Deep Joint Source-Channel and Encryption Coding: Secure Semantic Communications

Tze-Yang Tung, Deniz Gündüz
Department of Electrical and Electronics Engineering, Imperial College London
{tze-yang.tung14, d.gunduz}@imperial.ac.uk

Abstract—Deep learning driven joint source-channel coding (JSCC) for wireless image or video transmission, also called DeepJSCC, has been a topic of interest recently with very promising results. The idea is to map similar source samples to nearby points in the channel input space such that, despite the noise introduced by the channel, the input can be recovered with minimal distortion. However, the inherent correlation between the source sample and channel input makes DeepJSCC vulnerable to eavesdropping attacks. In this paper, we propose the first DeepJSCC scheme for wireless image transmission that is secure against eavesdroppers, called DeepJSCEC. The proposed solution not only preserves the results demonstrated by DeepJSCC, it also provides security against chosen-plaintext attacks from the eavesdropper, without the need to make assumptions about the eavesdropper’s channel condition or its intended use of the intercepted signal.

I. INTRODUCTION

Shannon’s separation theorem [1] is the foundation of most modern communication systems. It states that source coding, which reduces redundancies in the source signal, can be implemented separately from channel coding, which reintroduces redundancies in order to protect against channel distortions, without loss of optimality if given enough tolerance for end-to-end delay. In the context of wireless image transmission, commonly used compression schemes, such as better portable graphics (BPG) [2], allow for the reduction in communication load with minimal loss in reconstruction quality, before a channel code is applied to ensure reliable transmission of the compressed bits to the destination. This modular architecture also makes incorporating security into the design very easy, as the compressed bits can simply be encrypted using a known encryption scheme.

However, in practical applications we are limited to finite blocklengths, and it is known that combining the two coding steps, that is, joint source-channel coding (JSCC), can achieve lower distortion for a given finite blocklength than separate source and channel coding [3], [4]. JSCC has also been widely considered as a practical framework for semantic communications [5], where the communication scheme is jointly designed with the source, rather than

The source agnostic approach that the separation theorem suggests. Recently it was shown in [6] and [7] that deep neural networks (DNNs) can be used to break the complexity barrier of designing JSCC schemes for wireless image transmission. The scheme, called DeepJSCC, showed appealing properties, such as lower end-to-end distortion for a given channel blocklength compared to state-of-the-art separation-based schemes and graceful degradation of image quality with respect to channel quality.

Despite the success of DeepJSCC schemes, one major problem that has not been addressed is its security. DeepJSCC schemes learn the encoding and decoding functions from scratch by directly mapping the source signal to the modulated channel input, without conversion to bits. This means that the channel input symbols are directly correlated with the input sample. It is this property that enables graceful degradation of end-to-end distortion, but it also means known encryption schemes cannot be applied directly. Although the problem of joint source-channel coding and secrecy has been previously investigated in [8] and [9], these schemes are based primarily on randomization and permutation, making them susceptible to chosen-plaintext attacks. Information theoretic privacy over a wiretap channel has also been investigated in [10] and [11], where the objective is to ensure the intended communication link between two users is able to achieve its objective (e.g., reconstruct an image), while an eavesdropper cannot infer a certain feature of the source (e.g., color, gender,...etc.). However, in these works, the adversary’s channel quality is assumed to be significantly worse.

In this paper, we propose a DeepJSCC scheme for wireless image transmission that is secure against eavesdroppers, called deep joint source-channel and encryption coding (DeepJSCEC), without making any assumptions on the eavesdropper’s channel condition. We leverage the affine property of the encryption scheme presented in [12], based on the learning with errors (LWE) problem [13], to construct a secure wireless image transmission scheme. We emphasize that this paper does not prove the security of the cryptographic scheme presented in [12]. We will assume its security and present a DeepJSCC scheme exploiting it. The contributions of this paper are summarized as follows:

1) We propose the first DeepJSCC scheme for wireless image transmission that is secure against eaves-
droppers, called DeepJSCEC, without making any assumptions on the eavesdropper’s channel quality.

2) DeepJSCEC not only retains the properties that prior works on DeepJSCC have shown, such as graceful degradation of image quality against varying channel quality and lower end-to-end distortion, it is also secure against chosen-plaintext attacks from the eavesdropper.

3) The cryptographic scheme used is a public-key encryption scheme, meaning anyone can send encrypted messages to the intended user using the public key, without the eavesdropper obtaining the message. Moreover, new keys can be generated independent of the encoder and decoder models, meaning there is no need to retrain them.

4) Numerical results show that DeepJSCEC achieves similar or better image quality than separation-based schemes employing BPG [2] for source coding, low density parity check (LDPC) [14] codes for channel coding and advanced encryption standard (AES) [15] for encryption.

II. PROBLEM STATEMENT

We consider the problem of wireless image transmission over an additive white Gaussian noise (AWGN) channel, where Alice wants to send an image to Bob without Eve, the eavesdropper, obtaining a good estimate of the image based on a chosen metric. Formally, Alice sends an image \( x \in \{0, ..., 255\}^{H \times W \times C} \) (where \( H \), \( W \) and \( C \) represent the image’s height, width and color channels, respectively) using an encoder function \( f : \{0, ..., 255\}^{H \times W \times C} \rightarrow \mathbb{C}^k \). For an RGB image, \( C = 3 \). Letting \( y = f(x) \), we impose an average transmit power constraint \( \bar{P} \), such that

\[
\frac{1}{k} \sum_{i=1}^{k} |y_i|^2 \leq \bar{P},
\]

where \( y_i \) is the \( i \)-th element of vector \( y \). The channel input vector \( y \) is transmitted through a noisy channel, with the transfer function \( \hat{y} = \mathcal{Y}(y) = y + n \), where \( n \sim CN(0, \sigma_n^2 \mathbf{I}_{k \times k}) \) is a complex Gaussian vector with dimensionality \( k \). Bob receives the channel output and decodes it with the function \( g : \mathbb{C}^k \rightarrow \{0, ..., 255\}^{H \times W \times C} \) to produce a reconstruction of the input \( \hat{x} = g(\hat{y}) \). Eve, an eavesdropper, has access to the transmitted message \( y \) via an eavesdropping channel \( \mathcal{Y} = \mathcal{Y}(y) = y + \bar{n} \), where \( \bar{n} \sim CN(0, \sigma_{\bar{n}}^2 \mathbf{I}_{k \times k}) \) is the channel noise observed by Eve with variance \( \sigma_{\bar{n}}^2 \).

We will also assume that Eve has access to known pairs of \((x, y)\) such that she can try to find a decoder \( \hat{x} = g(\bar{y}) \). This is called a chosen-plaintext attack. A diagram illustrating the problem is shown in Fig. 1.

Given the above problem definition, we define the channel SNR between Alice and Bob as

\[
\text{SNR}_b = 10 \log_{10} \left( \frac{P}{\sigma_x^2} \right) \text{ dB},
\]

and the corresponding channel SNR for Eve as

\[
\text{SNR}_e = 10 \log_{10} \left( \frac{P}{\sigma_{\bar{n}}^2} \right) \text{ dB}.
\]

We also define the bandwidth compression ratio as

\[
\rho = \frac{k}{H \times W \times C} \text{ channel symbols/pixel},
\]

where a smaller number reflects more compression.

We will consider two distortion metrics for images. The first is the mean squared error (MSE),

\[
\text{MSE}(x, \hat{x}) = ||x - \hat{x}||_2^2.
\]

A common complimentary metric to the MSE distortion to measure image reconstruction quality is the peak signal-to-noise ratio (PSNR) defined as

\[
\text{PSNR}(x, \hat{x}) = 10 \log_{10} \left( \frac{A^2}{\text{MSE}(x, \hat{x})} \right) \text{ dB},
\]

where \( A \) is the maximum possible value for a given pixel. For a 24 bit RGB pixel, \( A = 255 \). Maximizing this metric corresponds to minimizing the PSNR.

The second is the structural similarity index measure (SSIM), defined as

\[
\text{SSIM}(x, \hat{x}) = \frac{2\mu_x\mu_{\hat{x}} + v_1}{\mu_x^2 + \mu_{\hat{x}}^2 + v_1} \frac{2\sigma_x\sigma_{\hat{x}} + v_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + v_2} \frac{2\sigma_{x\hat{x}} + v_3}{\sigma_{x\hat{x}}^2 + v_3},
\]

where \( \mu_x, \mu_{\hat{x}}, \sigma_x^2, \sigma_{\hat{x}}^2, \sigma_{x\hat{x}} \) are the mean and variance of \( x \), and the covariance between \( x \) and \( \hat{x} \), respectively, and \( v_1, v_2 \) are coefficients for numeric stability. Since SSIM has a maximum value of 1, maximizing this metric corresponds to minimizing \( 1 - \text{SSIM} \).

For the security metric, we will use ciphertext indistinguishability under chosen-plaintext attack (IND-CPA) to evaluate the security of the scheme. It can be defined formally in the form of a public-key cryptographic game as follows

Definition 1. Let \( K \) be a key generation function, \( \mathcal{E} \) be an encryption function, and \( \mathcal{D} \) be a decryption function that defines a cryptographic scheme \( \mathcal{S} \mathcal{E} = (K, \mathcal{E}, \mathcal{D}) \). Let an adversary (Eve), be computationally bounded by a probabilistic polynomial-time Turing machine. The game for testing IND-CPA is defined as:

1) Initialize \((P_k, S_k) \overset{\$}{\leftarrow} K, b \overset{\$}{\leftarrow} \text{Ber}(0.5)\), where \( \text{Ber}(0.5) \) is a Bernoulli distribution with success probability 0.5 and \( \overset{\$}{\leftarrow} \) refers to the sampling oper-
2023 IEEE International Conference on Communications (ICC): SAC Machine Learning for Communications and Networking Track

2) Knowing the public key $P_k$, Eve is able to generate any number of ciphertexts within polynomial-time.
3) Eve chooses two plaintext messages $M_0$ and $M_1$ of equal length, and provides them and $P_k$ to an oracle, who computes the ciphertext $C \rightleftharpoons \mathcal{E}(M_0, P_k)$, returning the ciphertext of one of the chosen plaintext messages determined by $b$.
4) Based on $C$, Eve tries to determine $M_b$ as the plaintext used to compute $C$.

The advantage that Eve has for a given encryption scheme $\mathcal{E}$ is defined as

$$Adv_{\mathcal{E}}^{IND-CPA} = 2P(b' = b) - 1. \quad (8)$$

As such, an advantage of 0 means $P(b' = b) = 0.5$, which corresponds to a random guess by Eve. In the context of our problem, we wish to allow Eve to best learn a decoding function $\hat{g}$ to reconstruct $\hat{g}(\hat{y}) \sim P(x)$, where $P(x)$ is the prior distribution on the source images, within probabilistic polynomial-time. The goal is then to design an encoder, decoder pair $(f, g)$, such that the reconstruction quality $d(x, \hat{x})$, measured by either Equations (6) or (7), is maximized, while giving Eve an advantage

$$Adv_{\mathcal{E}}^{IND-CPA} \approx 0 \text{ based on IND-CPA}.$$

III. PROPOSED SOLUTION

An input image $x$ is mapped with a non-linear encoder function $f_\theta : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^k$, parameterized by $\theta$, into a latent vector $z = f_\theta(x)$. Each value in the latent vector $z$ is then quantized into $N$ uniform quantization levels, with centroids $C_q = \{q_1, ..., q_N\}$, via the quantizer $q_{c_q} : \mathbb{R}^k \rightarrow C_q$, which we will define in Sec. III-A. That is, $\hat{z} = q_{c_q}(z)$, with $\hat{z} \in C_q$, where $\hat{z}$ is the $i$th element of the quantized vector $\hat{z}$. We will refer to $\hat{z}$ as the plaintext.

The plaintext $\hat{z}$ is then encrypted using a public-key encryption scheme $\mathcal{E} : \mathbb{Z}^n_p \times \mathbb{Z}^{n_1 \times k}_p \rightarrow \mathbb{Z}^k_p$, producing a ciphertext $C = E(\hat{z}, P(S))$, where $P(S) \in \mathbb{Z}^{n_1 \times k}_p$ is the public key and a function of the secret key $S \in \mathbb{Z}^{n_2 \times k}_p$. The constants $(n_1, n_2)$ will be determined later. Let there also be a corresponding decryption scheme $D(c, P(S), S) = \hat{z}'$. We will assume for now that the secret key $S$ cannot be inferred from the public key $P(S)$. The entire encryption procedure will be described in Sec. III-B.

The ciphertext $C$ is then modulated using a constellation

$$\mathcal{C} = \{c_1, ..., c_p\} \text{ of order } p,$$

by mapping each $c_j \in \mathbb{Z}_p$ to the corresponding constellation point in $\mathcal{C}$, producing channel input $y \in \mathbb{C}^k$. To ensure that the power constraint is met, we choose the constellation points such that the average power of the constellation points assuming uniform probability is $P = \frac{1}{p} \sum_{j=1}^{p} |c_j|^2$. The channel input $y$ is then transmitted through an AWGN channel, producing the channel output $\hat{y} = y + n$.

At the receiver, the likelihood of each received value is first computed as

$$l_i = P(\hat{y}_i | c_j), \quad (9)$$

where $\hat{y}_i$ is the $i$th element in $\hat{y}$ and $c_j$ is the $j$th constellation point in $\mathcal{C}$. We then compute the noisy ciphertext $\hat{c} \in \mathbb{R}^k$ by computing the softmax weighted sum of values in $\mathbb{Z}_p$ based on the likelihood vector $l_i = \{l_1, ..., l_i\}$ for $i = 1, ..., k$ as

$$\hat{c}_i = \sum_{j=1}^{p} e^{\sigma_i l_i} / \sum_{n=1}^{p} e^{\sigma_i l_i} (j - 1), \quad (10)$$

where $\hat{c}_i$ is the $i$th element in $\hat{c}$ and $\sigma_i$ is the parameter controlling the weighting of the likelihoods. Let us denote the channel noise in the ciphertext space as $n_c = \hat{c} - c$. Given the noisy ciphertext $\hat{c}$, we can obtain a noisy plaintext $\hat{x}$ by using the corresponding decryptor $D : \mathbb{R}^k \rightarrow \mathbb{R}_p^k$ and the secret key $S$ as $\hat{x} = D(\hat{c}, P(S), S)$.

The resultant noisy plaintext $\hat{x}'$ is generally too noisy to be decoded effectively. Therefore, we compute the softmax weighted sum of the symbols in $C_q$ based on their $l_2$ distances from $\hat{z}' \in \mathbb{Z}_q'$; that is,

$$\hat{z}_i = \frac{\sum_{j=1}^{N} e^{-d'_i j}}{\sum_{n=1}^{N} e^{-d'_i n}} q_j, \quad (11)$$

where $d'_i j = ||z'_i - q_j||_2$, is the $l_2$ distance between $z'_i$ and the centroid $q_j \in C_q$.

Finally, the estimate $\hat{z}$ is passed through a non-linear decoder function $g_\phi : \mathbb{R}^k \rightarrow \mathbb{R}^{H \times W \times C}$, parameterized by $\phi$, to produce a reconstruction of the input $\hat{x} = g_\phi(\hat{z})$.

A. Quantization

Given the encoder output $\mathbf{z}$, we first obtain a “hard” quantization, which simply maps element $z_i \in \mathbf{z}$ to the nearest symbol in $C_q$. This forms the quantized latent vector $\mathbf{z}$. However, this operation is not differentiable. In order to obtain a differentiable approximation of the hard quantization operation, we will use the “soft” quantization approach, proposed in [16]. In this approach, each quantized value is generated as the softmax weighted sum of the symbols in $C$ based on their $l_2$ distances from $z_i$; that is,

$$z_i = \frac{\sum_{j=1}^{N} e^{-\sigma_i d_{ij}}}{\sum_{n=1}^{N} e^{-\sigma_i d_{in}}} q_j, \quad (12)$$

where $\sigma_i$ is a parameter controlling the “hardness” of the assignment, and $d_{ij} = ||z_i - q_j||_2$ is the $l_2$ distance between the latent value $z_i$ and the centroid $q_j$. As such, in the forward pass, the quantizer uses the hard quantization $\mathbf{z}$, and in the backward pass, the gradient from the soft quantization $\hat{z}$ is used to update $\mathbf{z}$. That is, $\frac{\partial}{\partial \mathbf{z}} = \frac{\partial}{\partial \hat{z}}$. To ensure that the quantized values lie within $\mathbb{Z}_p$, we define the centroids to be $C_q = \{\lfloor \frac{C_q}{N} \rfloor \}_{i=0}^{N-1}$.
B. Encryption

In DeepJSCEC, we use the public key cryptographic scheme presented in [12] as follows. Let A ∈ Z_p^{n_1×n_2} be a matrix whose values are sampled uniformly from Z_p and (S ∈ Z_p^{n_2×k}, U ∈ Z_p^{n_1×k}) be matrices whose values are sampled i.i.d. from a discrete distribution χ. Let P(S) = (U - AS, A) be the public key and S as the secret key. Then, to encrypt with the public key, the ciphertext is computed from a plaintext z ∈ Z_p as

\[ E(z, P(S)) = ((U - AS)^{T}e_1 + e_3 + z, A^{T}e_1 + e_2) \pmod{p} \]

\[ \Delta = (c, d) \pmod{p}, \]

where e = (e_1, e_2, e_3) ∈ Z^{n_1} × Z^{n_2} × Z^k are all sampled i.i.d. from χ as well. In this paper, we will use a discrete Gaussian distribution χ = G_{Z_p, σ}, as proposed in [12], which is a zero mean Gaussian random variable with variance \( σ^2/2π \) and round the samples to the nearest integer. In order to decipher, the legitimate recipient, who has the secret key S, computes

\[ \hat{z} = D(c, d, P(S), S) = S^{T}d + c \pmod{p} \]

\[ = S^{T}e_2 + U^{T}e_1 + e_3 + \hat{z} \pmod{p}. \]

where [\hat{z}] = z (mod p). The conditions for correctness given a set of parameters \((n_1, n_2, σ)\) can be found in [12].

The reason that this scheme is secure is because it is based on the LWE problem [13], which stipulates that given A ∈ Z_p^{n_1×n_2} as above and (S ∈ Z^{n_1}, e ∈ Z^{n_2}) sampled independent and identically distributed (i.i.d.) from χ, it is difficult to recover s from the pair (A, A^{T}s + e) in polynomial-time. The difficulty lies in the fact that it is difficult to distinguish pairs of (A, A^{T}s + e) from (A, b'), where b' is sampled uniformly from Z_p^{n_1}. In fact, in [13], it is stated that the best known algorithm for distinguishing (A, A^{T}s + e) from (A, b') with an advantage > 0 requires \( 2^{O(n_1/\log n_1)} \) space and time. This meets the IND-CPA security metric defined in Def. 1.

To adapt it to DeepJSCEC, we will assume that both Alice and Bob have the same seed to generate the error terms in e. Therefore, we can assume that d is available to the receiver prior to transmission since it does not depend on the message z. Given the encryption procedure defined in Eqns. (13-15), the receiver receives a noisy ciphertext \( \hat{c} = c + n_c \), as outlined above, where n_c is the noise from the channel in the ciphertext space following Eqns. (9-10). The decryption is then performed as

\[ z' = D(\hat{c}, P(S), S) \]

\[ = S^{T}e_2 + U^{T}e_1 + e_3 + \hat{z} + n_c \pmod{p}. \]

Since the channel noise term n_c and the decryption error terms S^{T}e_2 + U^{T}e_1 + e_3 are additive noise and allow the DNNs to learn a JSCC codebook that is optimized for the compound noise from the channel and the encryption scheme, resulting in an IND-CPA secure DeepJSCEC scheme.

IV. Training Details

We train DeepJSCEC on the Tiny ImageNet dataset [17] and test on the CIFAR10 dataset [18]. We split the training samples in the ratio of 0.8 : 0.2 between training and validation, with a batch size of 64. We use the Adam optimizer [19] with learning rate 0.0001 and parameters \((β_1, β_2) = (0.9, 0.999)\). We also use early stopping with a patience of 10 epochs to prevent overfitting and a learning rate scheduler that multiplies the learning rate by a factor of 0.8 if the loss does not improve for 5 epochs in a row. As the encryption method and the modulation/demodulation procedures are not differentiable, the gradient through those operations is skipped during training, such that the gradient of the loss l(x, x̂) with respect to z is

\[ \frac{\partial l(x, x̂)}{\partial z} = \frac{\partial l(x, x̂)}{\partial z}. \]

The network architectures are shown in Table I. In the architecture, C refers to the number of channels in the output tensor of the convolution operation. C_{out} refers to the number of channels in the final output tensor of the encoder f_θ, which controls the number of channels per image. The “Pixel shuffle” module, proposed in [20], is used to increase the height and width of the input tensor.
by reshaping it, such that the channel dimension is reduced while the height and width dimensions are increased. The GDN layer refers to generalized divisive normalization, initially proposed in [21], and shown to be effective in density modeling and compression of images. The Attention layer refers to the simplified attention module proposed in [22].

For the encryption scheme, we use parameters from [12], where they computed the values for \( p, n_1, n_2, \) and \( \sigma_s \) that provide a certain level of security. Specifically, we used \((p,n_1,n_2,\sigma_s,N) = (4093,192,192,8.87,16)\). In [12], the authors showed that these parameters can provide an advantage close to zero \((2^{-32})\) and an eavesdropper would optimistically require in the order of \(2^{42}\) seconds to decode the message. Although increasing the number of quantization levels \( N \) would in theory improve the performance, due to the error terms from the cryptographic scheme, increasing \( N \) also increases the relative magnitude of the error terms, thus reducing the performance. As such, we found \( N = 16 \) to be empirically a good trade-off between the resolution and performance. We assume that the public key \( \mathbf{P}(\mathbf{S}) \) is published ahead of transmission.

To aid exploration, the quantization hardness parameter \( \sigma_q \) is linearly annealed using the annealing function

\[
\sigma_q^{(i)} = \min \left( 200, \sigma_q^{(i-1)} + 5 \left\lfloor \frac{i}{2000} \right\rfloor \right), \tag{21}
\]

where \( i \) is the parameter update step number and we initialize with \( \sigma_q^{(0)} = 5 \). As for the soft ciphertext parameter \( \sigma_t \), we empirically found that \( \sigma_t = 5 \) works well in practice.

For the channel input constellation, since the encryption lattice modulus is 4093, we use a 4096-QAM and simply ignore the last 3 constellation points. The reason we used this constellation is because it avoids having to design a constellation of a non-standard size, and since the constellation size is sufficiently large, a waste of 3 constellation points is negligible. Moreover, due to the gradient skipping, as described by Eqn. (20), it is not possible to train a custom constellation either.

To compare our scheme with separation-based schemes, we will consider BPG [2] for source coding and LDPC codes for channel coding. The LDPC codes we use are from the 802.11ad standard [23], with blocklength 672 bits for both the 1/2 and 3/4 rate codes. We use the AES encryption scheme [15] for the baseline with block and key size of 128 bits. As such, we compress the images to a bit rate such that it is under the rate support by the LDPC code considered and is within an integer multiple of 128 bits. Just like the proposed DeepJSCEC, we will assume the keys are exchanged prior to transmission.

V. NUMERICAL RESULTS

Herein, we will consider models trained for fixed SNR_{Train} and tested over a range of channel SNRs. The SNR_{Train} values were chosen such that it coincides with the SNR where the LDPC codes and modulation orders we consider would fail, in order to highlight the behavior of DeepJSCEC around the cliff-edge of the digital baseline. Fig. 3 shows the performance of the models trained on specific SNR_{Train} and tested over a range of channel SNRs for Bob. It can be seen that the performance of DeepJSCEC gracefully degrades as the test SNR decreases, which is in contrast to the cliff-edge deterioration of the digital schemes at their respective threshold SNRs. This is true when we optimize for either the PSNR or the SSIM metric. Moreover, the image quality from DeepJSCEC for a given SNR is higher than the digital schemes. We note that although these characteristics have been shown previously by [6], [7], this is the first time an end-to-end encrypted JSCC scheme, implemented through DNNs, has shown these properties.

To demonstrate the security properties of DeepJSCEC, we devise a chosen-plaintext attack, where Eve has access to the encoder function \( f_\theta \), the public key \( \mathbf{P}(\mathbf{S}) \), and the dataset of images used to train \( (f_\theta, g_\phi) \). This allows Eve to obtain pairs of \((\mathbf{x}, c)\) and the objective for Eve is to...
use these pairs to obtain the secret key $S$ so that any image transmitted using $f_{\theta}$ and the encryption scheme $E$ using the public key $P(S)$ can be deciphered. Due to the complexity of search algorithms for finding $S$ based on $(x,c)$, we will instead train a DNN decoder $\hat{x} = f_{\theta_0}(c)$, parameterized by $\theta$, and attempt to optimize the parameters $\theta$ so that $d(x,\hat{x})$, as measured by either the PSNR or SSIM, is maximized. This is akin to the cryptographic game defined in Def. 1, where we essentially use a DNN or SSIM, is maximized. This is akin to the cryptographic game defined in Def. 1, where we essentially use a DNN to parameterize the secret key and search with gradient descent. We note that this corresponds to $\text{SNR}_c = \infty$ for Eve, as the ciphertexts are obtained without any additional noise from the channel. Therefore, this attack acts as an upper bound on how well Eve is expected to decode the intercepted signal.

We show in Table II, the resultant image quality for Bob and Eve after training. Bob's decoder was trained jointly with the encoder for $\text{SNR}_{\text{train}} = 10 \text{ dB}$ and evaluated at the same SNR.

| Bob     | Eve    | $x = E(x)$ |
|---------|--------|------------|
| PSNR    | SSIM   | PSNR       | SSIM   | PSNR       | SSIM   |
| 23.57 dB | 0.8113 | 11.28 dB    | 0.1803 | 12.73 dB    | 0.1965 |

VI. Conclusions

In this paper we have proposed DeepJSCEC, the first DNN-driven JSCC scheme for wireless image transmission that is secure against an eavesdropper without making any assumptions on its channel quality. DeepJSCEC not only achieves better end-to-end distortion than the conventional separation-based digital schemes, using BPG for compression, AES for encryption, and LDPC codes for channel coding, but also provides graceful degradation of image quality in varying channel conditions. The security of DeepJSCEC is measured under the IND-CPA criterion and it is shown that the proposed solution provides an advantage close to 0 to the eavesdropper. The proposed solution is highly practical and readily extendable to other end-to-end JSCC problems.

References

[1] C. E. Shannon, “A mathematical theory of communication,” Bell Syst. Tech. J., vol. 27, pp. 379–423 and 623–656, July and October 1948.
[2] F. Bellard, Better Portable Graphics, 2014 (accessed March 13, 2020). https://bellard.org/bpg/.
[3] V. Kostina and S. Verdú, “Lossy joint source-channel coding in the finite blocklength regime,” IEEE Transactions on Information Theory, vol. 59, pp. 2545–2575, May 2013.
[4] R. G. Gallager, Information Theory and Reliable Communication. USA: John Wiley & Sons, Inc., 1968.
[5] D. Gunduz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, “Beyond transmitting bits: Context, semantics, and task-oriented communications,” Oct. 2022.
[6] E. Bourtsoulatzis, D. Burth Kurka, and D. Gunduz, “Deep joint source-channel coding for wireless image transmission,” IEEE Transactions on Cognitive Communications and Networking, vol. 5, pp. 567–579, Sep. 2019.
[7] D. Burth Kurka and D. Gunduz, “Joint source-channel coding of images with (not very) deep learning,” in International Zurich Seminar on Information and Communication (IZS 2020). Proceedings, pp. 90–94, ETH Zurich, 2020.
[8] E. Magli, M. Grangetto, and G. Olmo, “Joint source, channel coding, and secrecy,” EURASIP Journal on Information Security, vol. 2007, pp. 1–7, 2007.
[9] M. Sinaie and V. T. Vakani, “A low complexity joint compression-error detection-cryptography based on arithmetic coding,” in 10th International Conference on Information Science, Signal Processing and Their Applications (ISSPA 2010), pp. 233–236, May 2010.
[10] E. Erdemir, P. L. Dragotti, and D. Gunduz, “Privacy-aware communication over a multi-hop channel with generative networks,” in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2989–2993, May 2022.
[11] T. Marchioro, N. Laurenti, and D. Gunduz, “Adversarial networks for secure wireless communications,” in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8748–8752, May 2020.
[12] R. Lindner and C. Pelikart, “Better key sizes (and attacks) for LWE-based encryption,” in Topics in Cryptology – CT-RSA 2011 (A. Kiayias, ed.), vol. 6558, pp. 319–339, 2011.
[13] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” in Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, STOC ’05, pp. 94–103, May 2005.
[14] R. Gallager, “Low-density parity-check codes,” IRE Transactions on Information Theory, vol. 8, pp. 21–28, Jan. 1962. Conference Name: IRE Transactions on Information Theory.
[15] S. Heron, “Advanced encryption standard (AES),” Network Security, vol. 2009, pp. 8–12, Dec. 2009.
[16] E. Agustsson et al., “Soft-to-hard vector quantization for end-to-end learning compressible representations,” in Advances in Neural Information Processing Systems 30, pp. 1141–1151, 2017.
[17] Y. Le and X. Yang, “Tiny ImageNet visual recognition challenge,” p. 6.
[18] A. Krizhevsky, “Learning multiple layers of features from tiny images,” tech. rep., University of Toronto, 2009.
[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in International Conference on Learning Representations (ICLR), 2015.
[20] P. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1874–1883, June 2016.
[21] J. Ballé, V. Larsson, and E. P. Simoncelli, “Density modeling of images using a generalized normalization transformation,” arXiv preprint arXiv:1511.06281, 2015.
[22] Z. Cheng, H. Sun, M. Takeuchi, and J. Katko, “Learned image compression with discretized Gaussian mixture likelihoods and attention modules,” Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7393–7398, 2020.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.