Deep Joint Encryption and Source-Channel Coding: An Image Privacy Protection Approach

Jialong Xu, Student Member, IEEE, Bo Ai, Senior Member, IEEE, Wei Chen, Senior Member, IEEE, Ning Wang, Member, IEEE, and Miguel Rodrigues, Senior Member, IEEE

Abstract—Joint source and channel coding (JSCC) has achieved great success due to the introduction of deep learning. Compared with traditional separate source channel coding (SSCC) schemes, the advantages of DL based JSCC (DJSCC) include high spectrum efficiency, high reconstruction quality, and the relief of “cliff effect”. However, it is difficult to couple encryption-decryption mechanisms with DJSCC in contrast with traditional SSCC schemes, which hinders the practical usage of the emerging technology. To this end, our paper proposes a novel method called DL based joint encryption and source-channel coding (DJEJSCC) for images that can successfully protect the visual information of the plain image without significantly sacrificing image reconstruction performance. The idea of the design is using a neural network to conduct image encryption, which converts the plain image to a visually protected one with the consideration of its interaction with DJSCC. During the training stage, the proposed DJEJSCC method learns: 1) deep neural networks for image encryption and image decryption, and 2) an effective DJSCC network for image transmission in encrypted domain. Compared with the perceptual image encryption methods with DJSCC transmission, the DJEJSCC method achieves much better reconstruction performance and is more robust to ciphertext-only attacks.

Index Terms—Image encryption, image transformation, privacy protection, security, joint source-channel coding, deep learning.

I. INTRODUCTION

The modular design principle based on Shannon’s separation theorem [1] is the cornerstone of modern communications and has enjoyed great success in the development of wireless communications. However, the assumptions of unlimited codeword length, delay and complexity in the separation theorem are not possible in real wireless environments, leading to sub-optimal separate source channel coding (SSCC). Moreover, for time-varying channels, when the channel quality is lower than the target quality, the separation coding cannot further improve the reconstruction quality. This is the famous “cliff effect” [2], which increases the cost of SSCC during the wireless transmission. In the past years, joint source-channel coding (JSCC) has been demonstrated theoretically to have better error exponent than SSCC in discrete memoryless source channels [3]–[6] motivating the development of various JSCC designs over the years [7]–[10] are proposed.

More recently, deep learning (DL) based approaches have been proposed for source coding [11]–[14], channel coding [15]–[18], and JSCC [19]–[22]. Compared with the SSCC scheme (e.g., JPEG/JPEG2000 for image coding and LDPC for channel coding), the DL based JSCC (DJSCC) scheme designed in [19] has better image restoration quality especially in the low signal-to-noise ratio (SNR) regime. To well adapt variable bandwidth and exploit the utility of channel output feedback in real wireless environments, the schemes of DJSCC with adaptive-bandwidth image transmission and image transmission with channel output feedback are proposed by [20] and [21], respectively. However, all the aforementioned schemes are trained and inferred at the same channel conditions (the single SNR) to ensure optimality, demanding the use of multiple trained networks to suit a range of SNR that leads to considerable storage requirements in transceivers. To overcome this challenging problem in DJSCC, [22] proposed a single network for DJSCC which can adapt to a wide range of SNR conditions to meet the memory limit of the device in real wireless scenarios. So far, by using the data-driven approach, DJSCC successfully reduces the difficulty of coding design in traditional JSCC, and balances the performance and the storage requirement. These benefits brought by DL make DJSCC methods easier to be employed and deployed than traditional JSCC schemes.

Yet, to protect information privacy and confidentiality, one must also couple encryption and decryption mechanisms with DJSCC based wireless communication systems as illustrated in Fig. 1. The image owner intends to transmit a plain image (an unencrypted image) to the image recipient through the network containing a untrusted wired network and a wireless transmission service. To protect visual information of the plain image, the image owner encrypts the plain image into an encrypted image (the visually protected image) before providing it to the wireless service provider. Then the encrypted image is transmitted by the wireless service provider through the DJSCC transmission. After the DJSCC wireless transmission, the corrupted encrypted image (the decoded image by DJSCC...
decoder) is transmitted to the image recipient through the untrusted network. The image recipient decrypts the corrupted encrypted image to the plain image. Even if the encrypted image or the decoded image is leaked or stolen during the wired network transmission process, visual information of the plain image can not be acquired directly.

In SSCC, there are two strategies to safeguard image privacy: 1) encrypting the data encoded by source encoder before the channel encoder as shown in Fig. 2(a), and 2) encrypting the image source before the source encoder in SSCC. The first strategy regarded as compression-then-encryption (CtE) applies the source encoder to compress the image source to the binary data, and then uses some encryption method (e.g., data encryption standard [24], advanced encryption standard [25], and Rivest–Shamir–Adleman [26]) to encrypt the binary data generating the ciphertext. The second strategy called encryption-then-compression (EtC) is fit for the typical scenario, where the image provider only takes care of protecting the image privacy and the telecommunications provider has an overriding interest on improving spectrum efficiency. Following the EtC strategy, various perceptual image encryption methods have been developed via transforming an original image to a visually protected image [23], [27]–[48].

To protect image privacy for DJSCC transmission, the encryption module should be placed in front of deep joint source-channel encoder as shown in Fig. 2(c), which is akin to the position of encryption module in Fig. 2(b). However, a major issue is that EtC change the visual structure of the plain image, causing DJSCC transmission degradation. To the best of our knowledge, there are no works addressing the encryption problem in the DJSCC framework. In [48]–[52], different encryption methods are developed for DNN-based classification tasks. In addition, Homomorphic Encryption (HE) [53] is a promising privacy preserving computation method. Some of the HE-based methods have been applied in DL domain [54]–[57]. However, the high computational complexity, the loss of calculation accuracy due to the use of polynomial instead of nonlinear activation function, and the large ciphertext expansion when combining HE with DNNs is an obstacle to the use of HE in DJSCC.

In this paper, we design a DL based joint encryption and source-channel coding method that can generate a visually protected image suitable for DJSCC transmission with high reconstruction performance. The inspiration of our proposed method originates from image transformation illustrated in Fig. 3. Image transformation has been successfully applied in image compression (e.g., JPEG [58], JPEG2000 [59] and
image processing (e.g., image classification, image semantic segmentation [60]) for facilitating the subsequent operations. Although the transformed image contains the information of the plain image, little visual information can be perceived due to the transformation. Compared with the existing perceptual image encryption methods, the proposed method not only protects the image privacy, but also has a better reconstruction performance. Another advantage of our proposed method is that the proposed method is more robust against the ciphertext-only attacks than the existing perceptual image encryption methods. Moreover, the full convolutional architecture of the proposed method makes it more flexible to deal with images of different sizes without loss of the visual protection performance.

The rest of this paper is organized as follows. Section II presents related work on deep joint source-channel coding, perceptual image encryption, and image transformation. Then, the proposed method is presented in Section III. In Section IV, the proposed method is evaluated on datasets with low resolution and high resolution, respectively. Visual security of the proposed method is evaluated by ciphertext only attacks in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

A. Deep Joint Source Channel Coding

As shown in the lower part of Fig. 1, DJSCC follows the end-to-end autoencoder architecture [61] which replaces the source encoder and the channel encoder (with/without the modulation) with a deep joint source-channel encoder (DJSCCE) in the transmitter, and similarly, replaces the corresponding source decoder, the channel decoder (with/without demodulation) by a deep joint source-channel decoder (DJSCCD).

The initial DJSCC work proposed a recurrent neural network (RNN) based DJSCCE and DJSCCD for text transmission over binary erasure channels [62]. From then on, DJSCC attracted increasing interests, especially for image compression and transmission. In [19], fully convolutional neural networks (FCNNs) are used for DJSCCE and DJSCCD, and are shown to outperform SSCC method especially in the low SNR regime. Furthermore, the DJSCC method in [19] provides a graceful performance degradation in communications scenarios suffering from large channel estimation errors, associated with the well-known “cliff effect” in SSCC.

In the wireless communication systems, the transmission bandwidth is always dynamically allocated according to user requirements and system overload. DJSCC-l proposed by [20] can transmit progressively in layers; When the available bandwidth is limited, codewords of the first few layers are transmitted to the receiver to reconstruct the transmitted image with lower quality; When the available bandwidth is increased, codewords of the residual layers are transmitted to the receiver and are combined with the first codewords to reconstruct the transmitted image with higher quality.

Once channel output feedback is available, DJSCC-f proposed by [21] can further improve the reconstruction quality by exploiting the channel output feedback. The difference between DJSCC-l and DJSCC-f is that DJSCC-f not only aggregates the layered transmission information at the receiver, but also jointly processes the transmitted image and the processed channel output feedback at the transmitter. Compared with DJSCC without channel output feedback, DJSCC with both noisy and noiseless channel output feedback can achieve considerable performance gains. However, [19]–[21] must train multiple networks in a range of SNR and select one of them depending on the real SNR condition to keep their optimality, which would lead to heavy burden on the storage overhead of devices.

Inspired by the resource assignment strategy in traditional JSCE, [22] introduces the SNR feedback to DJSCC and proposes a general DJSCC method named Attention DL based JSCE (ADJSCC). ADJSCC can dramatically reduce the storage overhead while maintaining the similar performance by using attention mechanisms. In addition, [63] proposed a DJSCC based on maximizing the mutual information between the source and noisy received codeword for the binary erasure channel and the binary symmetric channel. [64] and [65] model their DJSCC systems via variational autoencoder and manifold variational autoencoders for a Gaussian source, respectively. However, all of the aforementioned work are not designed to guarantee information privacy protection.

B. Perceptual Image Encryption

Perceptual image encryption aims to transform a plain image into an encrypted image, which downgrades visual quality to protect the original perceptual information.

One can change image visual information along two dimensions: the space dimension and the pixel dimension. The space based encryption involves re-arranging (scrambling) the different pixels within the image. There are numerous image scrambling algorithms, e.g., Arnold [27], Baker Transformation [28], Fibonacci Transformation [29], Magic Square [30], RGB Scramble [31], Chaos Scramble [32], and SCAN Pattern [33]. The pixel based encryption only changes pixel values to protect visual information. Popular methods include DES [34], Hill algorithm [35], chaotic logistic map [36], and cellular automata [37] based methods. Combining space based encryption and pixel based encryption can make encryption more robust. Well-known techniques include Arnold and Chen’s chaotic system based method [38], compound chaotic sequence based method [39], and 3D chaotic map based method [40]. However, the aforementioned encryption methods do not take into account the effect of subsequent operations such as source or channel coding.
Considering the compression task in EIC system, stream cipher methods using a pseudo-random key generator followed by Slepian-Wolf coding and resolution progressive compression have been used for lossless compression in [42], [43]. To further improve the compression ratio, pixel based image encryption followed by a lossy scalable compression technique [44], image encryption via prediction error clustering and random permutation followed by a context-adaptive arithmetic coding [45], pseudorandom permutation based image encryption followed by orthogonal transformation based lossy compression [46], and block scrambling based image encryption followed by JPEG lossy compression [23] are proposed for lossy compression. However, when the final signal processing task is changed from image reconstruction to e.g. image classification, the methods designed for EIC systems stop working.

DL has led to state-of-the-art performance in various image processing tasks, motivating more recently the application of deep learning techniques to encrypted images. The methods proposed by [48], [52] are designed for DL based classification. Tanaka’s method [48] is a hybrid encryption method, adding an adaptation network prior to DNNs. The pixel based image encryption methods with/without key management [49], [50] can directly be fed into DNNs for classification. DL based encryption methods, e.g., generative adversarial networks (GAN) and DNNs, are proposed for image encryption in [51] and [52], respectively. Both of [51] and [52] employ image transformation—transforms the image from one domain to the other domain—to protect visual information. However, the image encryption methods designed for DL based classification are still not fit for DJSCC. The encryption images for classification only reserve some specific semantic information relevant to image class while the pixel based information of the image is discarded, which cause the performance degradation for the image reconstruction task, i.e., DJSCC.

III. DEEP JOINT ENCRYPTION AND SOURCE-CHANNEL CODING

Our method motivation is to successfully protect the visual information of the plain image without significantly sacrificing image reconstruction performance in DJSCC. Conventional image transformations, e.g., discrete cosine transform (DCT) and discrete wavelet transform (DWT), can provide the convenience for subsequent operations and simultaneously generate a visually protected image in the transformation. However, [60], [66], [67] demonstrate that conventional image transformations are hardly matched with the DNNs if the structures of DNNs are not modified. In this Section, a DL based joint encryption and source-channel coding (DJSCC) method is proposed for protecting visual information of plain images in DJSCC transmission.

A. System Model

Considering a visually protected DJSCC transmission system as shown in the lower part of Fig. 2, a plain image is represented by \( x \in \mathbb{R}^n \), where \( \mathbb{R} \) denotes the set of real numbers and \( n = h \times w \times c \). \( h, w, c \) denote the width, height, and the number of channels of an image, respectively. The encryption network transforms the plain image into a visually protected image. This encryption process can be expressed as:

\[
y = e_\mu(x) \in \mathbb{R}^n,
\]

where \( e_\mu(\cdot) \) represents an encryption deep neural network based parameterized by the set of parameters \( \mu \). Note that the plain image \( x \) and the visually protected image \( y \) have the same size.

Then the visually protected image \( y \) is encoded by the joint source-channel encoder as:

\[
z = f_\theta(y) \in \mathbb{C}^k,
\]

where \( \mathbb{C} \) denotes the set of complex numbers, \( k \) represents the size of channel input symbols, and \( f_\theta(\cdot) \) represents an joint source-channel encoder parameterized by the set of parameters \( \theta \). The encoded complex-valued \( z \) represents the transmitted signals at the transmitter. The real parts and the imaginary parts of \( z \) are considered as in-phase components I and quadrature components Q of the transmitted signals, respectively. Due to the average power constraint at the transmitter, \( \frac{1}{k}\mathbb{E}(zz^*) \leq 1 \) must be satisfied, where \( z^* \) is the complex conjugate transpose of \( z \).

The transmitted signals are corrupted by the wireless channel. We adopt a well known AWGN model given by:

\[
\hat{z} = \eta(z) = z + \omega,
\]

where \( \hat{z} \in \mathbb{C}^k \) is the channel output and \( \omega \in \mathbb{C}^k \) denotes the additive noise modeled by \( \omega \sim \mathbb{CN}(0, \sigma^2 I) \) where \( \sigma^2 \) represents the average noise power and \( \mathbb{CN}(\cdot, \cdot) \) denotes a circularly symmetric complex Gaussian distribution.

In turn, the channel output symbols \( \hat{z} \) is decoded by the joint source-channel decoder as:

\[
\hat{y} = g_\phi(\hat{z}) \in \mathbb{R}^n,
\]

where \( g_\phi(\cdot) \) represents a joint source-channel decoder parameterized by the set of parameters \( \phi \). The decoded image \( \hat{y} \in \mathbb{R}^n \) is with the same size of the visually protected image \( y \).

Similarly to the encryption step, the decryption network is employed to convert the decoded image to the decrypted image, as follows:

\[
\hat{x} = d_\nu(\hat{y}) \in \mathbb{R}^n,
\]

where \( d_\nu(\cdot) \) represents a decryption deep neural network parameterized by the set of parameters \( \nu \) and the decrypted image \( \hat{x} \in \mathbb{R}^n \) is an estimation of the plain image. The bandwidth ratio \( R \) is defined as \( k/n \), where \( n \) is the source size (i.e., image size) and \( k \) is the channel bandwidth (i.e., channel input size).
B. The Proposed Method

In sharp contrast with DJSCC methods [19]–[22], we require our DJESCC method to address two issues:

1) To hide visual information of a plain image.
2) To extract effective features from the visually protected image for subsequent DJSCC transmission.

The classical full-reference metric of image similarity is peak signal-to-noise ratio (PSNR) between the original image and the restored image, which is defined as:

$$\text{PSNR} = 10 \log_{10} \frac{\text{MAX}^2}{\text{MSE}} \text{(dB)}.$$  \hspace{1cm} (6)

where MAX is the maximum possible value of the image pixels and MSE is the abbreviation of mean square error between the original image and the restored image. Although the prediction of PSNR performance is not always consistent with quality perception by the human visual system, the simplicity and the inexpensive computational complexity make it widely used in the field of image processing [68].

However, as is illustrated in Fig. 5, PSNR is not a good metric to assess the effect of hiding visual information. Fig. 5(b) has a lower PSNR than Fig. 5(c), while the visual information (e.g., the birds and the leaf) are more easily identified in Fig. 5(b) than in Fig. 5(c). To solve this problem, a feature extraction network is employed to measure the effect of hiding visual information. The feature extraction method was initially used to measure the similarity between two images [69] and then successfully used to measure the difference between two images [52].

In the training stage, features of the plain image \(x\), the encrypted image \(y\), and the decoded image \(\hat{y}\) are extracted by the feature extraction network \(h_y\), in Fig. 4. The feature loss \(\mathcal{L}_e\) between the plain image \(x\) and the encrypted image \(y\) is expressed as:

$$\mathcal{L}_e = \frac{1}{m} \| h_y(x) - h_y(y) \|_2^2,$$  \hspace{1cm} (7)

where \(h_y(x) \in \mathbb{R}^m\), \(h_y(x) \in \mathbb{R}^m\), and \(h_y(y) \in \mathbb{R}^m\) are the features of the plain image \(x\), the encrypted image \(y\), and the decoded image \(\hat{y}\) extracted by the feature extraction network and \(m = h_f \times w_f \times c_f\), \(h_f\), \(w_f\), and \(c_f\) denote the width, height, and the number of channels of an extracted feature, respectively. The reconstruction loss \(\mathcal{L}_{r}\) between the plain image \(x\) and the decrypted image \(\hat{x}\) is expressed as:

$$\mathcal{L}_r = d(x, \hat{x}) = \frac{1}{n} \| x - \hat{x} \|_2^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2,$$ \hspace{1cm} (9)

where \(x_i\) and \(\hat{x}_i\) represents the intensity values of the plain image \(x\) and the decrypted image \(\hat{x}\), respectively.

Different from the image-to-image translation tasks [51], [69] which minimizes the feature loss in the training stage, the proposed method maximizes \(\mathcal{L}_e\) and \(\mathcal{L}_d\) to hide visual information for the encrypted image \(x\) and the decoded image \(\hat{y}\). The total loss used to train the proposed method:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_r - \lambda_e \mathcal{L}_e - \lambda_d \mathcal{L}_d,$$ \hspace{1cm} (10)

where \(\lambda_e \in \mathbb{R}^+\) and \(\lambda_d \in \mathbb{R}^+\) are the weights of \(\mathcal{L}_e\) and \(\mathcal{L}_d\), respectively. Under a certain bandwidth ratio \(R\), DJESCC learns the parameters of the encryption network \(\mu\), the deep joint source-channel encoder \(\theta\), the joint source-channel de-
coder \( \phi \), and the decryption network \( \nu \) by minimizing the end-to-end distortion as follows:

\[
(\mu^*, \theta^*, \phi^*, \nu^*) = \arg \min_{\mu, \theta, \phi, \nu} \mathbb{E}_{p(\sigma_\nu)} \mathbb{E}_{p(x, \hat{x})}(L_{\text{total}}),
\]

where \( \mu^*, \theta^*, \phi^*, \nu^* \) are the optimal parameters, \( p(x, \hat{x}) \) represents the joint probability distribution of the plain image \( x \) and the decrypted image \( \hat{x} \), and \( p(\sigma_\nu) \) represents the probability distribution of the average noise power \( \sigma_\nu \). Note that the probability distribution of the average noise power instead the fixed average noise power is adopted due to the considerations of the storage overhead and the difficulty to acquire the average noise power in the image owner/recipient. In addition, empirical average instead of statistic average is adopted in the training stage. During the training stage, the proposed DJESCC method learns: 1) an effective method to hide visual information of the plain image, 2) an easy to be extracted image domain for the subsequent DJSCC transmission, 3) an effective DJSCC transmission method, and 4) an effective method to reconstruct the plain image.

After the DJESCC network has been trained, the encryption network \( \epsilon_\mu \) and the decryption network \( \epsilon_\nu \) are securely distributed to the image owner and the image recipient by some security protocol, eg., Secure Sockets Layer (SSH) protocol, respectively. The deep joint source-channel encoder and the deep source-channel decoder are distributed to the DJSCC transmission service provider.

In the test stage, the plain image \( x \) is first converted to the encrypted image \( y \) by the image owner using Eq. (1). Then the encrypted image \( y \) is sent to the DJSCC transmission service provider, DJSCC transmission is executed using Eq. (2) and Eq. (4) and the decoded image \( y \) is obtained. The DJSCC transmission service provider sends the decoded image \( \hat{y} \) to the image recipient, which uses Eq. (5) to decrypt the decoded image \( \hat{y} \). The process of test stage is illustrated in the lower part of Fig. 4.

IV. EXPERIMENTAL RESULTS

Considering the generality of the DJESCC method, there are multiple architecture choices for the encryption network, the DJSCC encoder, the DJSCC decoder, and the decryption network. To prove the potential of our proposed method, the original DJSCC network architecture in [19] is chosen in the subsequent experiments. It is worthy noting that our proposed method can be applied to other extensions of the original DJSCC network architecture.

The DJSCC architecture proposed by [19] is shown in Fig. 6 where the encoder and decoder networks are adopted in our DJESCC method. The DJSCC encoder consists of the normalization layer, five alternant convolutional layers and PReLU layers, the reshape layer, and the power normalization layer. The DJSCC decoder consists of the reshape layer, five alternant transposed convolutional layers and activation layers (ie., four PReLU layers and one sigmoid layer), and the denormalization layer. The normalization layer converts the input image with the pixel value range \([0, 255]\) to the image with pixel value range \([0, 1]\), and the denormalization layer performs the opposite operation. The notation \( F \times F \times K|S \) in a convolution/transpose convolution layer denotes that it has \( K \) filters with size \( F \) and stride down/up \( S \). The power normalization layer is used to satisfy the average power constraint at the transmitter. The channel number of the last convolutional layer in the DJSCC encoder is \( t \). The bandwidth ratio of the proposed method is \( R = t/96 \). The architecture shown in Fig. 7 is employed as the architectures of the encryption network and the decryption network, which is a shallow version of U-Net [70]. In the training stage, the parameters of the feature extraction network is fixed.

Tensorflow [71] and its high-level API Keras is used to implement the proposed DJESCC method. The proposed method is trained under a uniform distribution within the SNR range \([0, 20] \) dB. The following experiments run on a Linux server with twelve octa-core Intel(R) Xeon(R) Silver 4110 CPUs and sixteen GTX 1080Ti GPU. Each experiment was assigned six CPU cores and a GPU.

We first consider the performance of the proposed method on CIFAR-10 dataset, which consists of 60000 \( 32 \times 32 \times 3 \) color images associated with 10 classes where each class has 6000 images. Note the goal of our proposed method is to generate visually protected images for the untrusted transmission channels and reconstruct the plain image at the receiver, so the class label of each image is not used in the following experiments. Training dataset and test dataset contain 50000 images and 10000 images, respectively. We use a part of the VGG16 [72] with batch normalization, i.e., a classical network for classification tasks, as the feature extraction network, which is pretrained on the CIFAR-10. All of the networks were trained for 500 epochs by using Adam optimizer with an initial learning rate of \( 10^{-3} \). Once learning stagnated for 10 epochs, the learning rate was reduced by a factor of 10. The performance of the DJESCC networks were evaluated at specific \( \text{SNR}_{\text{test}} \in [0,20] \) dB on CIFAR-10 test dataset. To alleviate the effect of the randomness caused by the wireless channel, each image in CIFAR-10 test dataset is transmitted 10 times. PSNR is used in the evaluation of the reconstruction performance between the plain image and the decrypted image.

Fig. 8 compares the reconstruction performance of the proposed method with different loss weights (e.g., \( \lambda_c = \lambda_d = 0.005, 0.05, 0.5 \)) at bandwidth ratio \( R = 1/6 \). The reconstruction performance of the DJSCC without visual protection, i.e., \( \lambda_c = \lambda_d = 0 \) is also plotted. The reconstruction performance is gradually decreased with the increase of \( \lambda_c \) and \( \lambda_d \), as the training promotes protection of visual information instead of reconstruction of the images. Fig. 9 shows the visualization of the plain image, the encrypted images transformed by the image owner, the decoded images decoded by the DJSCC transmission service provider, and the decrypted images transformed by the image recipient for \( \text{SNR} = 0, 10, 20 \) dB with different loss weights \( \lambda_c \) and \( \lambda_d \). The plain image comes from CIFAR-10 test dataset. Fig. 9(a) shows that even though the feature losses between the plain image and the encrypted/decoded image are not used, the visual information

---

\[ \text{Source codes for constructing the proposed method are available at: https://github.com/alexxxu1988/DJESCC.} \]
Fig. 6. The architecture of the DJSCC network [19] adopted in this paper.

Fig. 7. The architecture of the encryption and decryption networks adopted in this paper.

Fig. 8. Performance of DJESCC and DJSCC trained on CIFAR-10 training dataset and evaluated on CIFAR-10 test dataset with R = 1/6.

Of the plain image is partially protected in the encrypted image and the decoded image. The outline of the dog can be vaguely identified in the encrypted image in Fig. 9(a), and less visual information is captured in the decoded images than in the encrypted image. With the increase in loss weights $\lambda_c$ and $\lambda_d$, the visual protection in the encrypted image is enhanced, and the corresponding visual pattern in the decoded image is also changed. In Fig. 9(b) with the loss weights $\lambda_c = \lambda_d = 0.005$, the outline of the dog can be vaguely identified in the encrypted image. The encrypted image decays to regular black and white lattice with no visual information of the plain image in Fig. 9(c) with the loss weights $\lambda_c = \lambda_d = 0.05$. As the loss weights increase to $\lambda_c = \lambda_d = 0.5$, the left part of the encrypted image contains alternated red and blue stripes and the right part of it contains twisted black and white lattice, which is more irregular than the encrypted image in Fig. 9(d).

In Fig. 9(b)(c)(d), all of the decoded images with different SNRs can protect visual information of the plain image. The proposed method with $\lambda_d = \lambda_c = 0.05$ shows a better trade-off between the reconstruction performance and the visual protection performance than others.

Since our proposed method is the first work that constructs visually protected image for DJSCC transmission, we compare the proposed method with two perceptual image encryption methods, i.e. the learnable image encryption (LE) method [48] and the pixel-based image encryption (PE) method [50], which are designed for image classification task. The classification accuracy of ResNet-20 [73] based on the LE method and the PE method are 87.02% and 86.99% on CIFAR-10 test dataset, respectively [52].

Fig. 10 compares the DEJSCC method with the LE based DJSCC (DJSCC_LE) method and the PE based DJSCC (DJSCC_PE) method for bandwidth ratios R = 1/6 and R = 1/12. The performance of the DJSCC_PE with R = 1/6 is slightly increased around 11.5 dB as the SNR increases from 0dB to 20dB, which is much lower than that of the DJSCC with $\lambda_c = \lambda_d = 0.05$ and R = 1/6. Although the performance of the DJSCC_PE with R = 1/6 is better than that of the DJSCC_LE with R = 1/6, the performance of the DJSCC_PE with R = 1/6 is still 5 dB lower than that of the DJSCC at SNRtest = 0 dB and the performance gap between the DJSCC_PE and the DJSCC with $\lambda_c = \lambda_d = 0.05$ is further widened with the increase of the SNR. Comparing the methods with R = 1/12 shows the similar results. With the increase of R from 1/12 to 1/6, DEJSCC achieves more gains than DJSCC_PE and DJSCC_LE. Fig. 11 shows the
corresponding visual performance for DJECC, DJSCC_LE, DJSCC_PE with SNR = 0, 10, 20 dB and θ = 1/6. Different from the black and white lattice characteristic of the encrypted image shown in the DJECC method, the encrypted images of the DJSCC_LE method and the DJSCC_PE method are shown as noisy images. However, the pixels in the part of the encrypted images corresponding to that in the dog part of the plain image show less randomness than the pixels in other parts. The decoded images of the DJSCC_LE method and the DJSCC_PE method are similar to the corresponding encrypted images of the DJSCC_LE method and the DJSCC_PE method, respectively. The decrypted images of the DJSCC_LE method at SNR = 0, 10, 20 dB are disturbed by some noisy pixels, while the decrypted images of the DJSCC_PE method at SNR = 0, 10, 20 look blurred.

V. VISUAL SECURITY OF THE PROPOSED METHOD

As described in Section I, it is unsafe during the network transmission. The visually protected images—the encrypted image generated by the image owner then sent to the DJSCC provider through the Internet and the decoded image processed by the DJSCC provider then sent to the image recipient—may be leaked or stolen in the wired network. Hence the ciphertext-only attack (CoA), where the attacker has access to decoded image and wishes to recover plain image, must be considered to evaluate the visual security of the proposed method. In this paper, we focus on two CoAs: feature reconstruction (FR) attack and generative adversarial network based (GAN-based) attack.

A. FR Attack

Based on the fact that many visually protected methods deal with the plain image in patch (e.g., the LE method operates
the pixels in a 4×4 block) or in pixel (e.g., the PE method only deal with the pixels without change their positions), FR attack proposed by [74] is employed to reconstruct the edge information of plain images from the encrypted images. The algorithm of FR attack is described in Algorithm 1, where ⌊·⌋ denotes the floor function.

Algorithm 1 The FR-attack method [74]

Input: the 8-bit visually protected image \( y \in \mathbb{R}^{h \times w \times 3} \), leading bit \( b \in \{0, 1\} \)

Output: the 8-bit restored image \( \hat{x} \in \mathbb{R}^{h \times w \times 3} \)

1: the 8-bit RGB image is divided into \( m = h \times w \) pixels, \( y(i) \in \mathbb{R}^{1 \times 1 \times 3} \) represents the \( i \)-th pixel
2: for \( i = 0 : 1 : h \times w \) do
3: for \( j = 0 : 1 : 2 \) do
4: if \( y(i, j) \neq b \) then
5: \( \hat{x}(i, j) = y(i, j) \oplus (2^8 - 1) \)
6: end if
7: end for
8: end for
9: assemble \( \hat{x}_i, i \in [1, m] \) to form \( \hat{x} \in \mathbb{R}^{h \times w \times 3} \)
10: return \( \hat{x} \)

B. GAN-based Attack

GAN-based attack adopted in [75] has shown state-of-the-art attack performance for CoA. As illustrated in Fig. 12, the Generator intends to generate images from encrypted images to cheat the discriminator while the purpose of the discriminator is to well distinguish whether the images fed into it originates from the plain image dataset or from the decrypted images. In the training stage, the training dataset containing plain images is divided into \( X_1 \) and \( X_2 \) with the same size. \( X_2 \) is encrypted by some encryption method. Both \( X_1 \) and \( E(X_2) \) are employed for alternatively training the generator and the discriminator. Note that paired images are not required in the training stage. With the progress of training, the generator can gradually generate the images more and more like plain images from the encrypted images. In other words, the generator decrypts the encrypted images and gets the visual information.

The generators proposed in [75] are used as generators of GAN-based attack for the DJSCC_LE method and the DJSCC_PE method except using the sigmoid activation instead of tanh activation in the last layer. The transformation architecture shown in Fig. 7 is employed as the generator for the DJESCC method. The discriminator contains three convolutional layers with kernel size 2, stride 2, the number of channels 16, 16, 32 and leaky ReLU activation, and a dense layer with sigmoid activation. The models of GAN-based attack networks were trained for 600 epochs by using the Adam optimizer with the learning rate 0.0001 and batch size with 64.

C. Visual Performance of CoAs

We employ the STL-10 dataset in the expriments for a better visual comparison. Full convolutional architecture based DJSCC can deal with tasks of different image sizes in the test phase [19] and [22]. The DJESCC model is trained on CIFAR-10, which is unknown to the attacker. The LE method and the PE method are size sensitive. Hence the DJSCC_LE method and the DJSCC_PE method were trained on the STL-10 dataset under a uniform distribution within the SNR range [0, 20] dB. FR attack can be directly employed on the cipher images (the encrypted image and the decoded image generated by DJESCC, DJSCC_LE, or DJSCC_PE), while the GAN-based attack should first be trained and then be employed on the cipher images. The GAN-based attack was trained and tested on STL10 dataset for DJESCC, DJSCC_LE and DJSCC_PE, respectively.

Visual information reconstructed by FR attack and GAN-based attack with \( SNR_{test} = 10 \) dB are shown in Fig. 13. For attacking encrypted images by FR attack, edge information against the PE method can be successfully reconstructed and edge information against the LE method can be vaguely reconstructed. However, the blank image generated by FR attack against the DJESCC method shows that FR attack can not reconstruct any edge information from the encrypted image generated by the DJESCC method. Similar results can be found for attacking decrypted images by FR attack, except that the edge information can not be reconstructed by FR attack against the PE method because of the low DJSCC transmission quality in the DJSCC_PE method. GAN-based attack can successfully recover visual information from the encrypted image protected by the DJSCC_LE method, the decrypted image protected by the DJSCC_LE method, and the encrypted image protected by the DJSCC_PE method, while little visual information is recovered from the encrypted image and the decrypted image protected by the DJESCC method. Compared with the DJSCC_LE method and the DJSCC_PE method, the visual protected images (i.e., the encrypted image and the decoded image) generated by the DJESCC method are more robust against FR attack and GAN-based attack.
Plain Image | Encrypted Image | Decoded Image | Decrypted Image
---|---|---|---
![Plain Image](image1.png) | ![Encrypted Image](image2.png) | ![Decoded Image](image3.png) | ![Decrypted Image](image4.png)
PSNR: 4.81 dB | PSNR: 5.04 dB | PSNR: 5.12 dB | PSNR: 5.13 dB
SSIM: 0.004 | SSIM: 0.006 | SSIM: 0.006 | SSIM: 0.006
[i) SNR = 0 dB](image5.png) | [ii) SNR = 10 dB](image6.png) | [iii) SNR = 20 dB](image7.png) | [iv) SNR = 0 dB](image8.png)

(a) DJESCC (\(\lambda_e = \lambda_d = 0.05\))

![Fig. 11. Visually protected images comparison generated by the DJESCC method, the DJSCC_LE method, and the DJSCC_PE method with R = 1/6. The image in the first column is the plain image. The images in the second column are the encrypted images transformed by the image owner. The images in the third column are the decoded images decoded by the DJSCC decoder for SNR = 0, 10, 20 dB with different methods. The images in the last column are the decrypted images transformed by the image recipient for SNR = 0, 10, 20 dB with different methods. PSNR and SSIM are given under images.](image9.png)

(b) DJSCC_LE

(c) DJSCC_PE

![Fig. 12. GAN-based attack.](image10.png)

VI. CONCLUSION

Inspired by image transformation, we have proposed a novel DJESCC method. By applying end-to-end training, the proposed DJESCC method learned two DNNs to transform the images, i.e., one transformed from the plain image domain to the encrypted image domain for encryption, and the other one transformed from the DJSCC decoded image domain for decryption. Besides that, the proposed DJESCC method simultaneously learned an effective DJSCC transmission method in the encryption domain during the training stage.

Compared with perceptual image encryption methods (e.g., the LE method and the PE method) for DJSCC, the proposed DJESCC method has shown a much better reconstruction performance. In addition, the experimental results demonstrated that the proposed DJESCC method were robust to the two types of CoA, i.e., the FR attack and the GAN-based attack. It is worth noting that the proposed DJESCC method is a general method for protecting visual information of the plain image transmitted via DJSCC. With some appropriate modifications of the DJESCC, the proposed mechanism can be applied to various different DJSCC architectures, e.g., the DJSCC-I [20], the DJSCC-f [21], and the ADJSCC [22].

REFERENCES

[1] T. M. Cover, *Elements of Information Theory*. John Wiley & Sons, 1999.

[2] M. Skoglund, N. Phamdo, and F. Alajaji, “Hybrid digital–analog source–channel coding for bandwidth compression/expansion,” *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3757–3763, 2006.

[3] R. G. Gallager, *Information Theory and Reliable Communication*. Springer, 1968, vol. 2.

[4] I. Csiszár, “Joint source-channel error exponent,” *Problems of Control & Information Theory*, vol. 9, pp. 315–328, 1980.

[5] Y. Zhong, F. Alajaji, and L. L. Campbell, “Joint source–channel coding error exponent for discrete communication systems with markovian memory,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4457–4472, 2007.

[6] ——, “Error exponents for asymmetric two-user discrete memoryless source-channel systems,” in *2007 IEEE International Symposium on Information Theory*. IEEE, 2007, pp. 1736–1740.
| Plain Image                                      | Encrypted Image                                      | FR Attack | GAN-based Attack | Decoded Image                                      | FR-Attack | GAN-based Attack |
|------------------------------------------------|------------------------------------------------------|-----------|------------------|---------------------------------------------------|-----------|------------------|
| ![Plain Image](image1.png)                     | ![Encrypted Image](image2.png)                      | ![FR Attack](image3.png) | ![GAN-based Attack](image4.png)                    | ![Decoded Image](image5.png)                        | ![FR-Attack](image6.png) | ![GAN-based Attack](image7.png) |
| PSNR:6.90dB                                      | PSNR:6.90dB                                         | PSNR:3.99dB | PSNR:5.17dB       | PSNR:7.04dB                                        | PSNR:6.99dB |                 |
| SSIM:0.019                                       | SSIM:0.019                                          | SSIM:0.079 | SSIM:0.004        | SSIM:0.028                                         | SSIM:0.00  |                 |

(a) DJESCC ($\lambda_d = 0.05$)

| ![Plain Image](image1.png) | Encrypted Image                                      | FR Attack | GAN-based Attack | Decoded Image                                      | FR-Attack | GAN-based Attack |
|-----------------------------|------------------------------------------------------|-----------|------------------|---------------------------------------------------|-----------|------------------|
| ![Plain Image](image1.png) | ![Encrypted Image](image2.png)                      | ![FR Attack](image3.png) | ![GAN-based Attack](image4.png)                    | ![Decoded Image](image5.png)                        | ![FR-Attack](image6.png) | ![GAN-based Attack](image7.png) |
| PSNR:7.62dB                 | PSNR:10.22dB                                         | PSNR:17.32dB | PSNR:8.57dB       | PSNR:11.23dB                                       | PSNR:7.36dB |                 |
| SSIM:-0.002                 | SSIM:0.061                                          | SSIM:0.504 | SSIM:0.005        | SSIM:0.068                                         | SSIM:-0.159 |                 |

(b) DJSCC_LE

| ![Plain Image](image1.png) | Encrypted Image                                      | FR Attack | GAN-based Attack | Decoded Image                                      | FR-Attack | GAN-based Attack |
|-----------------------------|------------------------------------------------------|-----------|------------------|---------------------------------------------------|-----------|------------------|
| ![Plain Image](image1.png) | ![Encrypted Image](image2.png)                      | ![FR Attack](image3.png) | ![GAN-based Attack](image4.png)                    | ![Decoded Image](image5.png)                        | ![FR-Attack](image6.png) | ![GAN-based Attack](image7.png) |
| PSNR:8.04dB                 | PSNR:10.61dB                                         | PSNR:10.49dB | PSNR:8.50dB       | PSNR:10.29dB                                       | PSNR:7.87dB |                 |
| SSIM:-0.015                 | SSIM:0.246                                          | SSIM:0.330 | SSIM:0.014        | SSIM:0.044                                         | SSIM:0.027 |                 |

(c) DJSSC_PE

Fig. 13. Reconstructed visual information from the encrypted image and the decoded image by FR attack and GAN-based attack at SNR$_{can}$=10 dB. The image in the first column is the plain image. The images in the second column are the encrypted image, FR attack for the encrypted image and GAN-based attack for the encrypted image. The images in the third column are the decoded image, the FR-attack for the decode image, and the GAN-attack for the decoded image. decoded by the DJSCC decoder for SNR= 0, 10, 20 dB. PSNR and structural similarity index (SSIM) are given under the images. (a) DJESCC with $\lambda_e = \lambda_d = 0.05$, (b) DJSCC_LE, (c) DJSSC_PE.

[7] B. Belzer, J. D. Villasenor, and B. Girod, “Joint source channel coding of images with trellis coded quantization and convolutional codes,” in *Proceedings., International Conference on Image Processing*, vol. 2. IEEE, 1995, pp. 85–88.

[8] S. Heinen and P. Vary, “Transactions papers source-optimized channel coding for digital transmission channels,” *IEEE Transactions on Communications*, vol. 53, no. 4, pp. 592–600, 2005.

[9] J. Cai and C. W. Chen, “Robust joint source-channel coding for image transmission over wireless channels,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 962–966, 2000.

[10] T. Guionnet and C. Guillemot, “Joint source-channel decoding of quasairithmetic codes,” in *Data Compression Conference, 2004. DCC 2004*. IEEE, 2004, pp. 272–281.

[11] G. Toderici, S. M. O’Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, “Variable rate image compression with recurrent neural networks,” *arXiv preprint arXiv:1511.06085*, 2015.

[12] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” *arXiv preprint arXiv:1611.01704*, 2016.

[13] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” in *Advances in Neural Information Processing Systems*, 2018, pp. 10,771–10,780.

[14] D. Minnen and S. Singh, “Channel-wise autoregressive entropy models for learned image compression,” *arXiv preprint arXiv:2007.08739*, 2020.

[15] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–573, 2017.

[16] J. Xu, W. Chen, B. Ai, R. He, Y. Li, J. Wang, T. Juhana, and A. Kurniawan, “Performance evaluation of autoencoder for coding and modulation in wireless communications,” in *2019 11th International Conference on Wireless Communications and Signal Processing (WCP)**. IEEE, 2019, pp. 1–6.

[17] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, “Learn codes: Inventing low-latency codes via recurrent neural networks,” *IEEE Journal on Selected Areas in Information Theory*, 2020.

[18] ——, “Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels,” in *Advances in Neural Information Processing Systems*, 2019, pp. 2758–2768.

[19] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, “Deep joint source-channel coding for wireless image transmission,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[20] D. B. Kurka and D. Gündüz, “Bandwidth-agile image transmission with deep joint source-channel coding,” *IEEE Transactions on Wireless Communications*, 2021.

[21] ——, “Deepjssc-f: Deep joint source-channel coding of images with feedback,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[22] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, “Wireless image transmission using deep source channel coding with attention modules,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[23] T. Chuman, W. Sirichotedumrong, and H. Kiya, “Encryption-then-compression systems using grayscale-based image encryption for jpeg images,” *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 6, pp. 1515–1525, 2019.

[24] ——, “Special feature exhaustive cryptanalysis of the nbs data encryption standard,” *Computer*, vol. 10, no. 6, pp. 74–84, 1977.

[25] J. Daemen and V. Rijmen, “Reijndael: The advanced encryption stan-
machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[72] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

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

[74] A. H. Chang and B. M. Case, “Attacks on image encryption schemes for privacy-preserving deep neural networks,” *arXiv preprint arXiv:2004.13263*, 2020.

[75] W. Sirichotedumrong and H. Kiya, “Visual security evaluation of learnable image encryption methods against ciphertext-only attacks,” in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 1304–1309.