T) \) is 400K images.

4.4 Experiments and Discussions

To confirm that the pixel-based encryption with a common security key for privacy-preserving DNNs is effective, the classification accuracy was evaluated under two data augmentation conditions.

4.4.1 Experimental Conditions

The image database CIFAR10, which contains 32 \( \times \) 32 pixel color images and consists of 50K training images and 10K test images in 10 classes [75], was employed. Two data augmentation techniques (shifting and horizontal flip) were used to enlarge the number of training images for all cases, i.e. both plain images and encrypted ones. Hence, the number of training images was 400K, as described in Section 4.3.5. In addition, there are two types of data augmentation: data augmentation in the plain domain and data augmentation in the encrypted domain, as shown in Fig. 4.5.

The image classification accuracy of encrypted images was evaluated under the use of deep residual networks (ResNet-18) [76, 77], which consist of 18 layers. ResNet-18 models were trained by using stochastic gradient descent (SGD) with momentum for 200 epochs. The learning rate was initially set to 0.1 and was decreased by a factor of 5 at 60, 120, and 160 epochs. The experiments used a weight decay of 0.0005, a momentum of 0.9, and a batch size of 128.

4.4.2 Results

Data Augmentation in the Plain Domain

Table 4.1 shows the classification accuracy when testing DNN models (ResNet-18) with \( Enc(Q) \), where data augmentation was carried out in the plain domain, as illustrated in Fig. 4.5(a). Figure 4.11(b) depicts examples of images encrypted by the pixel-based method used for training and testing privacy-preserving DNNs, where Fig. 4.11(a) is the original ones. The performance of the pixel-based method with adaptation network was compared with three conventional methods [19, 20, 25, 28].
Table 4.1: Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with $Enc(f_d(T))$.

<table>
<thead>
<tr>
<th>Encryption</th>
<th>Data Augmentation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Plain Domain</td>
<td>Encrypted Domain</td>
</tr>
<tr>
<td>Plain Image</td>
<td>95.53</td>
<td></td>
</tr>
<tr>
<td>With Adaptation (NP) [29]</td>
<td>91.14</td>
<td>92.40</td>
</tr>
<tr>
<td>With Adaptation (NP and CS) [29]</td>
<td>89.92</td>
<td>91.43</td>
</tr>
<tr>
<td>Tanaka’s Scheme [28]</td>
<td>85.78</td>
<td>83.29</td>
</tr>
<tr>
<td>Pixel-based [25]</td>
<td>71.74</td>
<td>71.85</td>
</tr>
<tr>
<td>EtC [19, 20]</td>
<td>82.35</td>
<td>73.35</td>
</tr>
</tbody>
</table>

Figure 4.11: Examples of images from CIFAR10 dataset [75] used in the experiments. (a) Plain images (b) Images encrypted by using the pixel-based method (c) Encrypted images after data augmentation in the encrypted domain.

The conventional method for EtC systems [19, 20] and the conventional pixel-based one [25] cannot be applied to privacy-preserving DNNs so that classification performance was degraded due to the influence of image encryption. The pixel-based method with adaptation network [29] was able to maintain a high classification accuracy and offered the highest accuracy among encryption methods.

Data Augmentation in the Encrypted Domain

In Table 4.1, the performance of the pixel-based method is compared with the conventional methods [19, 20, 25, 28], after data augmentation was carried out in the encrypted domain, as illustrated in Fig. 4.11(b). Figure 4.11(c) illustrates examples of images encrypted by the pixel-based method where data augmentation was performed in the encrypted domain.
Chapter 4 Learnable Image Encryption for Privacy-Preserving Deep Neural Networks

Table 4.2: Advantages of the pixel-based image encryption (○: robust, ×: non-robust)

<table>
<thead>
<tr>
<th>Encryption Type</th>
<th>Number of Training Parameters</th>
<th>Augmentation in Encrypted Domain</th>
<th>Classification Performance</th>
<th>Key Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tanaka’s Scheme [28]</td>
<td>Larger</td>
<td>×</td>
<td>Fair</td>
<td>96! · 2^{96}</td>
</tr>
<tr>
<td>Chapter 4 [29]</td>
<td>Smaller</td>
<td>○</td>
<td>High</td>
<td>2^{3n} · 6^n</td>
</tr>
</tbody>
</table>

The pixel-based method with adaptation network [29] provided the highest accuracy in all encryption methods as well as for data augmentation in the plain domain. In addition, data augmentation in the encrypted domain heavily damaged the conventional methods [19, 20, 25, 28]; therefore, the accuracy is lower than that with the pixel-based method.

4.5 Summary

In this chapter, a learnable image encryption scheme with a common security key, which is a pixel-based image encryption method, has been proposed to not only to apply images without visual information to DNNs, but also to consider data augmentation in the encrypted domain. Negative-positive transformation and color shuffling are randomly carried out to each pixel of a full color image to generate an encrypted image. In addition, an adaptation network, which is added prior to DNNs, has been proposed to enhance the classification performance of DNNs by obtaining the representations of each pixel before passing through the conventional DNNs. As a result, the pixel-based scheme relaxes the use of image encryption on DNNs in terms of signal processing in the encrypted domain and classification performance. The advantages of the pixel-based scheme are summarized in Table 4.2.

In the experiments, classification performance was evaluated by applying the pixel-based image method with a common security key to deep residual networks. The experimental results showed that the pixel-based image encryption outperforms the conventional encryption methods, including the block-based one for EtC systems, in terms of the classification performance. In addition, it was confirmed that the proposed pixel-based method allows us to carry out data augmentation in the encrypted domain. As a result, the advantages of DNNs can be fully utilized while the visual information of original images is protected.
Chapter 5

Learnable Image Encryption with Independent Encryption Keys

5.1 Introduction

Recently, data owners utilize cloud servers to compute and process a large amount of data instead of using local servers. This is because cloud environment provides the flexibility and cost-saving computation. However, there are security issues when using deep learning in cloud environments to train and test data, such as data privacy, data leakage, and unauthorized data access.

The conventional learnable image encryption methods [28] have been proposed to directly apply encrypted images for training and testing DNNs, but it cannot avoid the influence of data augmentation in the encrypted domain. Although the pixel-based encryption method [29] allow us to carry out data augmentation in the encrypted domain, training and testing images are encrypted by using only one common security key. Namely, all training and testing images have to be encrypted by using one encryption key. Therefore, it is necessary to safely manage keys, and this method is not very robust against ciphertext-only attacks (COAs).

Thus, this chapter presents a novel privacy-preserving method for DNNs that enables us to not only apply images without visual information to DNNs for both training and testing but to also consider the use of independent encryption keys, which means that all images are encrypted by using different security keys. Therefore, there is no need to manage security keys [30]. In addition, the proposed method allows us to carry out data augmentation in the encrypted domain without any performance degradation. Moreover, it makes it possible for clients to classify plain images even though DNNs are trained by encrypted images. Moreover, using independent encryption keys also enhances the security in terms of the robustness against DNN-based attack called
Unshared Weight Attack (UWA). Several experiments were conducted to confirm the effectiveness of learnable image encryption with independent keys in terms of image classification accuracy and robustness against UWA. Classification performance was evaluated by applying the pixel-based image encryption with independent keys to well-known networks, that is, deep residual networks \[76,77\] and densely connected convolutional networks \[78\], for image classification. Moreover, the robustness against UWA was evaluated and discussed in terms of the visibility of reconstructed images.

5.2 Related Work

There are two learnable encryption methods \[28,29\] that use encrypted images for both training and testing DNN models. One, the first perceptual encryption for privacy-preserving DNNs, is Tanaka's scheme \[28\]. Tanaka's scheme applies encrypted images to DNNs by reducing the influence of image encryption by adding an adaptation network prior to DNNs. The other is a pixel-based encryption method that directly applies encrypted images to DNNs \[29\].

However, Tanaka’s scheme cannot avoid the influence of data augmentation in the encrypted domain, and an encrypted image has some visual information from the original image, as shown in Fig. 4.4(c), compared with the conventional pixel-based image encryption \[25\] in Fig. 4.4(d). Accordingly, the pixel-based encryption method \[29\] carries out data augmentation in the encrypted domain. However, training and testing images are encrypted by using only one common security key, so it is necessary to safely manage the keys, and the pixel-based method with a common key is not very robust against COAs.

Instead of using a common key, this chapter provides a new idea for learnable image encryption by using independent encryption keys for encrypting training and testing images \[30\].

5.3 Pixel-based Image Encryption with Independent Encryption Keys

This section describes the pixel-based image encryption with independent encryption keys \[30\] that allows the use of different encryption keys for privacy-preserving DNNs, namely, all training and testing images are encrypted with different keys.
Figure 5.1: Frameworks of model testing for image classification where trained model is tested by $X_Q = Q$.

5.3.1 Training Framework

As previously described in Chapter 4, there are two training frameworks used for privacy-preserving image classification: data augmentation in the plain domain and data augmentation in the encrypted domain (See Fig. 4.5).

After a DNN model is trained by using encrypted images, the classification results can be returned by using two testing frameworks as follows.

- **Test with encrypted images**: The client $u$ encrypts a set of testing images ($Q$), which includes $h$ testing images, by using $K_Q$ and sends $Enc(Q)$ to a server, as shown in Fig. 4.6. Hence, $X_Q = Enc(Q)$.

- **Test with plain images**: The client $u$ sends $Q$ to a server, as shown in Fig. 5.9. As a result, $X_Q = Q$.

Then, the server solves a classification problem with an image classification model trained in advance, and then returns the classification results to the client.

Note that the server has no secret key, so clients are able to control the privacy of images by themselves even when the classification process is done on the server.

As previously described, this chapter aims to present the use of different encryption keys; therefore, the key conditions are discussed as follows.

- **Same encryption key**: Like the conventional methods [28, 29], all training and testing images are encrypted by using only one secret key, i.e. $K_{t_1} = K_{t_2} = \ldots = K_{t_g} = K_{q_1} = \ldots = K_{q_h} = K$.

- **Different encryption keys**: The different secret keys are independently assigned to training and testing images, i.e. $K_{t_1} \neq K_{t_2} \neq \ldots \neq K_{t_g} \neq K_{q_1} \neq \ldots \neq K_{q_h}$. 
Chapter 5 Learnable Image Encryption with Independent Encryption Keys

Since all clients are able to utilize independent keys for training and testing a model, there is no need to manage the keys. In addition, since the pixel-based encryption method and DNN models are independent, the method is expected to be applicable to any DNNs.

5.3.2 Key Management

In conventional privacy-preserving DNNs, training and testing images are encrypted by using one common security key. This means that all images used for training and testing DNNs have to be encrypted by using the same encryption key. Therefore, clients are required to manage the secret keys used for encrypting training and testing images. Namely, the clients have to share a secret key to other clients in order to use a trained model, and all encrypted images are vulnerable if the key is leaked. As a result, the clients have to not only confidentially store the key in trusted environments but also transmit the key through a trusted channel.

In comparison, images encrypted by different encryption keys can be utilized for training and testing a DNN model, so there is no need to manage the keys for privacy-preserving DNNs. In addition, under the use of different keys, it is more difficult for adversaries to carry out collusion attacks as well as known-plaintext attacks (KPAs).

5.3.3 Requirements of Encrypted Images

Image encryption methods for privacy-preserving DNNs should meet the following requirements.

- **Visual information protection:** to protect an individual, the time, and the location of a taken photograph.

- **Lightweight computation:** to train and test privacy-preserving DNNs with the same computational cost as with plain images.

- **Low damage to DNNs:** to maintain the performance of DNNs as with plain images.

- **Data augmentation in encrypted domain:** to carry out data augmentation on encrypted images.

- **Security:** to provide robustness against COA, such as UWA.
Figure 5.2: Security issue of the learnable image encryption against unshared weight attack (UWA)

5.4 Robustness against Ciphertext-only Attacks

Security mostly refers to protection from adversarial forces. The visual information of encrypted images has to be difficult to reconstruct. Suppose that a cloud server is semi-trusted, so a client encrypts images to protect the visual information before sending the images to the server. Hence, this chapter focuses on robustness against COAs, such as brute-force attacks and UWA.

5.4.1 Brute-force Attack

As discussed in the previous chapter, the pixel-based encryption has a larger key space than Tanaka’s method if $U \times V$ is more than $11 \times 11$ pixels. Since the encryption method used in this chapter is also a pixel-based one, the pixel-based method under the use of different keys has the same robustness against brute-force attack as that with a common security key.

5.4.2 Unshared Weight Attack

As illustrated in Fig. 5.2, if the adversaries have some of encrypted images and the corresponding plain images, they can reconstruct visual information of the plain images from the encrypted ones by using unshared weight attack (UWA). Namely, UWA may be able to reconstruct the visual information of $I$ from $I_e = Enc(I)$. Under the use of different keys, since it is easy for an adversary to prepare encrypted images and the
corresponding plain ones, it is necessary to evaluate the robustness against unshared weight attack (UWA). Therefore, robustness against this attack has to be evaluated.

As shown in Fig. 5.3, a reconstruction model is trained by using $Enc(T)$, and then the training loss is calculated from a set of reconstructed images ($T'$) and $T$. The network model of UWA consists of three $1 \times 1$-locally connected layers ($C_1$, $C_2$, and $C_3$) each with both a kernel size and a stride of (1,1), as shown in Fig. 5.4. A locally connected layer similarly works as a $1 \times 1$-convolution layer, but weights are unshared. As shown in Fig. 5.4, the numbers of filters of $C_1$, $C_2$, and $C_3$ are 8, 32, and 3, respectively. In the testing process, the reconstruction model, which is trained by $Enc(T)$, is utilized to recover encrypted test images ($Enc(Q)$) to obtain the reconstructed test images ($Q'$).

Images encrypted under the use of different keys will be demonstrated to be robust against this attack later.

### 5.5 Experiments and Discussions

To confirm that the pixel-based encryption with different keys is effective, the classification accuracy and robustness against UWA were evaluated under various conditions.
Figure 5.5: Training and testing frameworks with various key conditions. (a) Same encryption key ($K_T = K_Q = K$) (b) Different encryption keys ($K_{t_1} \neq K_{t_2} \neq \ldots \neq K_{t_g} \neq K_{q_1} \neq \ldots \neq K_{q_h}$) (c) Test with plain images where DNN models was trained under the use of different keys.

5.5.1 Image Classification

Experimental Conditions

The image database CIFAR10, which contains 32 × 32 pixel color images and consists of 50K training images and 10K test images in 10 classes [75], was employed. Two data augmentation techniques (shifting and horizontal flip) were used to enlarge the number of training images for all cases, i.e. both plain images and encrypted ones. Hence, the number of training images was 400K, as described in Section 4.3.5.

In the experiments, there are two encryption key conditions: same encryption key and different encryption keys, as described in Section 5.3.1(See Fig. 5.5). Moreover, the proposed pixel-based method enables us to train DNNs by using encrypted images with different keys and then test it with plain images, as illustrated in Fig. 5.5(c).

The image classification accuracy of encrypted images was evaluated under the use of deep residual networks (ResNet-18) [76, 77], which consist of 18 layers, and densely connected convolutional networks (DenseNet) [78].

The models with ResNet-18 were trained by using stochastic gradient descent (SGD) with momentum for 200 epochs. The learning rate was initially set to 0.1 and was decreased by a factor of 5 at 60, 120, and 160 epochs. The experiments with ResNet-18 used a weight decay of 0.0005, a momentum of 0.9, and a batch size of 128. According to [78], the models with DenseNet were trained by using SGD for 300 epochs. The initial learning rate was set to 0.1, and was lowered by 10 times at 150 and 225 epochs. Moreover, The experiments with DenseNet used a weight decay of 0.0001, a momentum
Chapter 5 Learnable Image Encryption with Independent Encryption Keys

of 0.9, and a batch size of 64.
Figure 5.6: Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with $Enc(f_d(T))$. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.
Figure 5.6 shows the classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where data augmentation was carried out in the plain domain, as shown in Fig. 4.5(a). The performance was evaluated under two key conditions: same encryption key, and different encryption keys. The performance of the pixel-based method with different keys was compared with four conventional methods [19,20,25,28,29]. The pixel-based method was able to maintain a high classification performance and provided the highest accuracy in the encryption methods even when the training and testing images were encrypted under the use of the same key. In addition, under the use of different keys, the performances of the conventional methods [19,20,25,28] degraded heavily, although the pixel-based encryption method was able to maintain almost the same accuracy as under the use of the same key. Note that the conventional pixel-based method considers only using the same encryption key [29]. The accuracy of the conventional method under the use of different encryption key was confirmed to be almost the same as that with the same key.

Figure 5.7 shows 64 filters with a size of $3 \times 3$ in the first convolution layer of ResNet-18. From this figure, we can see that each model has filters different from those other models, although the classification accuracies of the models are almost the same.
Figure 5.8: Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with $f_d(Enc(T))$. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.
Table 5.1: Image classification accuracy of the proposed method under various data augmentation techniques

<table>
<thead>
<tr>
<th>Type</th>
<th>Key Condition</th>
<th>Data Augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Plain Domain</td>
</tr>
<tr>
<td>Random Vertical Flip</td>
<td>Plain Image</td>
<td>89.12</td>
</tr>
<tr>
<td></td>
<td>Same</td>
<td>76.97</td>
</tr>
<tr>
<td></td>
<td>Different</td>
<td>72.55</td>
</tr>
<tr>
<td>Random Rotation</td>
<td>Plain Image</td>
<td>92.17</td>
</tr>
<tr>
<td></td>
<td>Same</td>
<td>84.70</td>
</tr>
<tr>
<td></td>
<td>Different</td>
<td>75.77</td>
</tr>
<tr>
<td>Random Erasing [74]</td>
<td>Plain Image</td>
<td>90.67</td>
</tr>
<tr>
<td></td>
<td>Same</td>
<td>82.14</td>
</tr>
<tr>
<td></td>
<td>Different</td>
<td>74.65</td>
</tr>
</tbody>
</table>

**Data Augmentation in Encrypted Domain (ResNet-18)**

In Fig. 5.8, the performance of the pixel-based method is compared with the conventional methods [19, 20, 25, 28, 29] after data augmentation was carried out in the encrypted domain, as shown in Fig. 4.5(b).

The conventional methods [19, 20, 25, 28] were heavily damaged by data augmentation carried out in the encrypted domain even when images were encrypted under the use of the same key. The pixel-based method provided the highest accuracy in all encryption methods as well as for data augmentation in the plain domain. In contrast, it was proved that the pixel-based method was able to maintain the classification performance under the use of both key conditions.

In addition, Table 5.1 demonstrates the classification performance of the pixel-based method under various data augmentation techniques, such as random vertical flipping, random rotation for $-30^\circ$ or $30^\circ$, and random erasing [74]. It was shown that the proposed method is affected by random flipping, random rotation, and random erasing while only horizontal flipping and shifting can be used for data augmentation in the encrypted domain, as illustrated in Fig. 5.8. The influence of data augmentation will be described later.
Figure 5.9: Image classification accuracy when testing DNN models (ResNet-18) with plain images $Q$, where models were trained with encrypted images. Horizontal axis corresponds to conditions of images used for training DNNs. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.
Use of Plain Test Images (ResNet-18)

Figure 5.9 shows that the classification accuracy when plain images were used for testing DNNs, where the models were trained by using encrypted images. The models trained by Tanaka’s scheme were not able to classify due to the damages caused by the encryption methods. In comparison, it was confirmed that the pixel-based method and the conventional one under the use of different keys enabled us to test the DNNs with plain images. In addition, the performance of the pixel-based method was maintained under the use of two data augmentation conditions.

Influence of Data Augmentation

To show the influence of data augmentation in more detail, we trained ResNet-18 by using plain images and encrypted ones under the following data augmentation conditions.

- **Without data augmentation**: Data augmentation was not carried out. The number of training images was 50k.
- **Shifting**: Only shifting was utilized for generating training images. As a result, the number of training images was 200k.
- **Horizontal flipping**: Only horizontal flipping was performed. Hence, the number of training images was 100k.

Figure 5.10 illustrates the influence of data augmentation on classification performance. Compared with the results in Fig. 5.10(a), two augmentation techniques in the encrypted domain were confirmed to improve the performance of the proposed method, respectively, and using both techniques provided better results than in Fig. 5.10, as shown in Figs. 5.6 and 5.8. In addition, the difference of the accuracy between the pixel-based method and plain images was shown to be reduced by using data augmentation techniques.

Classification Accuracy (DenseNet)

To confirm that the pixel-based method and models are independent, DenseNet was assigned as a DNN model for training and testing. The performance was evaluated under the use of two data augmentation conditions: data augmentation in the plain domain, and data augmentation in the encrypted domain.

As shown in Fig. 5.11, the classification accuracy of the pixel-based method had almost the same tendency as when using ResNet-18 as a model. In addition, the pixel-based method with different keys was confirmed to maintain the performance even when
Figure 5.10: Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with following data augmentation conditions: (a) Without data augmentation, (b) Shifting, and (c) Horizontal flipping. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.

data augmentation was carried out in the encrypted domain. Therefore, it was proved that the pixel-based method and models are independent.
5.5.2 Robustness against Unshared Weight Attack

An unshared weight attack (UWA) may be able to reconstruct the visual information of $I$ from $I_e$. Therefore, robustness against UWA in Fig. 5.3 is discussed here.

Experimental Conditions

This experiment employed the STL-10 dataset, which consists of 5K training images and 8K testing images [79], and each image has $96 \times 96$ pixels. Note that data augmentation was not carried out in the experiment.

The network in Fig. 5.4 was trained by using SGD with momentum for 70 epochs, and the mean squared error (MSE), which compared the differences between $T_T$ and $T$, was used as a loss function. The learning rate was initially set to 0.1 and decreased by a factor of 10 at 40 and 60 epochs. The experiment used a weight decay of 0.0005, a momentum of 0.9, and a batch size of 128.

The robustness against UWA was evaluated in terms of the visibility of reconstructed images.

Results

Examples of reconstructed images under the use of same and different encryption keys are shown in Fig. 5.12, where Fig. 4.4(a) is the original image.
Chapter 5 Learnable Image Encryption with Independent Encryption Keys

The visual information was slightly recovered by UWA when the model was trained by using encrypted images with the same key, and the test image was encrypted with the same key, as shown in Fig. 5.12(a), and 5.12(c). In comparison, when the model was trained by using encrypted images with different keys, the reconstructed images had almost no visual information, as shown in Fig. 5.12(b), and 5.12(d).

![Figure 5.12: Examples of reconstructed images. (a) T and Q were encrypted by using NP with same key. (b) T and Q were encrypted by using NP under use of different keys. (c) T and Q were encrypted by using NP and CS with same key. (d) T' and Q were encrypted by using NP and CS under use of different keys.](image)

Table 5.2 shows the average structural similarity (SSIM) values [80] and average peak signal-to-noise ratio (PSNR) ones of 8K testing images, where lower values mean lower visual information. Note that the SSIM value is in the range from 0 to 1. In other words, lower scores indicate that encrypted images are more robust against UWA. From this table, it was confirmed that the use of different keys enhances robustness against UWA.
Table 5.2: Average SSIM and PSNR of 8K reconstructed images.

<table>
<thead>
<tr>
<th>Key Conditions</th>
<th>Encryption</th>
<th>SSIM</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same encryption key</td>
<td>NP</td>
<td>0.1732</td>
<td>10.73</td>
</tr>
<tr>
<td></td>
<td>NP and CS</td>
<td>0.1715</td>
<td>10.72</td>
</tr>
<tr>
<td>Different encryption keys</td>
<td>NP</td>
<td>0.0424</td>
<td>9.50</td>
</tr>
<tr>
<td></td>
<td>NP and CS</td>
<td>0.0425</td>
<td>9.50</td>
</tr>
</tbody>
</table>

5.6 Summary

This chapter addressed a new idea of privacy-preserving DNNs using pixel-based image encryption that protects visual information and considers the use of different encryption keys for training and testing images. Therefore, there is no need to manage encryption keys. In addition, the privacy-preserving scheme for DNNs allows us to train a DNN model with encrypted images and then test it with plain images.

The experimental result demonstrated that the pixel-based method with independent encryption keys is able to maintain the classification performance and provide a higher robustness against unshared weight attack (UWA) than the pixel-based one with a common security key. Moreover, the results confirmed that the pixel-based method with different keys was able to classify plain images as well as encrypted images, even when data augmentation was carried out in the encrypted domain.
Chapter 6

Conclusion

This thesis addressed the problems of image encryption for encryption-then-compression (EtC) systems and privacy-preserving deep neural networks (DNNs). A novel compressible image encryption scheme has been proposed to enhance the compression performance and provide the robustness against image recompression forced by cloud providers. In addition, since image encryption for EtC systems cannot be applied to DNNs, this thesis also proposed a novel learnable image encryption scheme for privacy-preserving DNNs.

The contents described in each chapter and the advantages of the proposed method are summarized as follows.

In chapter 1 and 2, the background and the issues of image encryption for untrusted cloud environments are described and discussed.

Chapter 3 described a novel grayscale-based block scrambling image encryption scheme using YCbCr color space that has been proposed to not only to provide almost the same security level as the conventional grayscale-based encryption scheme, but also to improve the compression performance for EtC systems with JPEG compression. In addition, the proposed scheme allows us to consider the color sub-sampling operation which can improve the compression performance, although the encrypted images have no color information. The grayscale-based scheme firstly converts an RGB color image into YCbCr color image, and then, generates a grayscale-based image using the proposed grayscale-based image generation method, which consists of two types: grayscale-based image without color sub-sampling and grayscale-based image with color sub-sampling. Then, encryption steps are performed to each block to generate an encrypted image. In addition, a novel JPEG quantization table, which is especially designed for the proposed grayscale-based image, has been considered to enhance compression performance of JPEG compression. The experimental results showed the significa-
robustness against image recompression forced by the providers, and robustness against COA.

Chapter 4 focused on a learnable image encryption scheme with a common security key, which is a pixel-based image encryption method, that has been proposed to not only to apply images without visual information to DNNs, but also to consider data augmentation in the encrypted domain. In addition, an adaptation network, which is added prior to DNNs, has been presented to enhance the classification performance of DNNs by obtaining the representations of each pixel before passing through the conventional DNNs. As a result, the proposed scheme relaxes the use of image encryption on DNNs in terms of signal processing in the encrypted domain and classification performance. The experimental results showed that the pixel-based image encryption outperforms the conventional encryption methods, including the block-based one for EtC systems, in terms of the classification performance, although data augmentation is carried out in the encrypted domain.

Chapter 5 addressed a new idea of privacy-preserving DNNs using the pixel-based image encryption that protects visual information and considers the use of independent encryption keys for training and testing images. Namely, all training images and testing images are independently encrypted by using different encryption keys. Therefore, there is no need to manage encryption keys. In addition, the pixel-based image encryption with independent keys allows us to train a DNN model with encrypted images and then test it with plain images. Several experiments were conducted to confirm the effectiveness of learnable image encryption with independent keys in terms of image classification accuracy and robustness against COA. The experimental results demonstrated that the pixel-based method with independent encryption keys can maintain the classification performance and provide higher robustness against COA than the pixel-based one with a common security key. Moreover, the results proved that the pixel-based method with different keys was able to classify plain images as well as encrypted images.

6.1 Future Works

There are two possible future issues in this thesis.

The first issue is the study of an image encryption method that is not only compatible to international compression standards, but also able to be applied to privacy-preserving DNNs. Namely, the image encryption method has to be compressible and learnable while visual information of encrypted images is protected.

The second one is the robustness against adversarial examples, as mentioned in chapter 2. In addition to visual information protection, encrypted images have to be
robust against adversarial examples, which aim to confuse users by misclassifying the results of machine learning. Namely, an encrypted image has to be classified correctly, although imperceptible adversarial perturbation has been added to the encrypted image.
References


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES


List of Figures

1.1 Secure image sharing through cloud providers using EtC systems . . . 5
1.2 Relationship of each chapter . . . . . . . . . . . . . . . . . . . . . 7
2.1 EtC system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.2 Sub-sampling in JPEG encoder and interpolation of chroma components in JPEG decoder . . . . . . . . . . . . . . . . . . . . . . . . . . 13
2.3 Example of the decrypted images with the color-based scheme including block distortion ($B_x = B_y = 8$, $Q_{fu} = 100$, and 4:2:0 sub-sampling) . . . 14
2.4 Security issue of the conventional color-based image encryption . . . . 14
2.5 Framework of privacy-preserving DNNs . . . . . . . . . . . . . . . . . 16
2.6 Examples of images. (a) Original image ($U \times V = 96 \times 96$). (b) Image encrypted by block-based encryption [19, 20] (Block size = $4 \times 4$). (c) Image encrypted by pixel-based image encryption [25]. (d) Image encrypted by grayscale-based image encryption [23] (Block size = $4 \times 4$). (e) Image encrypted by block-based encryption [28] (Block size = $4 \times 4$). 18
3.1 JPEG compression procedure . . . . . . . . . . . . . . . . . . . . . . . 23
3.2 JPEG quantization table . . . . . . . . . . . . . . . . . . . . . . . . . . 24
3.3 JPEG decompression procedure . . . . . . . . . . . . . . . . . . . . . . 24
3.4 Block rotation and inversion . . . . . . . . . . . . . . . . . . . . . . . . 27
3.5 Color-based encryption . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.6 Conventional grayscale-based encryption . . . . . . . . . . . . . . . . . 29
3.7 Conventional grayscale-based image generation . . . . . . . . . . . . . . 30
3.8 Examples of images encrypted by using the color-based, conventional grayscale-based, and proposed grayscale-based schemes. $n$ is the number of divided blocks. . . . . . . . . . . . . . . . . . . 33
3.9 Encryption Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
3.10 Grayscale-based image generation methods where the color space transformation is carried out as in [68]. . . . . . . . . . . . . . . . . . . 34
3.11 Decryption procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.12 Reconstruction of decrypted images . . . . . . . . . . . . . . . . . . . . 36
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.13</td>
<td>Summary of quantization tables used for JPEG compression</td>
<td>36</td>
</tr>
<tr>
<td>3.14</td>
<td>G-table</td>
<td>37</td>
</tr>
<tr>
<td>3.15</td>
<td>R-D curves of uploaded JPEG images</td>
<td>40</td>
</tr>
<tr>
<td>3.16</td>
<td>Decrypted images after compressing and decompressing encrypted ones ($Q_f = 90$)</td>
<td>41</td>
</tr>
<tr>
<td>3.17</td>
<td>Examples of encryption and decryption procedures when 4:2:0 color sub-sampling is carried out in the grayscale-based image generation method. An encrypted image has robustness against color sub-sampling forced by providers; therefore, the decrypted image does not include any block artifacts.</td>
<td>42</td>
</tr>
<tr>
<td>3.18</td>
<td>R-D curves of downloaded JPEG images from Twitter</td>
<td>43</td>
</tr>
<tr>
<td>3.19</td>
<td>R-D curves of downloaded JPEG images from Facebook</td>
<td>44</td>
</tr>
<tr>
<td>3.20</td>
<td>Example of the decrypted images downloaded from Twitter, where images were encrypted with $B_x = B_y = 8$, $Q_{fu} = 100$, and 4:2:0 sub-sampling.</td>
<td>45</td>
</tr>
<tr>
<td>3.21</td>
<td>Running time of assembling encrypted images from resized Ultra-eye dataset ($256 \times 144$)</td>
<td>46</td>
</tr>
<tr>
<td>3.22</td>
<td>Assembled images by using the extended jigsaw puzzle solver [34–36]</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Encryption process of a block by Tanaka’s scheme [28]</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Adaptation network of Tanaka’s scheme [28]</td>
<td>53</td>
</tr>
<tr>
<td>4.3</td>
<td>Issue of the conventional learnable encryption when data augmentation is carried out in the encrypted domain.</td>
<td>54</td>
</tr>
<tr>
<td>4.4</td>
<td>Examples of images. (a) Original image ($U \times V = 96 \times 96$). (b) Image encrypted by block-based encryption [19,20] (Block size = $4 \times 4$). (c) Image encrypted by block-based encryption [28] (Block size = $4 \times 4$). (d) Image encrypted by pixel-based image encryption [25].</td>
<td>55</td>
</tr>
<tr>
<td>4.5</td>
<td>Frameworks of model training for image classification. (a) A DNN model is trained by $\text{Enc}(f_d(T))$. (b) $f_d(\text{Enc}(T))$ is used for training a model.</td>
<td>56</td>
</tr>
<tr>
<td>4.6</td>
<td>Frameworks of model testing for image classification where $X_Q = \text{Enc}(Q)$ is applied to trained model.</td>
<td>57</td>
</tr>
<tr>
<td>4.7</td>
<td>Encryption procedure of pixel-based image encryption.</td>
<td>57</td>
</tr>
<tr>
<td>4.8</td>
<td>Examples of images encrypted by proposed encryption method, where Fig. 4.4(a) is original. (a) Image encrypted by NP. (b) Image encrypted by NP and CS.</td>
<td>58</td>
</tr>
<tr>
<td>4.9</td>
<td>Decryption procedure of pixel-based image encryption.</td>
<td>58</td>
</tr>
<tr>
<td>4.10</td>
<td>Adaptation network.</td>
<td>60</td>
</tr>
<tr>
<td>4.11</td>
<td>Examples of images from CIFAR10 dataset [75] used in the experiments. (a) Plain images (b) Images encrypted by using the pixel-based method (c) Encrypted images after data augmentation in the encrypted domain.</td>
<td>62</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Frameworks of model testing for image classification where trained model is tested by $X_Q = Q$.</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Security issue of the learnable image encryption against unshared weight attack (UWA).</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>Training framework of UWA.</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>Network architecture of reconstruction model. Each box denotes multi-channel feature map produced by each layer. Number of channels is denoted above each box. Feature map resolutions are $U \times V$. Kernel size and stride of locally connected layers are $(1,1)$.</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Training and testing frameworks with various key conditions. (a) Same encryption key ($K_T = K_Q = K$) (b) Different encryption keys ($K_{t_1} \neq K_{t_2} \neq \cdots \neq K_{t_g} \neq K_{q_1} \neq \cdots \neq K_{q_h}$) (c) Test with plain images where DNN models was trained under the use of different keys.</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with $Enc(f_d(T))$. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>64 filters with size of $3 \times 3$ in first convolution layer of ResNet-18. Data augmentation was carried out in plain domain. (a) Model trained by using plain images. (b) Model trained by using encrypted images with NP under the use of different keys. (c) Model trained by using encrypted images with NP and CS under the use of different keys.</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with $f_d(Enc(T))$. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.</td>
<td></td>
</tr>
<tr>
<td>5.9</td>
<td>Image classification accuracy when testing DNN models (ResNet-18) with plain images $Q$, where models were trained with encrypted images. Horizontal axis corresponds to conditions of images used for training DNNs. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.</td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with following data augmentation conditions: (a) Without data augmentation, (b) Shifting, and (c) Horizontal flipping. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images.</td>
<td></td>
</tr>
</tbody>
</table>
5.11 Image classification accuracy when testing DNN models (DenseNet) with $Enc(Q)$, where depth and growth rate of DenseNet were 40 and 12, respectively. Note that same key and different keys correspond to the encryption key conditions used for encrypting training and testing images. 80

5.12 Examples of reconstructed images. (a) $T$ and $Q$ were encrypted by using NP with same key. (b) $T$ and $Q$ were encrypted by using NP under use of different keys. (c) $T$ and $Q$ were encrypted by using NP and CS with same key. (d) $T$ and $Q$ were encrypted by using NP and CS under use of different keys. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
## List of Tables

2.1 Contributions of the proposed methods. ◊ indicates that it is superior to the conventional method, and ◯ indicates that it is equivalent to the conventional method. Note that an input image is an RGB full color image. .......................................................... 19

3.1 Relationship between uploaded JPEG files and downloaded ones in terms of sub-sampling ratios and the maximum resolutions. Providers do not resize uploaded images when their resolutions are less than or equal to the maximum resolutions, e.g. the maximum resolutions of Twitter and Tumblr are $4096 \times 4096$ and $1280 \times 1280$, respectively [29]. .......... 26

3.2 Permutation of color components for a random integer. For example, if the random integer is equal to 2, the red component is replaced by the green one, and the green component is replaced by the red one while the blue component is not replaced. .................................................. 28

3.3 Parameters used for uploaded JPEG images ................................................. 39

3.4 Key space of the color-based, conventional, and proposed scheme ($N_p = 256 \times 144$) .................................................................................................................. 46

3.5 Security evaluation of the color-based, conventional, and proposed scheme against the extended jigsaw puzzle solver [34–36]. .................. 47

4.1 Image classification accuracy when testing DNN models (ResNet-18) with $Enc(Q)$, where models were trained with $Enc(f_d(T))$. ........ 62

4.2 Advantages of the pixel-based image encryption (◊: robust, ×: non-robust) .................................................................................................................. 63

5.1 Image classification accuracy of the proposed method under various data augmentation techniques ................................................. 76

5.2 Average SSIM and PSNR of 8K reconstructed images. ......................... 82
Acknowledgements

I wish to thank many people who have helped to complete this thesis, and it would not have been possible to do without the support and guidance.

First of all, I gratefully acknowledge Tokyo Human Resources Fund for City Diplomacy received towards my PhD from Tokyo Metropolitan Government. Thanks to my supervisor, Prof. Hitoshi Kiya, for his guidance and advice through each stage of the thesis. I would like to offer my special thanks to Asst. Prof. Sayaka Shiota for helping and supporting the author during three years of PhD period.

Furthermore, I would like to acknowledge all co-authors of my publications for precious advice and kind supports.

In addition, I would like to express my gratitude to Prof. Nobutaka Ono, and Prof. Masaaki Fujiyoshi for valuable advice and guidance though the review of this thesis.

I would like to express my gratefulness to Assoc. Prof. Masayuki Tanaka from Tokyo Institute of Technology for kind advice through the review of this thesis

Special thanks also to laboratory members who have been good counselors and helped the author during PhD period in Japan. Finally, I owe my deepest gratitude to my family and partner for their warm support.